EFFORT ESTIMATION
in
AGILE DEVELOPMENT

Muhammad Tariq

Double M.Sc. in Computer Science and Applied IT

Supervised by

Professor Stephen MacDonell

A THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN THE
AUCKLAND UNIVERSITY OF TECHNOLOGY (AUT)
NEW ZEALAND

This Thesis is Dedicated
to the Loving Memories
OF

My belated Father Muhammad Rafique Zafar
Who was my Hero,
Great Role Model,
Wonderful Father,
Mentor,
& Great Soldier.

And

My Mother Nafeesa Parveen,
for her endless love and support

And
My wife Fozia for her all types of support

And
My Elder Brother Muhammad Tahir, for Supporting my Mother back at home

Abstract

Context: Organizations that develop software deploy considerable resources on projects in order to drive innovation and transformation. However, enduring evidence indicates that many software projects fail to deliver within time or budget and do not achieve business objectives. One of the reasons underlying such failures has been an inability to provide accurate forecasts of the time and resources needed for project completion. This may be due to the use of inherently poor estimation techniques, the poor use of good estimation techniques, the lack of use of any such techniques, or some combination of the three.

Numerous estimation techniques have been proposed for use in the software industry to forecast project or artefact size, the human effort required, the likely cost of development and the project schedule, with highly varying degrees of success. Some have proved to be effective in traditional software development (TSD) but their effectiveness in agile software development (ASD) is yet to be determined. ASD is currently the most popular methodology for the development of software because of its emphasis on fast release cycles, frequent delivery of value-added features to customers, and close collaboration between the product owner, the customer and the development team. The prevailing approach to forecast size, human effort and schedule for a software project in ASD is simple, which is a benefit, but still there are gaps and limitations in that approach. The current study is proposed to address the most important of these.

Purpose: This research study is conducted to understand and improve the utility of forecasts of size, effort, cost and schedule for ASD projects through variants of approaches based on a range of relevant objective and subjective indicators, including the size of stories, the velocity/capacity of the team, sprint and system complexity, the domain knowledge, expertise and experience of the team, stakeholder coordination, team relationships, project/sprint risks, and organizational culture, type, and work environment.

Methods: The Design Science Research Methodology encompasses three phases, being problem identification, solution design and evaluation. These phases have been enacted in this study through reviews of the state-of-the-art and the state-of-practice, quasi-experimentation, and detailed validations. The core empirical work in the study is conducted via quasi-experimental methods and face-to-face interviews with 21 IT experts. COEEMO – the proposed model of estimation along with two other suggested techniques (Bottom-Up and variants of Program Evaluation and Review Technique (PERT)) are applied to three real-industry ASD projects of AUSIT, and their performance is assessed using Mean Absolute Error (MAE), Balanced Relative Error (BRE), BREbias, and Standardized Accuracy (SA). To determine the statistical significance of the above suggested

methods one-way ANOVA (Parametric), and Kruskal-Wallis (Non-Parametric) tests are conducted, while the effect size (Glass delta – Δ) is calculated to determine the practical efficacy of COEEMO. Further, COEEMO is corroborated through empirical and laboratory validations: the model is implemented by using Mamdani Fuzzy Inference (MFIS) in the MATLAB toolbox and applied to the data of 85 sprints from ten previously completed ASD projects.

Contributions: There are three primary contributions deriving from this research in the area of effort estimation with the intent to improve accuracy in effort estimates for ASD projects: design and development of the novel COEEMO model, which delivers a pooled effect size Glass delta Δ of 0.23, outperforming the existing method of estimation by exhibiting 56% and 43% overall improvements in accuracy during the field-trials and empirical evaluations, respectively; a tailored approach of PERT, referred to as PERT's Variant-1, which shows 25% enhancement in accuracy during field-trials; and the use of Expert Judgement based on a Bottom-Up approach, demonstrating a 47% improvement in accuracy for effort estimation as compared to AUSIT's existing method during the field-trials.

Acknowledgments

I am delighted to express my earnest gratitude to my primary supervisor Prof Stephen MacDonell for bestowing his dedication, professionalism, guidance, and continuous support during the journey of my doctoral research. Steve has an extraordinarily professional aptitude and personality who kept me motivated and directed through the multiple phases of my research.

Steve has not only supported me in achieving the intellectual goals of my research but also assisted me to meet my financial constraints so that I kept myself focused on the accomplishments of my academic objectives without any disruptions. I can say without any doubt that Steve is one of the best teachers/mentors/supervisors/scholars to whom I ever met.

I would also like to sincerely thank to my secondary supervisor Jim Buchan for his valuable feedback, advice, and guidance during the different stages of my Ph.D. I acknowledge his effort which he made for the arrangements of industry data for the experimentation stage of my research.

I also want to convey my sincere acknowledgements to all 21 interviewees and teams of ASD projects, who provided me an opportunity, to conduct interviews and experimentations to gather data to develop solutions for the grave long standing issue of effort estimations. I want to express very special gratitude and recognitions to Naveed for his support and coordination during my research. I must declare that without their valuable support, cooperation, and time it would not have been possible to conduct this research successfully.

I sincerely want to express my gratitude to Dr Amjed Almas who supported me during the whole journey of my PhD. He has not only supported me morally, but also played a key role in the approval of study leaves for my PhD studies from my parent organization. I must declare that without his support it would not have been possible.

I am extremely grateful to my family members including my mother Nafeesa Perveen for her limitless prays for my success, my wife Fozia for her moral, domestic, and financial patronages, my children (Zawiyar, Hassaan, and Ayyaan) for their unconditional cooperation and to all my brothers and sisters along with their families for their love, prays and support who always encouraged me towards excellence. I want to concede very special appreciations to my elder brother Professor Tahir who helped me whenever, I asked for any support.

I would like to thank all colleagues of my parent organization for their moral supports especially Asif Ameen for looking after my personal issues during my absence, Dr Bilal Saeed Raja and Kashif for their unconditional extended assistance whenever I requested for. I would also like to thank my SERL mates, AUT to whom I shared space during my PhD studies. I am very grateful to School of Engineering, Computer and Mathematical Sciences (SECMS), AUT for awarding me PhD scholarship for the duration of 3.5 years.

At the end, I would like to thank all my friends (around the world!), including my best friends Dr Hafiz Fayyaz, Dr Dilawer, Jehan, Shahzad, Barry, Zahoor, Dr Aftab, , Kinza, and relatives for their moral support, and prays.

I feel honoured being surrounded by wonderful people like you! Thank you all …

Table of Contents

List of Tables

List of Figures

Attestation of Authorship

"I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning."

(Muhammad Tariq)

Acronyms and Abbreviations

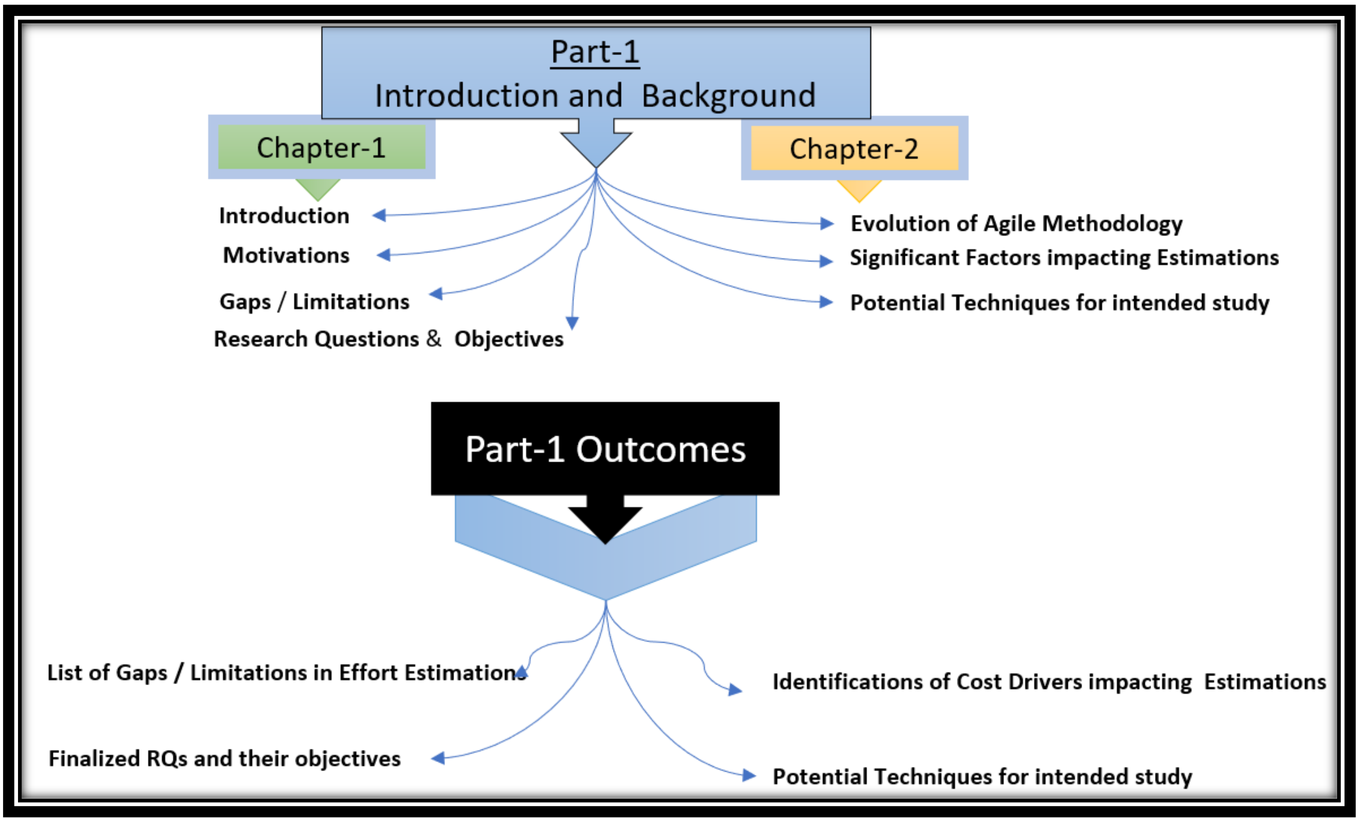
Term	Description
ABE	Analogy Based Estimation
AE	Actual Effort
AEE	Adjusted Effort Estimation
AFSPs	Actual Finished Story-Points
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
ASD	Agile Software Development
ASEE	Analogy Based Software Effort Estimation
AUSIT	Company Participated in Experiment activities of this study
AUTEC	Auckland University of Technology Ethics Committee
BFTT	Big Five Teamwork Theory
BRE	Balanced Relative Error
BREbias	Balanced Relative Error Bias
CAEA	Constructive Agile Estimation Algorithm
CEAF	COEEMO Effort Adjustment Factor
CEE	COEEMO Effort Estimation
CIO	Chief Information Officer
COCOMO	Constructive Cost Model
COEEMO	Constructive Effort Estimation Model
CSCat	COEEMO Subcategory
DIA	Daily Interpretive Analysis
DPs	Data-Points
DSR	Design Science Research
DT	Decision Tree
DTLF	Difference of Team in LF
ECF	Environmental Complexity Factor
EE	Estimated Effort
EF	Effort Factor
EJBE	Expert Judgement Based Estimation
EJBU	Expert Judgment based on Bottom-Up approach
EnvF	Environment Factor
ERP	Enterprise Resource Planning
ES	Effect Size
FP	Function Points
FPA	Function-Points Analysis
GP	Genetic Programming
HI	Human Impact
IoT	Internet of Things
IVPs	Individual Value Plots
LC	Learning Curve
LF	Learning Factor
LR	Learning Rate
MAE	Mean Absolute Error
MBRE	Mean Balanced Relative Error
MDELP	Multilayer Dilation Erosion Linear Perceptron
MEAF	Mamdani Fuzzy Inference Effort Adjustment Factor
MEE	Mamdani Fuzzy Effort Estimation
MFIS	Mamdani Fuzzy Inference System
MH	Man-Hour
ML	Machine Learning
MRE	Magnitude of Relative Error
OEI	Organization Efficiency Index

OrgT	Organization Type
PERT	Program Evaluation Review Technique
PH	Person-Hour
PO	Product Owner
PP	Planning Poker
PPBE	Planning Poker Based Estimation
PrjTeamCap	Projected Team Capacity
RCBD	Randomized Complete Block Design
RQ	Research Question
RUP	Rational Unified Process
SA	Standardized Accuracy
SERL	Software Engineering Research Laboratory
SLR	Systematic Literature Review
SM	Scrum Master
SP	Story-Point
SptCpxy	Sprint Complexity
SptNCpxy	Sprint Complexity for n number of sprint
ST	Story
StkCoord	Stakeholder Coordination
SVR	Support Vector Regression
SysCpxy	System Complexity
T1-T6	Treatment-1 to Treatment-6
TCap	Team Capability
TCF	Technical Complexity Factor
TEE	Team Effort Estimation
TESP	Total Estimated Story-Point
TExp	Team Experience
TExpert	Team Expertise
TF	Technical Factor
TM	Team Member
TPrd	Team Productivity
TR	Team Relationship
TSD	Traditional Software Development
UCP	Use case Points
US	User Story
WBS	Work Breakdown Structure
WebMO	Web Model
XL	Extra Large
XP	Extreme Programming
Xproject	One of the ongoing projects of AUSIT used for experimentation
XS	Extra Small
Yproject	One of the ongoing projects of AUSIT used for experimentation
Zproject	One of the ongoing projects of AUSIT used for experimentation

PART-1 INTRODUCTION AND BACKGROUND

This thesis comprises of six parts. The first part, contains two chapters which introduces the thesis topic and describes the gaps and limitations of agile estimation, motivating the present study. Further, it discusses evolution of the agile methodology, some of its benefits, and issues in effort estimation in an agile development context. This is followed by an overview of the most significant factors that are said to impact estimation in agile and non-agile contexts, but with a principal focus given to agile development. Furthermore, some of the potentially relevant estimation techniques are also discussed.

Detailed views of the research activities with respect to the chapters within this part are presented in the figure as given below. On the top of the diagram, the chapters covered in this part are shown along with their different research endeavours as accomplished within the individual chapters, while the outcomes of Part-1 based on its related chapters are exhibited on the bottom of the figure.



Chapter-1 Introduction

***Content Summary**: This chapter introduces effort estimation in an agile development context as the topic area on which this research is conducted. It covers the background and motivation for the research along with the research questions (RQs) and their intended outcomes. The methodology used to address these questions is also briefly introduced. At the end of this chapter, the outline/structure of the thesis is described.*

1.1 Context and Motivation

Organizations that develop software deploy considerable resources on projects in order to drive innovation and transformation (Sauser et al., 2009; Shenhar and Dvir, 2007). However, enduring evidence indicates that many software projects fail to deliver within time or budget and do not achieve business objectives (KPMG, 2017, PMI-Pulse, 2017).

Given its universal and pervasive nature, software is produced in various environments and under numerous contextual configurations (Prikladnicki et al., 2013). Its development and use inherently form part of wider 'socio-technical systems' where the human and organizational contexts play significant roles (Fuggetta & Di Nitto, 2014), along with factors including budgets, infrastructure (e.g., offices, hardware, communications, travel, training) and others (Banker, 1987). The success of a software project is consequently multi-faceted – common considerations are adherence to scope, time, and cost goals and/or stakeholders' satisfaction (Schwalbe, 2014; Dragicevic et al., 2017). As such, estimates of the personnel effort likely to be required to complete a software project are central for forecasting size, cost or schedule (Jørgensen and Shepperd 2007; Stutzke 2006), and these may need to be mediated by a wide range of potentially influential factors, including (Wegner & Ruhe, 2008; Arora et al., 2018):

a. product factors such as precedence, reliability, reuse
b. process factors such as architecture risk resolution, process maturity
c. development environment factors such as the use of particular practices or tools
d. human factors such as corporate culture, team culture, team capabilities and experience
e. physical factors such as the workplace, telecommunication facilities.

Accurate estimation of effort has long been held to be important in planning a software development project and it remains an ongoing challenge in the field of software engineering (Souza & Aquino, 2014; Dragicevic et al., 2017, Arora et al., 2018; Shams et al., 2019, Fernández-Diego et al., 2020). As such, it has received extensive attention in research and practice, yet in many instances software project managers still fail to arrive at correct estimates (Tariq & Shobha, 2011). Several reviews of such studies have shown the disbursement of 30% to 40% more effort than estimated (Molokken-Ostvold & Jorgenson, 2003) and 20% to 30% average error on effort estimation (Grapenthin et al., 2014; Abrahamsson et al., 2011; Kang et al., 2010). Project bids and work schedules are developed based on some form of effort projection; and the accuracy of that projection is important in terms of enabling personnel to manage financial issues and to monitor a project's development and likely time-to-market (Suharjito et al., 2016). Poor estimation of software development effort can lead to detrimental outcomes such as schedule and budget overruns of the project and could in some cases ultimately threaten organizational competitiveness and viability (Kocaguneli, 2012: Minku and Yao, 2017). A collaborative survey conducted by McKinsey and Oxford University of more than 54,000 software projects reported in 2013 that 45% of projects ran over budget, 7% fell behind schedule, and 56% provided less functionality than expected. When these effects were considered overall, 17% of projects were said to threaten their organizations' existence (Bloch et al. 2013). Further, a business review conducted by Flyvbjerg & Budzier in 2011, revealed 27% average cost overrun after analysing a sample of 1,471 IT projects. One out of every 6 projects in the sample had shown 200% and 70% overruns, on average, in terms of the budget/cost and schedule, respectively.

In light of the above, numerous estimation techniques have been proposed for use in the software industry (Jørgenson & Shepperd, 2007; Sommerville, 2007), most of them formulated in the 1980s, and to varying degrees of success. Almost all have been founded on the prior estimation of the scale, and perhaps the complexity, of the project to be undertaken, based on some granular breakdown of the functionality to be delivered. While none has found universal applicability some of these estimation methods have proved to be effective in specific development contexts, particularly over the 1990s. However, many software teams have moved from what we term here as traditional software development (TSD) (also known as planned or plan-driven and typified by the waterfall method and its variants) to agile software development (ASD), due to the latter's emphasis on fast release cycles, frequent delivery of value-added features to customers, and close collaboration between the product owner and the development team (Hamdan et. al., 2013). Whether the estimation methods established prior to the advent and adoption of agile approaches have any relevance or value in ASD environments remains unknown. The most widely known examples of sizing and estimation techniques used for projects undertaken in TSD environments are COCOMO (Briand et al., 1999; Zhao et al., 2003; Jodpimai et al., 2009), Function Point Analysis (Albrecht, 1979; Wu, 1997; Jun et al., 2008), Top-Down & Bottom-Up approaches (Pengelly, 1995; Wu, 1997), Use Case Points (UCP) Estimation (Diev, 2006; Ochodek et al., 2010; Collaris & Dekker, 2011), and expert estimation (Kumar et al., 2016), whereas the most common techniques used for sizing and forecasting (per sprint) in an ASD environment are Planning Poker, T-Shirt Sizing, and Relative Mass Valuation (Green, 2014).

1.2 Problem Statement

ASD is not a single method; rather, it represents a class of contemporary methods of software development. Although variants of agile methods have been used in multiple forms since the late 1950's, it is only relatively recently that agile software development has received global recognition and adoption (Hamdan et. al., 2013).

ASD has fundamentally changed the software life cycle. The emphasis of ASD is on the iterative and continuous delivery of working software features, replacing the formerly large-scale (and often document-heavy) phases of TSD including Vision, Analysis, Design, Build and Test. Moreover, just as ASD introduced the disruptive concept of agility in every step of the software life cycle, it has also introduced a range of new and unique methods for sizing and cost estimation. Rather than being based on a breakdown of the functionality to be delivered by the project to the customer, the breakdown in ASD typically centres on an increment of activity for the development team – in some ASD methods this increment is referred to as a sprint.

For instance, some executions of ASD include a group estimation methodology in which the team of developers sits with the product backlog manager, and they estimate the relative size of each story, i.e., a feature of the software to be developed comprising a written description, discussions, and tests (Cohn, 2004), often in unit-less story points. The number of story points completed in a sprint is called team's velocity. Over time, and assuming stability over a range of factors, the anticipated velocity of a team per sprint may be calculated based on actual development outcomes. Such data can inform those charged with the planning and delivery of features and releases in terms of management and scheduling.

As stated above, while in ASD a range of techniques are being utilized to forecast the story points (used here to refer to the range of feature sizing methods) that would make up an increment, evidence suggests that the most popular are Planning Poker, T-Shirt Sizing, and Relative Mass Valuation (Green, 2014). The existing methodologies of ASD to estimate story points and inform the scheduling of a software project are intentionally simple and can produce results that are close to the actual values of size and effort for a project. However, the trade-off for that simplicity is that there are many gaps in the prevailing methodology; it is asserted here that their consideration may help to improve the estimations further:

a. Estimates of the work required/expected in a sprint are dependent on team experience in software development and/or effort estimation (Nguyen-Cong & De Tran-Cao, 2013), and on how effectively a product owner explains a story, making the process vulnerable to aspects of team composition and experience.

b. Offshore software development is increasingly common due to its access to cheaper labour costs (Islam, et al., 2009) whereas there is an assumption in ASD that a team meets frequently and relies on verbal rather than written material; this may compromise shared knowledge and hence may affect project schedules (Hamdan et. al., 2013). Geographical, temporal, and cultural differences are some of the global barriers which should be considered as additional cost and effort drivers while estimating effort for distributed ASD (Usman & Britto, 2016).

c. In addition, extensive travel may be needed when a project is developed at different sites, and specialist tools and technologies may need to be employed, which may both contribute to further project costs. However, this specific aspect is not directly related to the effort cost.

d. Existing estimation and tracking methods for agile software development cannot be easily related to time durations (relevant to scheduling) because story points are used to estimate the size and complexity of a story rather than the amount of work required, and its completion time is velocity-dependent where velocity may differ from team to team (Popli & Chuhan, 2013; Dragicevic et al., 2017).

e. In TSD senior project managers are frequently responsible for the estimations of size and effort in projects and in general they are not contributing substantially towards the overhead of the project because they are also involved in other miscellaneous activities of management in addition to the project activities. In ASD, however, the team – which may consist of 6-8 members – is responsible for estimations of all of the increments in a project. While the specifics will vary it is not unreasonable to assume that a one-month increment may require a time-boxed period of 8 hours from the development team (Schwaber & Sutherland, 2016) to carry out the sizing and forecasting exercise. This is a non-trivial cost that, if it is to be expended on a regular basis, should be offset by the benefits of having the team involved. This can be further explained by an example: let's assume one team of 8 members is working on a single agile software development project made up of 4 increments. If the team utilizes time-boxed 8 hours to derive size estimates per increment, then they would need 32 hours exclusively for estimating effort. The approximate cost of that activity over the project would be NZ$32,000 (based on the hourly rate of an experienced IT professional with multiple areas of expertise (Recruit IT, 2015).

f. The techniques used in agile forecasting of component size (e.g., user story size) are based on expert judgement which tends to be the most popular method applied (Jørgensen, 2005; Usman et al., 2014; Popli & Chuhan, 2014), probably due to its simplicity and flexibility. However, limited substantial empirical evidence is available to support this method over other methods of effort estimation (Jørgensen, 2005). Furthermore, a more recent study in a medium-sized organization (focused on bespoke software development for domestic and international markets) showed cost estimation based on unstructured expert judgement to be unreliable during the bid phase of the project (Faria & Miranda, 2012).

g. Techniques including Function Point Analysis, UCP estimation, COCOMO and the like, which have been used successfully in TSD to estimate the size and effort of software projects, indicate that many factors can affect the cost of a project, including the desired reliability and usability of the software, team experience, team culture, corporate culture, workplace factors, use of modern tools/practices, offshore development, and others; how such factors are or could be accommodated in ASD sizing and estimation is not clear.

h. ASD embraces the concepts of self-management and self-organization in a team. Such an approach assumes that all members are dutiful, responsible and honest in their dedication to a project. This is potentially an optimistic assumption. It is a matter of fact that some people are more capable, hardworking and dedicated compared to others, and/or more suited to particular roles and responsibilities. Furthermore, ASD supports team work more so than that of individuals. It is contended here that any compensation to the team members may have a good impact on the overall velocity of a team and may also create a sense of positive competition among the team members. Consideration of such factors may impact on the overall velocity of a team and have flow-on effects for project scheduling.

i. ASD suggests close collaboration between development personnel and the Product Owner (or similar) – being the person responsible for ensuring the right features are prioritised and delivered – and user requirements are clarified through verbal, not written, communications (Marchesi & Succi, 2002). The same level of collaboration is also expected between the team and other stakeholders at the time of project estimation. Non-availability of stakeholders or their inadequate input may therefore affect project estimates (Hamdan et. al., 2013).

j. The accuracy levels for the estimates derived using various techniques in ASD are shown in Table 1.1. These levels of accuracy are drawn from a 2018 Systematic Literature Review of the state of the art on estimating effort in ASD post-2014 (Dantas et al., 2018). Results in the table reveal that there remains substantial room for improvement (the weaknesses of the measures used notwithstanding). Further, the average error of various ASD time predictions has been found to be about 20-30% (Halkjelsvik & Jorgensen, 2018).

Technique	Accuracy Achieved %
Planning Poker	*MMRE: 16-61
	**BRE: 35
Expert Judgement	MMRE: 22
	***PRED (25): 73.13
	BRE: 38.7 – 78.5

* Mean Magnitude of Relative Error – mostly used metrics to measure accuracy
** Balanced measure of Relative Error
*** Prediction Accuracy

Table 1. 1 Accuracy Achieved by Techniques (Dantas et al., 2018)

Taking into consideration the many issues just described it may be possible to improve the accuracy of ASD forecasts for size, effort, cost and schedule. This study therefore investigates how the above identified gaps and opportunities may be assessed, and addressed, in terms of their effects on ASD forecasts. This intent is captured in the research questions and objectives that follow.

1.3 Research Questions and Objectives

There are four research questions that will be answered through the proposed study:

RQ-1: What are the gaps and limitations in contemporary estimation and scheduling techniques when applied in agile development contexts?

Objective: To study prevailing estimation techniques employed (in ASD) to forecast the size, human effort and schedule of a software project and to identify any weaknesses and limitations that may constrain the delivery of products within time or budget constraints.

RQ-2: What is the minimum data required for 'accurate-enough' agile estimation and scheduling? Which factors are influential in agile estimation and so should be measured, and which can be ignored?

Objectives: To identify and classify factors based on their relevance, significance and utility for the forecasting of size, effort, cost and scheduling of projects as indicated by academia and practice in agile and traditional software development. Further, to include only those factors that are relevant and may impact/improve the forecasting accuracy of size, effort, cost and scheduling of the projects in TSD and ASD both contexts, but the principal focus is given to the agile development context. The core aim is to improve accuracy by considering factors that are (most) relevant/significant in terms of their impact on development effort and cost.

RQ-3: How can the significant factors as discovered in RQ-2 be effectively calibrated to build a dynamic model of estimation, on its own and in combination with other subjective assessments, to further improve estimation accuracy?

Objectives: Once significant/relevant factors as identified in RQ-2 are quantified, an algorithmic model is developed to improve the utility of forecasts of size, effort, cost and schedule for ASD projects. The proposed values of these factors are further calibrated to accommodate their impacts on accuracy of forecasts of size, effort, cost and schedule for ASD projects with and without combinations of other subjective assessments/techniques as considered in RQ-4.

RQ-4: What is the effect of augmenting expert judgement, as applied in existing estimation methods for ASD projects, with the PERT and expert judgment with Bottom-Up approaches in terms of utility on forecasts of size, effort, cost and schedule for ASD projects?

Objectives: Evidence has shown that expert judgement is one of the dominating techniques in current methods of estimation for ASD projects. This prevailing technique is planned to be extended by two other well established methods - Bottom-Up and PERT - and their utility assessed in terms of accuracy of forecasts of size, effort, cost and schedule for ASD projects.

Note: Size refers to the (relative) scale of a feature, task or other unit of development estimated using various techniques, e.g., Planning Poker (using the Fibonacci sequence), Relative Mass Valuation, T-shirt sizing, and like; Effort means the person-units of time required by the team to implement a development unit; Cost is typically the product of person-hours by their salary rates; and Schedule refers to duration over time and numbers of iterations.

1.4 Research Methodology

In order to answer the four research questions stated above the DSR (Design Science Research) methodology as per the guidance provided by Offermann et al. (2009) is employed. DSR undergone continuous refinement over more than 40 years (Cross, 2007) and gained positive recognition in the discipline of Information Systems (Nunamaker et al., 2013). Numerous researchers have not only informed the multiple frameworks of DSR but also shared their benefits while employing DSR as a methodology in conducting and reporting research in the IS domain (Gregor and Hevner, 2013; Nunamaker and Briggs, 2011, Peffers et al.,2007; Hevner et al., 2004; Nunamaker et al., 1991).

The DSR process as proposed by Offermann et al. (2009) is shown in Figure 1.1. This process suggests three core phases, i.e., Problem identification, Solution design and Evaluation which are further divided into various steps as shown.

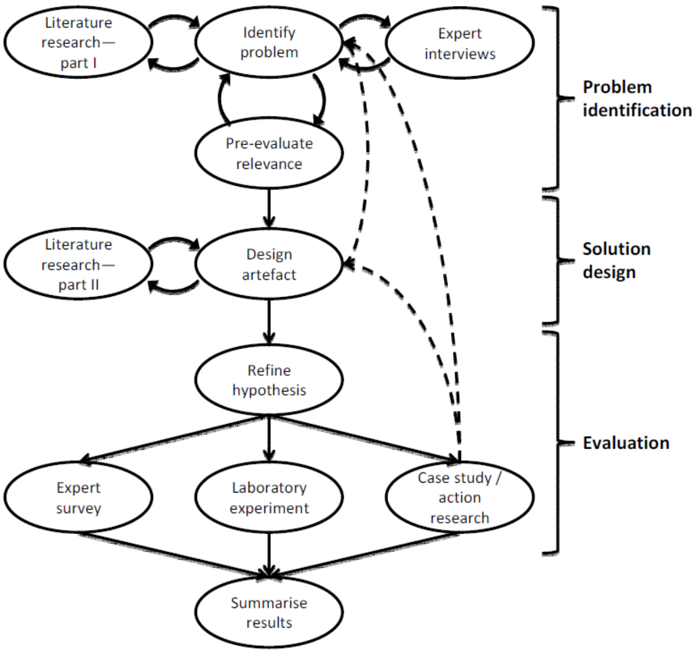


Figure 1. 1 DSR Process as proposed by (Offermann et al., 2009)

Mapping of the DSR process with this research is presented in Figures 1.2 to 1.4, starting from the phase of problem identification, and concluding with the summary of results, as suggested by Offermann et al., (2009). Further, these three phases are interconnected which means that after the completion of each phase the previous phases are revisited for the purpose of improvements, modifications, revalidation and revaluation. A detailed discussion of the selected research methodology is reported in Chapter 5.

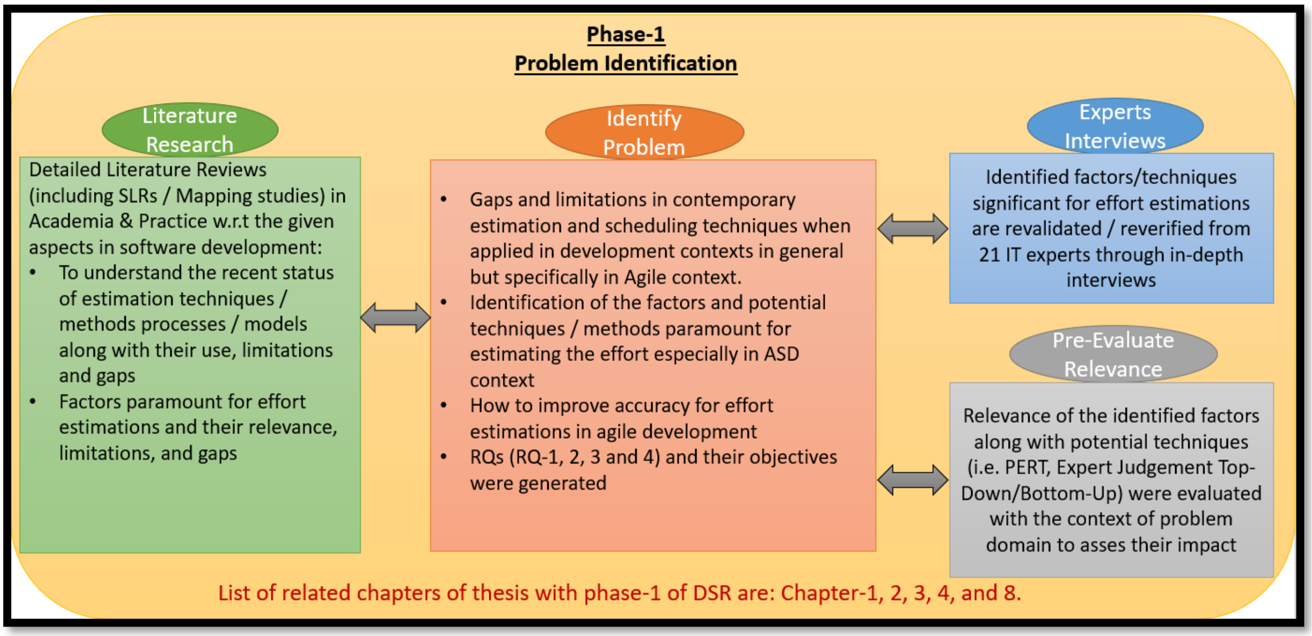


Figure 1. 2 Mapping of envisaged study with DSR (Problem Identification Phase)

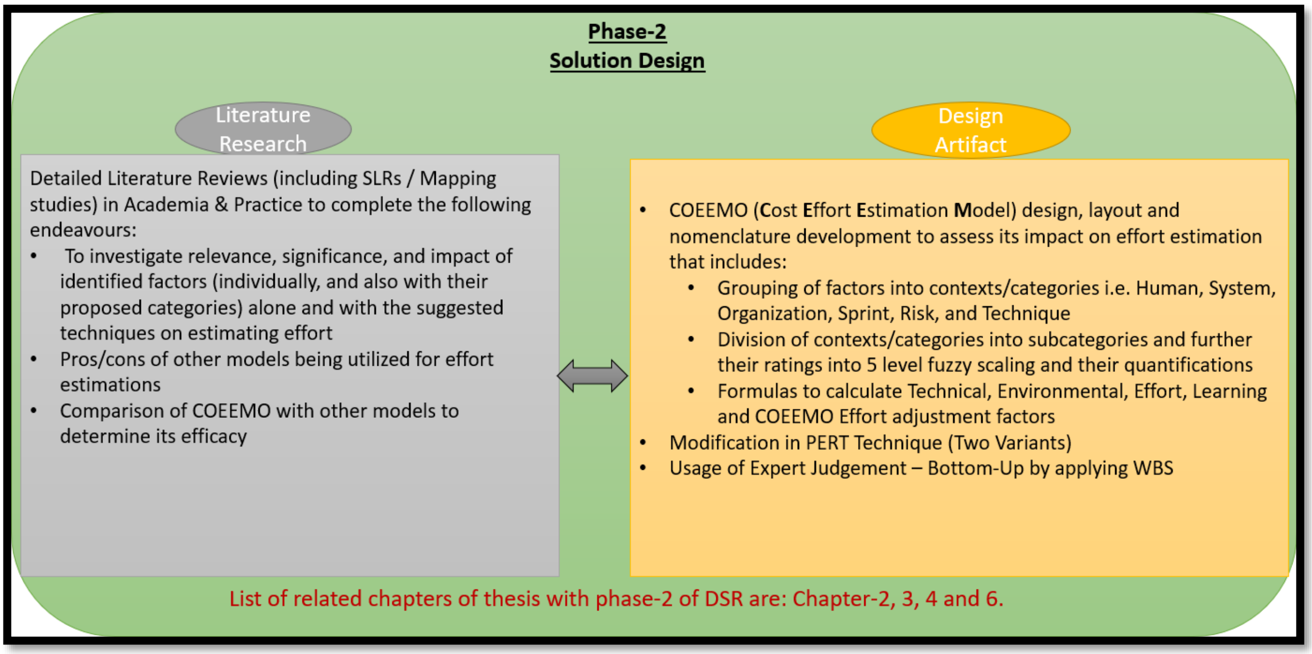


Figure 1. 3 Mapping of envisaged study with DSR (Solution Design Phase)

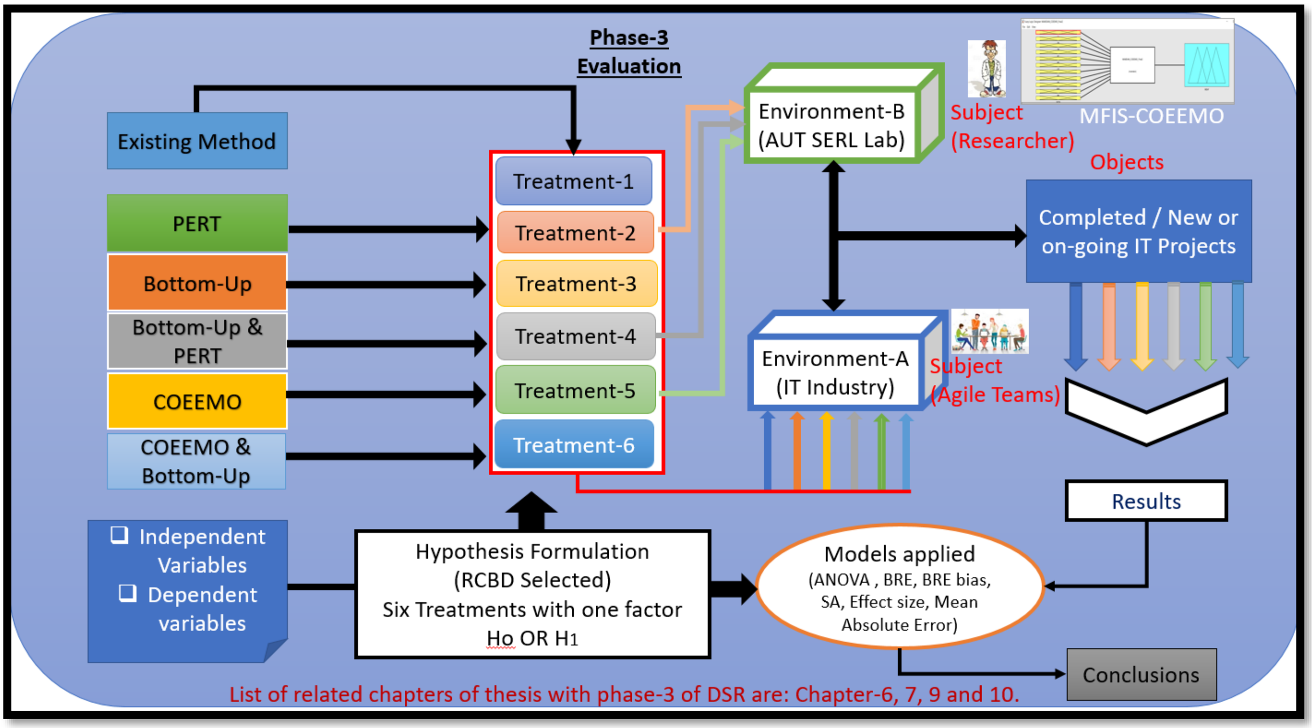


Figure 1. 4 Mapping of envisaged study with DSR (Evaluation Phase)

1.5 Thesis Layout / Structure

The remainder of this thesis is structured as shown in Figure 1.5. As per the demonstrated layout, this thesis is divided into six parts. Each part includes a group of chapters that reflect a logical flow of activities as conducted in this study. Part-wise chapters' details are given as follows:

- Part-1 (Chapter-1 & 2): This introduces the thesis topic and describes the gaps and limitations of agile estimation, motivating the present study. Further, it discusses evolution of the agile methodology, some of its benefits, and issues in effort estimation in an agile development context. This is followed by an overview of the most significant factors that are said to impact estimation in agile and non-agile contexts, but with a principal focus given to agile development. Furthermore, some of the potentially relevant estimation techniques are also discussed.

- Part-2 (Chapter-3 & 4): This part reviews related studies conducted in the area of the proposed research and presents comparisons of the weaknesses and strengths of some of the current and traditional methods/techniques/models used for effort estimation. Further, the theoretical and functional frameworks of the proposed model is discussed in this part.
- Part-3 (Chapter-5, 6, & 7): This part describes the detailed research methodologies that are adopted in this study along with the ethics approval from AUTEC. This part also discusses detailed plans of the experimentation and its execution.
- Part-4 (Chapter-8 & 9): This part reports the conduct and results of the interviews conducted with 21 IT practitioners. This covers details of the instruments used for interviews, and the methods used in the execution of the interviews, including collection and categorization of the data, their interpretation, analysis and synthesis. Moreover, this part exhibits descriptive statistics of the experiments conducted in real-industry settings along with evaluations of the hypotheses and accuracy comparisons of different treatments.
- Part-5 (Chapter-10 & 11): This part encompasses the validation of the proposed model based on three perspectives: i.e., empirical, field trials, and laboratory. Further, this reports the discussions and findings of the study.
- Part-6 (Chapter-12, References, and Appendices): This part includes a summary of the study, along with the core contributions of this research. Further, this part also discusses the implications, future work and limitations of the study. Furthermore, the lists of references and appendices are presented.

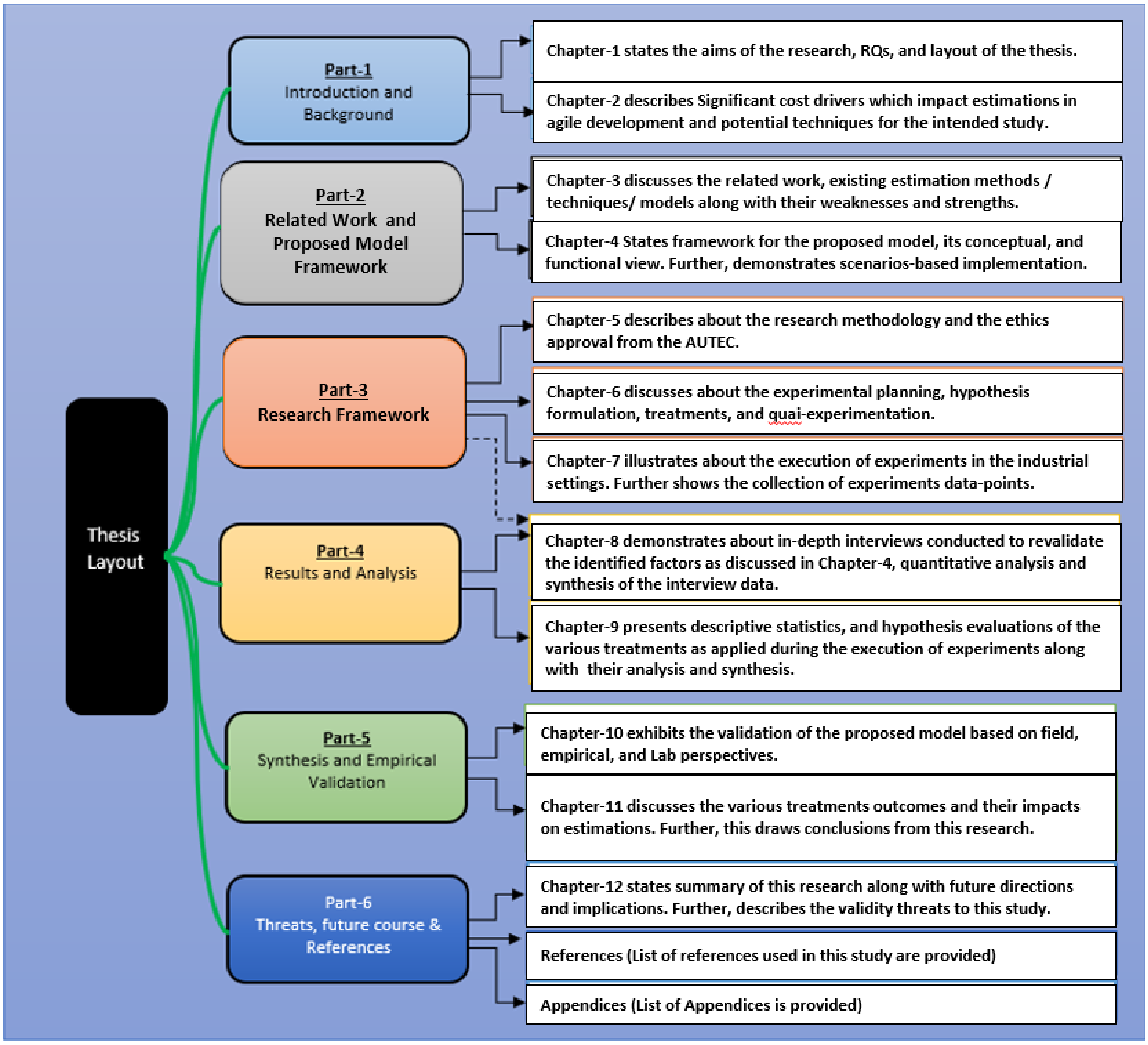


Figure 1. 5 Thesis Structure / Layout 1

1.6 Mapping of Research Activities and Outcomes based on Thesis Parts and Chapters

Detailed views of the research activities with respect to the chapters within the related parts are presented in Figure 1.6. On the left side of the diagram all parts of thesis and chapters are shown along with their different research endeavours as accomplished within the individual chapters, while the outcomes of each individual part based on their related group of chapters are exhibited on the right side of the figure.

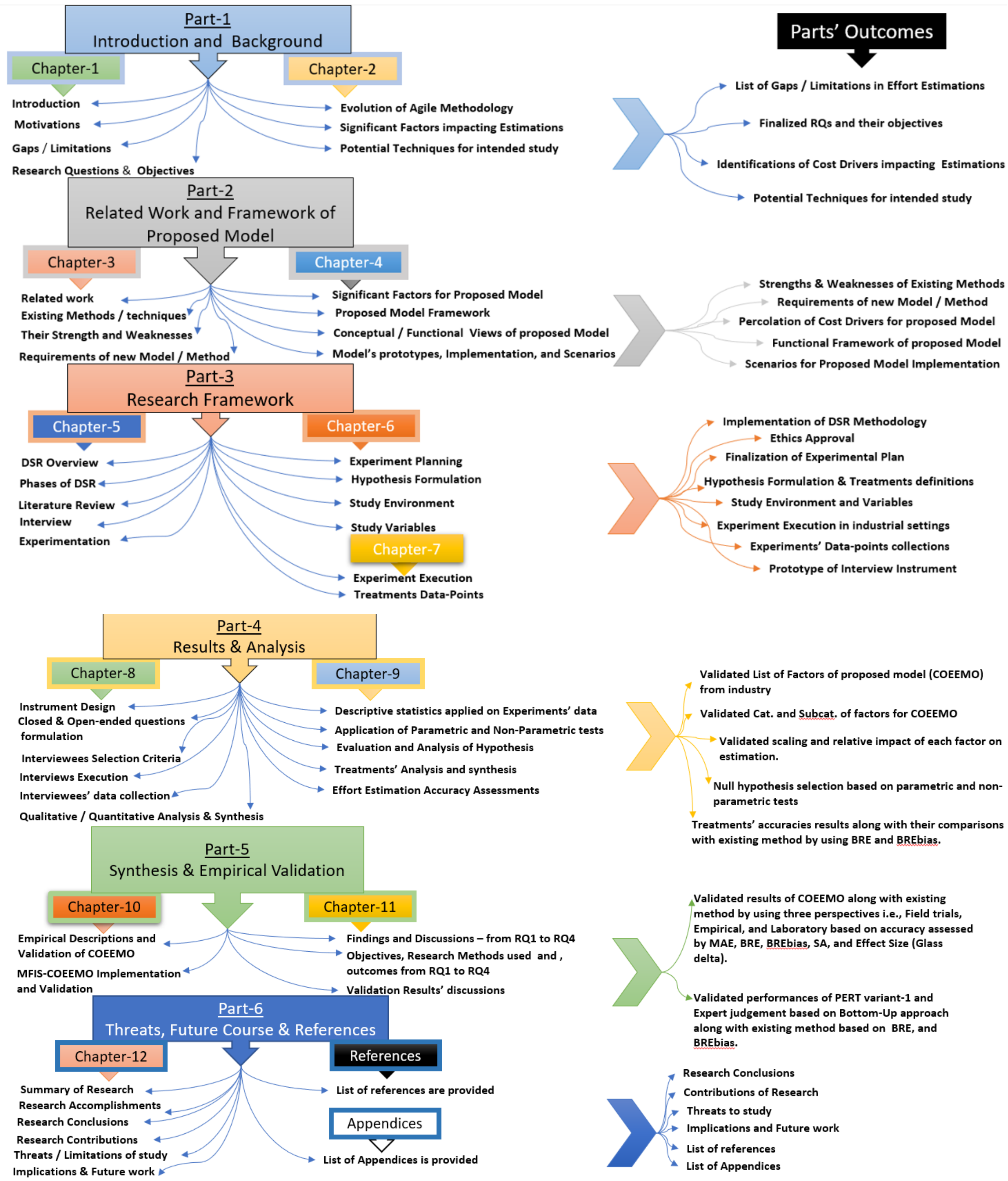


Figure 1.6 Mapping of Research Activities and outcomes based on Thesis Parts and Chapters

1.7 Chapter Review

In this chapter, background literature regarding effort estimation in agile and non-agile contexts, gaps / limitations in current agile estimation methodology, research questions along with objectives for the intended study, research methodology selection for the envisaged study, and layout / structure of the thesis were covered.

In next chapter, the materials regarding agile development, factors' significance and their relevance, potential techniques for the intended study, agile team roles and responsibilities, description of the existing methodology, and learning curves are discussed.

Chapter-2 Background

***Content Summary**: This chapter provides a brief detail about evolution of the agile methodology, some of its benefits, and issues in effort estimation in an agile development context. Further, it discusses factors that could be significant for estimating effort in agile development. Finally, the chapter describes some of techniques significant to the intended study.*

2.1 Agile Software Development (ASD)

Software Development (SD) encompasses all the interconnected endeavours that bring a software project from conception to implementation. IBM Research defines SD as "…a set of computer science activities dedicated to the process of creating, designing, deploying and supporting software." Numerous methodologies have been utilized in the development of software, among them code-and-fix, evolutionary (McCracken and Jackson, 1982), transform (Balzer et al., 1983), waterfall (Royce, 1987), spiral (Boehm, 1988), and others. Among those methods the iterative waterfall model attained extensive popularity and was an effective development method for many years. It had resolved many issues and limitations of earlier methods and was utilized particularly widely for the development of large-scale systems (Misra et al., 2012).

However, all of these methods including the waterfall (normally referred to as traditional software development method (TSD)) were 'process-heavy', in that they relied on prescribed steps and extensive documentation (e.g., plans and Gantt charts, detailed requirements specifications, design blueprints, and others). As such they suffered a range of limitations – it was difficult to accommodate change in client requirements, the methods were complex to learn and use, and were considered to be labour-intensive (Tolvanen, 1998; Kalermo and Rissanen, 2002; Awad, 2005). Consequently, to overcome TSD shortcomings and limitations (Dingsøyr, et al., 2010), agile software development methods were formally introduced in 2001 by a group of seventeen practitioners (Beck et al., 2001) via an agile manifesto.

This manifesto was based on four basic guidelines which gave preference to stakeholders' interactions, working software, customer collaboration, and response to change, over processes and tools, detailed documentation, contract negotiation, and comprehensive planning (Agile Manifesto, 2001). Variants of agile methods (Dingsøyr, et al., 2010) had been used in multiple forms since the late 1960's (Merisalo-Rantanen et al., 2005) including: Scrum (Schwaber 1995; Schwaber and Beedle 2002), Extreme programming (Beck 1999; Beck 2000), Feature-driven development (Coad et al. 2000; Palmer and Felsing 2002), Crystal (Cockburn 1998; Cockburn 2000b; Cockburn 2002), Adaptive software development (Highsmith 2000), Radical software development (Bayer and Highsmith 1994), Agile modelling (Ambler 2002), Dynamic systems development method (DSDM Consortium 1997; Stapleton 1997), and others. However, it is only relatively recently that agile software development has received global recognition and adoption (Hamdan et. al., 2013). The evolutionary mapping of the various agile methods is shown in Figure 2.1, which not only demonstrates their interrelatedness but also presents their intellectual source (displayed by arrow lines). Further, links to previous studies vis-à-vis agile methods are also exhibited in Figure 2.1 to illustrate which methods and/or their developers supported the publication of the agile manifesto (Dingsøyr, et al., 2010).

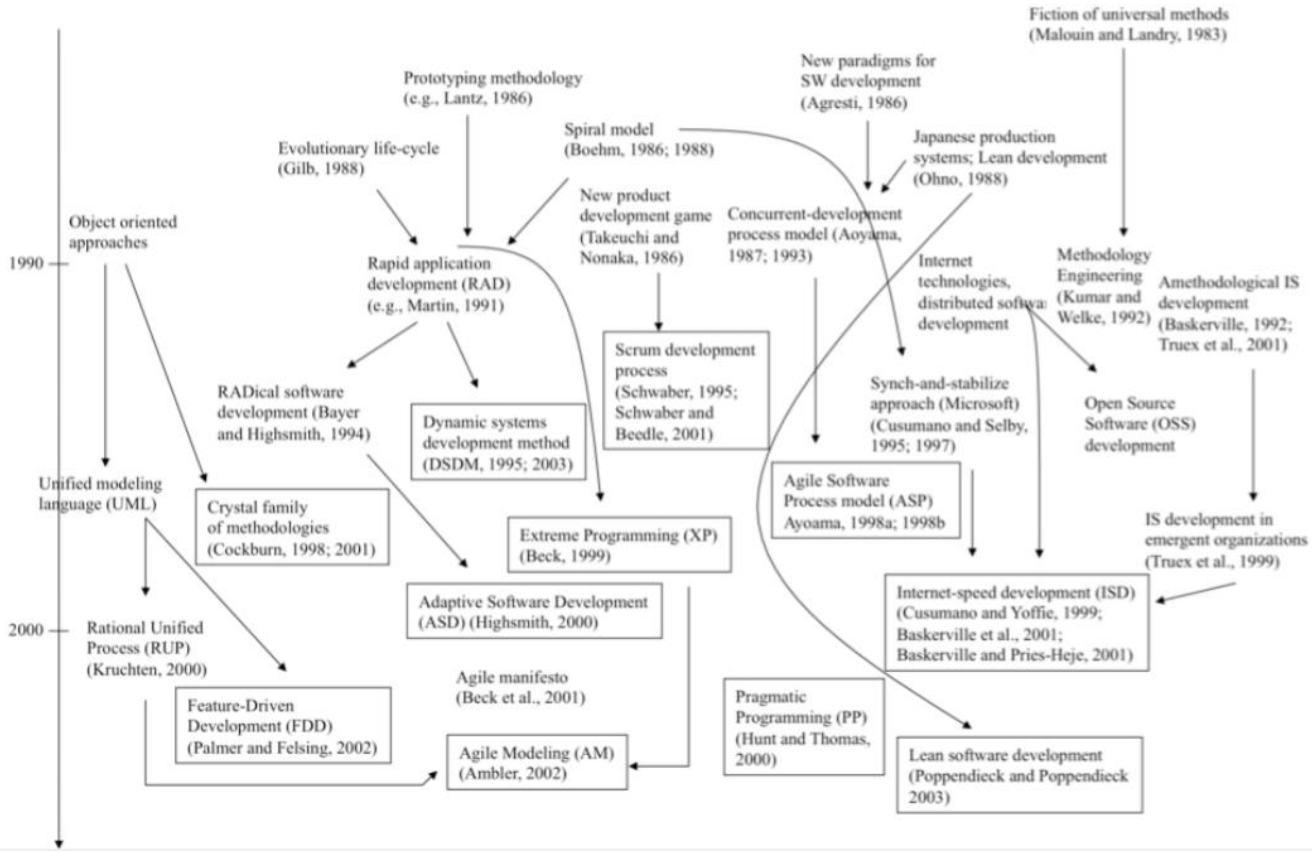


Figure 2. 1 Evolutionary mapping of Agile Software Methods (Dingsøyr, et al., 2010)

Agile software development methodologies have since gained extensive popularity worldwide (Haines at al., 2017). A 2016 report demonstrated that various aspects of agile had been embraced by approximately 95% companies while 41% organizations were using agile as their core system development endeavour (Rago, 2020). Numerous studies and reports have presented practitioner trends and pursuits in agile development along with related risks and organizational challenges (Conboy, et al., 2011; Cram & Newell, 2016; Janes & Succi, 2012).

2.2 Effort Estimation in ASD

Effort estimation in software development provides forecasts of the effort that is likely to be required to accomplish a project and/or a task (Roark, 2020). The approximation of this effort not only facilitates a team or an organization to assess the cost required to retain a workforce to complete the project but it can also underpin the delivery schedule through to the handover of the project to the client (Bilgaiyan et al., 2017). A software project’s success is based on numerous parameters but accurate effort estimation (Bilgaiyan et al., 2017) is one of the core factors among them, because it assists stakeholders in planning, budgeting, and scheduling the project (Roark, 2020).

Although there are numerous success stories for ASD, the fact remains that the IT industry still has problems delivering complex software development projects on time, resulting in budgeting and scheduling overruns. A recent SLR presents an updated review of the state of the art on estimating effort in ASD since 2014 (Dantas et al., 2018) and it states that even the best level of accuracy is not sufficient for good practice. Further, another recent study states that the time prediction's average error remains stubbornly at about 20-30% (Halkjelsvik & Jorgensen, 2018). Several factors contribute to inaccurate effort estimates which include but are not limited to the following: project size, estimator experience, environment dependency, developer experience, team skill, communication with client, complexity and others (Sehra et al., 2016 & 2017; Tanveer, et al., 2017; Usman et al., 2017). More details about the factors that may impact accuracy for estimating effort, and so are potentially of interest in this study, are covered in the next section.

2.3 Potentially Significant Factors for this Study

Estimates for a project may need to be mediated by a wide range of potentially influential factors, including product, process, development environment, soft and physical factors (Wegner & Ruhe, 1980).

The initial conceptual model of the estimation process at the centre of this study is based on some of the many parameters, and potential predictors (given in tables 2.1, 2.2, 2.3, 2.4, 2.5 and figure 2.3 respectively), as considered significant in previous survey studies in general (Usman et al. 2014 & 2015; Ahimbisibwe et al., 2015; Tanveer et al, 2017; Dantas et al., 2018; Carbonera et al., 2018) and as relevant to the agile development context in particular; some additional parameters based on the twelve principles in the Agile Manifesto; other academic and practice evidence; and the researcher's own domain knowledge and software industry experience (described in the Preface); all as shown in Figure 2.2.

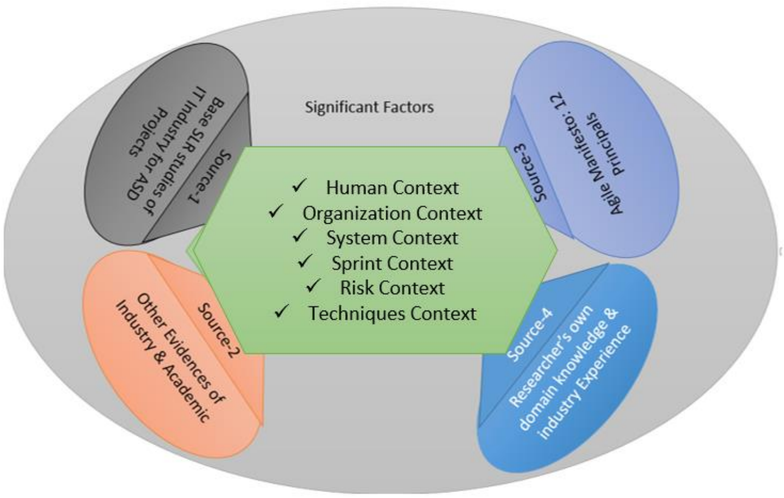


Figure 2. 2 Mapping of Potentially Significant Factors

Ahimbisibwe et al. (2015) reported 37 critical success factors (CSFs) (given in Table 2.1 and shown in Figure 2.4) identified through a systematic literature review of 148 previous articles, included because of their reported significant influence on the success of software development projects. The literature is unclear, however, regarding their *individual* roles affecting project success. They are grouped into four major CSF categories (organizational, team, customer and project factors) and ranked based on the number of occurrences in the literature, set alongside their associated development methodologies.

Category	Critical Software factors (CSFs[1])	Ranking based on number of occurrences in the considered literature (from Table III) - Ahimbisibwe et al., 2015		
		Overall	Agile Projects	Traditional plan-based projects
Organizational factors	Top-level management support	1	13	4
	Organizational culture	4	2	10
	Level of project planning	5	25	1
	Leadership	6	12	9
	Vision and mission	7	27	2
	Monitoring and controlling	8	26	3
	Change management skills	9	6	13
Team factors	Project team commitment	3	10	7
	Internal project communication	10	5	16
	Project team empowerment	15	11	21
	Project team's composition	16	7	22
	Project team's expertise with the task	19	16	19
	Project team's general expertise	20	24	5
	Lack of development team skill	21	19	17
	Project team's experience with SDM	25	23	8
Customer factors	User participation	2	3	6
	User support	11	18	12
	Customer training and education	17	14	18
	Customer (client) experience	18	8	23
	Lack of end user experience	27	17	27
Project factors	Technological uncertainty	12	1	24
	Development methodologies	13	15	14
	Project complexity	14	4	25
	Urgency	22	21	15
	Relative project size	23	20	20
	Specifications changes	24	9	26
	Project criticality	26	22	11

CSFs1 – Critical success factors categories and ranking based on methodologies.

Table 2. 1 Critical Success Factors for Sofwtare Development Projects (Ahimbisibwe et al., 2015)

The potentially significant factors considered in this study are reviewed in the following subsections under the main categories as depicted in Figure 2.2.

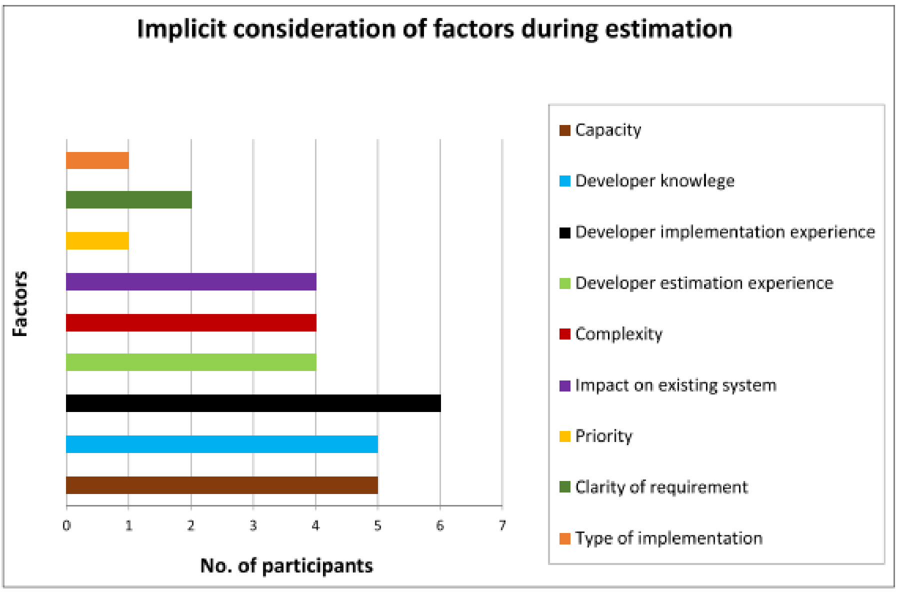


Figure 2. 3 Factors considered implicitly by practitioners during effort estimation (Tanveer et al., 2017)

2.4 Human Context

2.4.1 Relevant Evidence

Numerous software projects have failed to be delivered within time and budget projections and many have also not achieved their associated business objectives (PMI-Pulse, 2017, KPMG, 2017; Ahimbisibwe et al., 2017). However, it has been asserted that this rate of failure could be reduced if organizations and teams were to adopt a more suitable project management methodology according to various team, client, organizational and other project factors (Ahimbisibwe et al., 2017).

In a study, Pereira et al. (2017) identified and assessed aspects of maturity in software engineering teams by using a framework of proficiency for capabilities across eight categories: intra-team relationships, intra-team communication, team processes, extra-team relationships, extra-team communications, relating with Product Owner, individual characteristics and team characteristics. These categories as stated in the aforementioned study overlap to a degree with those presented by Fontana et al. (2015) for an agile context, in which their study specifies seven outcome categories as a means of identifying ASD team maturity: practice learning, team-conduct; pace of deliveries, features disclosures, software product, customer relationship and organizational support.

Cost Drivers	Academia	Practice
	No of occurrences (n=25)	No of Responses (n=60)
Team skills / expertise	3	43
Task size	2	25
Team prior experience	2	39
Project Domain	2	25
Customer Communication	6	14
Others (Keywords in user story, design metrics, use case elements)	6	10
Multiple sites	2	05
Performance & Configuration Requirements	2	Not Stated
Volume of data transactions, Ops. ease for users	2	
Security	2	
Not Used	9	

Table 2. 2 Cost Factors for Estimation in Agile Context (Usman et al., 2014 & 2015)

A further earlier study on agile and plan-driven approaches asserted that most of the critical success factors for software projects are human-related factors, although other factors related to methodologies, technical and management techniques have their own importance. Further, this same study specified the particular significance of five human-related factors: staffing, culture, values, communications, and expectations management (Boehm & Turner, 2009).

Acquisition of knowledge is significant to build effective teams for the purpose of producing high-quality software products successfully and proficiently (Richardson et al., 2012; Yilmaz et al., 2016). Inexperienced and unskilled team members, on the other hand, may be responsible for inaccurate estimates (Usman & Britto, 2016; Grimstad & Jorgensen, 2005). According to Pressman and Maxim (2014), such human aspects have fundamental significance for the success of a software project.

Addressing a related topic, it has been observed more generally that software experts or senior software engineers are typically responsible for the estimation of software size and cost (Gandomani, Wei & Binhamid, 2014). Their experience and judgement are most often only helpful in the organization where they have worked, however, because the adequacy of such predictions strongly depends on team experience, project complexity, organizational culture, and so on (Gandomani, Wei & Binhamid, 2014).

Developer professional experience was considered one of the parameters in a suggested scheduling model for software projects within agile frameworks (Jahr, 2014). In another study Caballero & Calvo-Manzano (2012) concluded that team prior experience and excellent technical skills are influential in terms of productivity in a SCRUM environment.

Research conducted by Sedelmaier & Landes (2014) proposed a body of skills for software engineering that includes competencies for professional collaboration with others, communicative competencies, structuring one's own way of working, motivation to contribute one's share, personal competencies that reflect developer capabilities, a capability to understand complex systems, processes and their relationships, and some additional competencies such as a willingness to accept responsibility for others and shared projects.

A study reported by Augen (2006) found that more accurate group estimation results (via the Planning Poker Technique) were achieved when the team had previous experience with similar tasks; whereas it was likely that less accurate results would be achieved from an inexperienced team even working with similar tasks. This outcome was reiterated in another study: in one evaluation an experienced team estimating effort by using Planning Poker produced a more accurate estimate (0.5 MMRE vs 0.70) than that produced by a non-experienced team; in a second evaluation for large task estimation the difference was more marked (MMRE 0.21 vs 0.87) (Nguyen-Cong, Tho & Tran-Cao, 2013).

There is also longstanding agreement among numerous authors on the significance of social and interpersonal abilities (e.g., D.P. Flanagan.et.al 2012, L. C. Buffardi, 2000, E. A. Fleishman.et.al 1984 & 1968). The software development process is dependent on the situational aspects of the individual software development environment. Such characteristics include the nature of the application under progress, requirements volatility, team size, and personnel experience (Clarke & O'Connor, 2012).

S.#	Category of reasons for inaccurate estimates in ASD from practice	Responses (Total n= 25)
		Actual
1	Requirements' related issues (Missing and changing Requirements, Poor user stories, Overlooking NFRs etc.)	9
2	Project management issues (Poor change control, SM not guiding the team, Scope creep)	7
3	Team related issues (Inexperience, knowledge sharing problem in team, Unskilled team members)	5
4	Over-optimism (Best case scenario, Purposely underestimating to obtain work)	3
5	Lack of formal estimation process	3
6	Ignoring testing effort	2
7	Insufficient customer involvement during story sizing	2

Table 2. 3 Reasons for inaccurate estimates from practice in ASD (Usman et al., 2015)

Ziauddin et al. (2012) used so-called 'friction factors' in their proposed model of estimation for ASD projects, considering team composition (e.g., appropriately skilled team members) and team dynamics (variations in team members' attitudes towards their roles).

A good working relationship with the customer has long been recognized as significant for project success (Beath & Orlikowski, 1994). The first principle in the agile manifesto notes the importance of customer collaboration and their satisfaction. Further, this principle states "our highest priority is to satisfy the customer through early and continuous delivery of valuable software" (Agile Alliance, 2001). In an agile context the customer relationship is dependent on how the team collect, dispatch, understand and utilize feedback from the customer; the relationship has been acknowledged as being complex and multidimensional (Conboy et al., 2011). Numerous studies in agile project management have explored the relationship between the customer and the development team, specifically in the context of their communications (Korkala et al., 2009), their contributions (Kautz, 2009) and their contentment (Mann & Maurer, 2005).

Chow & Cao (2007) identified delivery strategy (with respect to scope, timeliness and cost), agile software engineering techniques (in terms of quality and scope) and team capability (with respect to timelines and cost) as the three critical success factors for agile software development in a survey study conducted among agile professionals across 109 agile projects from 25 different countries.

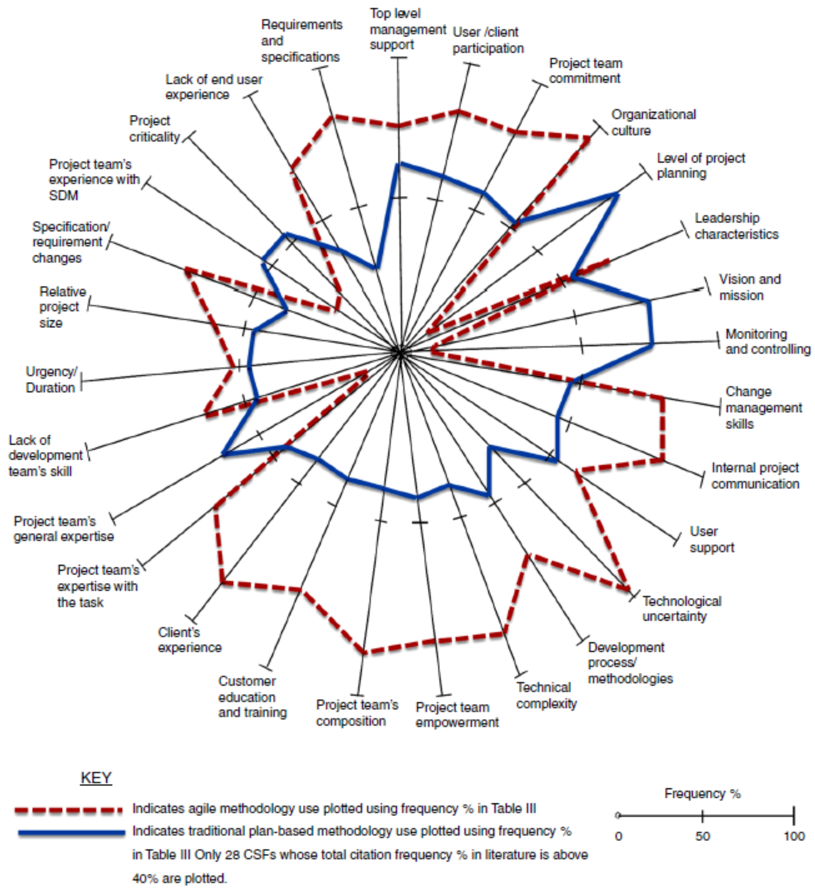


Figure 2. 4 Comparison of CSFs between methodologies (Ahimbisibwe et al., 2015)

Cost Drivers	No of Reponses
Quality Management	9
Complexity	7
Experience	7
Team composition	5
Work environment	5
Technical ability	5
Task size	4
Integration	4
Priority	4
Delay response stakeholders	4

Table 2. 4 Cost factors for estimation in Agile Context (Dantas, et al., 2018)

2.5 Systems Context

2.5.1 Relevant Evidence

In studies where 'software type' is considered to be an indicator of system complexity most of those researchers also consider it as a significant factor of influence on the required effort. However, the specific classification varies among the researchers: they may use different terms (e.g., real-time; control systems; scientific programs), and even when similar terms are used it may be that each researcher interprets the terms differently (Levy, 1990).

Software complexity is typically thought of in three forms: structural, which refers to the complexity of the design & composition of the software product; conceptual, which views complexity as a function of how easy it is to understand the system, the source code, the requirements, and/or the novelty of the software product; and computational, which reflects complexity in computational performance (Laird & Brennan, 2006; Camci & Kotnour, 2006).

Project complexity is also a relatively common concept described in the literature, but one that again has various interpretations based on the context of use, or on individuals' perceptions; in some cases, this concept is used in place of project size, or as an alternative to project difficulty or product complexity. The underlying intent is to acknowledge that project size is not the only important aspect to consider, though the operationalisation of complexity presents other challenges. Thus, project size is a significant feature and one that often serves as a proxy to reflect the essential complexity of a project, and it is one of the most important and often considered attributes of a software project. In most cases it is considered to be directly related to the effort required to implement the project and to the productivity of the development team (Fitsilis et al., 2011).

In a proposed framework for the analysis of estimation error Grimstad & Jorgensen (2006) provided a comprehensive list of factors that can have a principal impact on estimation accuracy, including estimation ability, complexity and measurement process factors. Further, this study indicates "inherent Project execution complexity" as a sub-factor of the complexity factors, which includes innovative projects; i.e., projects developed using leading-edge technology and having complex functionality are inherently more difficult to estimate as compared to simple or repeating projects. "In the software engineering literature, there is a notion that applications of a similar type or algorithmic domain experience have similar level of complexity & productivity". Further, a study regarding the moderating effects of structure on volatility and complexity in software enhancement shows application type as an important factor (Banker & Slaughter, 2000).

Applications that incorporate induction-based algorithms to support activities such as planning, forecasting and management for decision making are more complex and volatile as compared to the rule-based algorithms in applications that support operational and transaction-processing activities (Jones, 1991).

Software complexity also derives from the amount of data that the application must process (Card & Glass, 1990). Rombach (1987) conducted an experiment using graduate programmers and concluded that more structured code requires less maintenance effort. One further study reveals high maintenance effort for informally developed applications as compared to structured applications (Kada et al., 1993).

Many software cost estimation models have considered application type as a significant adjustment factor for cost but they have used different terminologies in their models; for example, Putnam and Myers (1997) referred to it as the “fourth factor”; Jones (1991) considered “project type” and “algorithm type” in his SPQR and feature point model, respectively; Kemerer (1993) considered this aspect as an “influential factor” in his Function Point model; while Boehm named it “modp” in the COCOMO model in 1981.

2.6 Organization Context

2.6.1 Relevant Evidence

“An organization is a social arrangement which pursues collective goals, which controls its own performance, and which has a boundary separating it from its environment, and the culture within such an organization is defined and lived by its people” (Bulander & Dietel, 2015). In today’s business environment the majority of processes within an organization are accomplished through combinations of people and technologies (Weske, 2012).

Organization structure and culture both affect the capability to accomplish project success. The culture of an organization is a set of shared values, behaviours, and assumptions that describes its functional view and has 10 characteristics: member identity, group emphasis, people focus, unit integration, control, risk tolerance, reward criteria, conflict tolerance, means-ends orientation, and open-systems focus (Schwalbe, 2014).

Sharma & Gupta (2012) describe how the provision of a resilient organisational environment is more likely to ensure project success. Furthermore, team cohesiveness, mutual trust, freedom of expression, team autonomy, and domain-specific and creative skills are the dominating characteristics of an open, trusted & robust organizational environment.

Understanding the concept of organizational culture and its implications within a specific company context has thus been considered as important in prior studies (Moor, 1995; Shore & Warden, 2008). It influences how processes are managed, which in turn influences both the software projects undertaken as well as the context in which software development takes place. It can directly affect the project selection process and the usage of technologies and practices in industry. The acclimatisation of a new practice or behaviour in software projects requires either the practice to be adjusted to fit the organizational culture or the culture to be moulded to fit the behavioural requirements of the new practice – or some combination of the two. An appropriate fit between the values and beliefs amalgamated in the context of software development and the organization's overall worth will lead to a more effective implementation (Strod et al., 2009; Tolfo & Wazlawick, 2008).

Sharp & Robinson (2010) give attention to the cultural and communal aspects in successful XP practices, the characteristics of teams and the kinds of social interactions that occur in these practices, and how they are themselves influenced by organizational culture and team characteristics. In prior work they had discovered five characteristics of 'XP culture': respect and responsibility that lie with both the team and its members; preserving the quality of working life; the XP process itself values both individuals and teams, to create in them faith and trust in their own abilities in terms of achieving their set goals throughout the life cycle of their activities. Robinson & Sharp (2010) state "XP is as much about human values as about technical values" (p.139).

Ziauddin et al. (2012) suggest the relevance of 'friction factors' in their proposed model of estimation for ASD projects, including environmental factors (turbulences that may hinder team productivity, e.g., noise, poor ventilation and lighting, uncomfortable seating arrangements and improper hardware/software availability) and process factors (e.g., unexpected or undesirable changes to an adopted process).

Schein (2009) established three different levels of organizational culture, as shown in Figure 2.5: First, visible Artefacts - this level deals with aspects such as the work setting, technology-dependent working routines, myths, language, patterns of behaviour of the organization members, and stories; Second, Espoused Values - this level explains behaviours and includes aspects such as company strategies, beliefs, goals, rules and philosophy; the Third level, Basic Underlying Assumptions, includes the right way(s) to feel, perceive and think for members of a group.

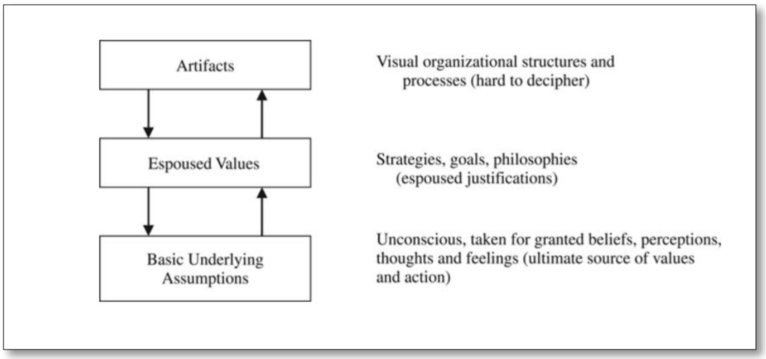


Figure 2. 5 Levels of Organization Culture (Schein, 1999)

2.7 Iteration / Sprint Context

2.7.1 Relevant Evidence

It seems intuitive to expect that estimates and plans be generally less accurate early in projects when information and certainty are at a premium, and that accuracy should improve as projects progress (all other things being equal).

A degree of evidence in support of this expectation was found in a case study conducted at the University of Ljubljana that had the aim of studying the behaviour of development teams who were using Scrum for the first time (Mahnic, 2011). In this study, the planned velocity estimates of teams were very optimistic in the first sprint and only one team out of 13 actually accomplished all of the functionalities as per their commitment in the sprint planning meeting. Further, a systematic study (Nguyen-Cong & De Tran-Cao, 2013) reviewing the current literature on effort estimation in agile, iterative and incremental software projects shows how the magnitude of errors over the iterations / releases are reduced, as shown in the table 2.6.

Estimation approach	Metric	Magnitude of estimation errors are reduced over iterations/releases (Proji=Project i)				
		1st	2nd	3rd	4th	5th
UCP [P05]	MRE	0.21	0.17			
UCP [P25]	MRE	0.20	0.28	0.14	0.03	0.52
Model-based estimation [P09]	MRE	–	< 0.12			
Composite model [P10]	MMRE	1.07	0.32	0.23	0.23	0.14
Regression-based estimation [P12]	MRE	–	–	Proj1: 0.51 Proj2: 0.57	Proj1: 0.32 Proj2: 0.25	Proj1: 0.11 Proj2: 0.16
Customized model [P13]	MRE	Proj1: 0.25 Proj2: 0.03 Proj3: 0.28	Proj1: 0.09 Proj2: 0.00 Proj3: 0.19	Proj1: 0.00 Proj2: 0.00 Proj3: 0.00		
Bayesian network [P15]	MRE	0.96	0.73	0.46	0.55	0.44

Table 2. 5 Estimation of Errors over iterations / releases (Nguyen-Cong & De Tran-Cao, 2013)

2.8 Risk Context

2.8.1 Relevant Evidence

Empirical evidence supports the assertion that inadequate awareness and management of software project risk is an important reason for project failure; and that high-level risks are associated not just with failure but with undesirable project outcomes such as low quality, budget overruns, and late delivery of the software product. Numerous studies have proposed conceptual frameworks that depict a range of software development risks and their various mitigation strategies (Sharma & Gupta, 2012).

Software project risk management has rarely addressed organizational risks even though most problems occur when the social aspects of organisational work are not addressed. A recent study has proposed an organizational risk diagnosis framework in order to convey how these types of risks could be understood and managed in a better way (Vrhovec et al., 2015). However, organizational risks remain challenging to manage and knowledge of their possible presence alone is insufficient to contribute to project success (Thamhain, 2013; de Bakker et al., 2010).

Unmanaged risks are one of the (albeit numerous) reasons for the failure of software projects (Winch.et .al, 1997, Charette, 2005). Some of the risks linked with project failures are uncertainty regarding project complexity and project goals, and personality differences (William, 2005). Effective risk management can enable project managers to identify the potential risks of their projects and to mitigate them through dynamic courses of actions (Morski & Miller, 2002).

2.9 Techniques of Potential Relevance to the Current Study

This research study seeks to understand and improve the utility of forecasts of size, effort, cost and schedule for ASD projects through variants of approaches. Some of these approaches or their variants as explained in the preceding sections are chosen based on all or a few reasons as stated below to analyse their usefulness and efficacy in the context of the current study:

a) Popularity/use of technique in software development for estimating effort and/or schedule
b) Researcher's prior knowledge and practical exposure about the technique
c) Significance of the technique for the intended study

2.9.1 Expert Judgement

Expert judgement is an informal means of estimating effort for software development activities/projects based on experts' own experiences (MacDonell & Shepperd, 2003). Results of a review of studies on expert estimation of software development effort have shown that expert judgement based estimation is the most used strategy (Dantas et al., 2018) for software projects as compared to estimation models (Tanveer et al., 2017), and in some situations expertise-based estimation is more accurate than model-based estimations (Jørgensen, 2004).

The Delphi technique and work breakdown structure (Tausworthe, 1980), informal analogy-based reasoning, rules of thumb (Jones, 1996), top-down and bottom-up estimation (Tausworthe, 1980; Jørgensen, 2003; Shekhar & Kumar, 2016) all fall under the class of expert judgment techniques.

Top-down estimations are made based on an assessment of an entire project, with effort then allocated over the various components, whereas bottom-up estimations are based on determining the effort of each of the components of the project (Jørgensen, 2003; Shekhar & Kumar, 2016) and then aggregating these up to a total for the project as a whole. Each technique has its advantages depending upon the specific situation, i.e., the top-down estimation technique may be applied in a situation where the estimator(s) has little or no knowledge / experience about developing the components of a software project but has already estimated a similar project or has some experience with similar projects, whereas the bottom-up approach can be very helpful in producing accurate estimates when the estimator(s) may be inexperienced in general but has an awareness of the project elements; it is typically a more time-consuming approach (Jørgensen, 2003). Use of both top-down and bottom-up approaches independently in a single software project may reduce the chances of extreme inaccuracies in estimation (Jørgensen, 2004).

2.9.2 Program Evaluation Review Technique (PERT)

The PERT methodology, developed by the U.S. Department of Defense, is used to estimate an average value for effort based on three estimates: optimistic time, pessimistic time and the most likely time required for completing a task or an activity. The average time is calculated by the following formula:

$$E_t = (O_t + 4M_t + P_t) / 6$$

Where:

- Estimated Time (E_t): is the average time
- Optimistic time (O_t): is the minimum possible time required to accomplish a task, assuming everything proceeds better than is normally expected
- Pessimistic time (P_t): the maximum possible time required to accomplish a task, assuming everything goes wrong; for example maximum possible delay in the availability of the hardware or software resources for the task and/or the expertise for a particular tool (but excluding major catastrophes)
- Most likely time (M_t): the best estimate of the time required to accomplish a task, assuming everything proceeds as normal.

A Quasi-experiment was conducted (by the candidate) to apply PERT to the Technical and Environmental Factors of the Use case points (UCP) estimation method to determine its impact on the accuracy of the resulting estimations. This resulted in a six percent improvement on average in the accuracy of the effort estimation for ten small-scale software projects (Tariq & BC, 2011). Further, numerous researches have been conducted to illustrate the PERT concepts and their application (Nickolet, 2007); e.g. PERT applications in project management and quantitative risk analysis (Intaver Institute, 2006) and health management organization (Stefan & Burcea, 2007). One early study also demonstrated the effectiveness of PERT in project planning and contracting by proposing new uncertainty and criticality measurements (Mummolo & Giovanni, 1997).

2.9.3 SCRUM

SCRUM is one of most popular variants of agile development (Usman et al., 2017; Torrecilla-Salinas et al., 2015; Inayat et al., 2015; Grapenthin et al., 2014; Usman et al., 2014; Nguyen-Cong et al., 2013), used around the globe as a development framework for IT and other projects (Baham, 2019). SCRUM focuses mainly on three roles (Product owner, Scrum Master, and Team), three documents (Product backlog, Sprint Backlog, and sprint results), and three meetings (Sprint Planning, Review, and Daily Scrum) as shown in the figure 2.6 (Stober and Hansmann, 2010). Descriptions of the above stated roles are given in the table 2.7 (Cohn, 2010; Sutherland & Schwaber, 2016).

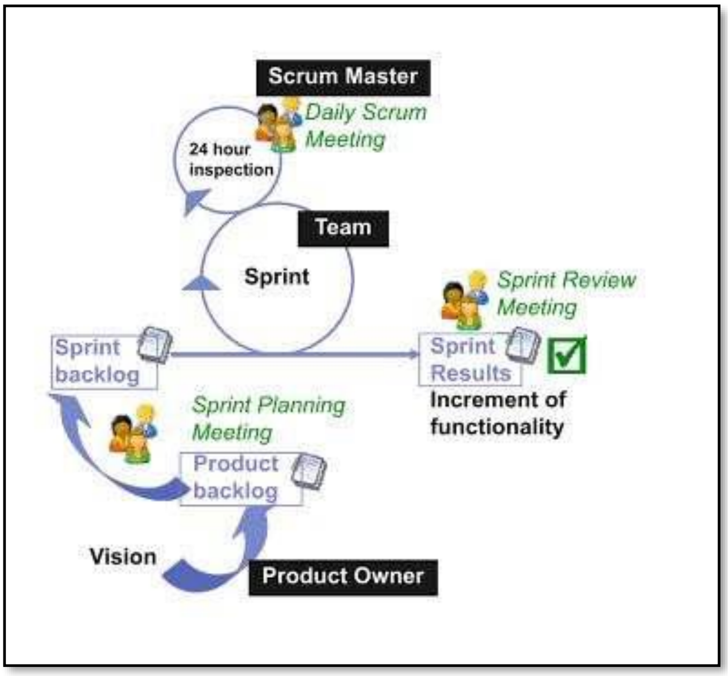


Figure 2. 6 SCRUM Overview (Duka, 2013)

SCRUM working is very simple (Duka, 2013, MGS, 2019) as exhibited in the following steps:

a. The Product Owner (PO) takes decisions about the items of the product backlog that should be prioritised, i.e. the features anticipated from the next release of the product/project, and their associated deliverables. S/he acts as a representative of the stakeholders.
b. In every sprint planning meeting, the team decides which items from the product backlog are taken to the sprint backlog. The selection of these items is dependent upon their value to the business, technical requirements and/or other commitments as set by the Product Owner and/or team. During the execution of each sprint (typically lasting from 1-4 weeks), the team tries their best to complete the items as allocated to the current sprint.
c. A daily time-boxed Scrum meeting (normally 15 minutes duration) is held preferably in the morning at the same place/time to follow-up the ongoing tasks and further to resolve any impediments / issues. The Scrum Master, being a facilitator, helps in resolving issues discussed during the meeting.
d. The Sprint review meeting is conducted after the conclusion of each sprint to evaluate the outcomes and the lessons learned of the finishing sprint. Each concluding sprint is an increment to the intended software product.
e. The same process as stated above from sub-para (a-d) is repeated until the whole product backlog become emptied and product's releases are ready to deliver. Further, the releases can be repeated to cater for any changes or updates.

Roles	Description
Product Owner (PO)	PO is one of the significant members of the Agile team who is responsible for defining user stories and prioritize them based on their business values and also to maximize the product's value and the team's work.
Scrum Master (SM)	SM is one of the important members of the agile team who works as a facilitator between team and the management to address project related issues/impediments, etc. and coach to train team on SCRUM's various aspects. Moreover, to make sure that these aspects have been understood and enacted.
Development Team	Professionals/developers who works in agile development environment and responsible for shippable deliveries of the "done" product at the end of each Sprint.

Table 2. 6 SCRUM's Roles Descriptions (Cohn, 2010; Sutherland & Schwaber, 2016)

2.9.4 Planning Poker (PP)

Planning Poker is one of the contemporary techniques used commonly for estimating the size of each story in Scrum projects. PP is a consensus-based estimation effort technique proposed initially by Grenning in 2002 and later popularized by Cohn in his book in 2005. The Planning Poker method was believed to be consistent with the people-oriented approach of ASD (Javdani et al., 2012). In Planning Poker, the estimates are limited to precise numbers as written on a set of cards. During the effort estimation of a story, each member of the estimating team is holding a deck of Planning Poker cards with values such as 0,1, 2, 3, 5, 8, 13, 20, 40 and 100 (a popular sequence (Cohen, 2004-5) but there are others). Each of the values written on the cards represents the number of story points (referring to the relative size of the client's functional requirements), ideal days (required to implement the client's specific requirements), or other units depending upon the team's preferred unit of estimation. Figure 2.7 shows a sample of a card deck for Planning Poker.

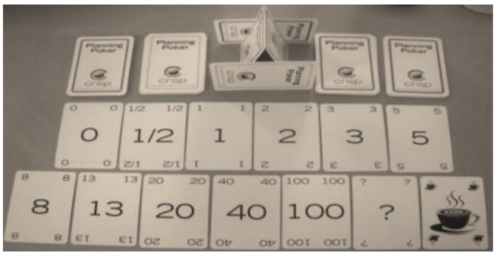


Figure 2. 7 Planning Poker Cards (Gandomani et al., 2014)

During the estimation process, team members discuss the features if required with the stakeholder and / or with the product owner (who represents a customer / stakeholder in their absence). After completing the desired discussion each of the team members individually selects one card from the deck of the cards to represent his or her estimate and reveals it at the same time as all other members of the team. If all team members have shown the same value, then it is considered as an estimate of that specific story, otherwise they need to discuss their estimate differences with reasons, especially those members with the highest and lowest estimate values. The whole process of selecting and revealing the cards is repeated again until a consensus is established among the team members, or a decision could be made to defer this activity until further additional information is provided to complete estimate for a specific story (Cohen, 2005).

2.9.5 Learning Curve (LC)

Learning Curve accounts for learning from experience and/or repetitive similar tasks (manual and/or tool based) that enriches the performance of workers, a phenomenon evident in numerous studies conducted in a wide range of industrial domains, including software, electronics, construction, chemical and automotive (Anderson,1982; Pananiswami and Bishop, 1991; Adler and Clark, 1991; Nembhard and Osothsilp, 2002; Nembhard and Uzumeri, 2000a; Vits and Gelders, 2002; Hamade et al., 2007; Chen, 2009; Jarkas, 2010; Weber and Fayed, 2010).

LC was developed empirically for the first time by Wright in 1936 when he observed the cost reduction in airplanes' assembly due to the repetition of tasks. This reduction in cost presented at a constant rate when the assembled airplanes doubled which was called 80% learning rate (LR), a figure widely employed in the aeronautical industry of that time. The LR demonstrates that the cumulative cost of the assembly reduced on average by 20% on each doubled-up number units (Teplitz, 1991; Cook, 1991; Badiru, 1992; Argote, 1999; Askin and Goldberg, 2001). The first formal LC model which Wright proposed in 1936, a Log-Linear Model, is stated as follows:

$$C_X = C_u X^b$$

Where:

- C_X is the average time or cost to produce X units
- C_u is the time or cost for the first unit
- b is the slope of the LC which indicates workers' LR. The value of b is calculated by dividing the Log (LR) by Log base 2 (i.e., Log_2).

A smaller LR implies faster learning, e.g., 75% LR indicates faster learning while 99% suggests slower improvements in productivity. LR may differ from one industrial process to another but in practice most industrial processes note their LR between 80-90% (Racoon, 1995). Typical LRs are shown in Table 2.8. Furthermore, Racoon suggests an 80% LR for software development processes, based on 75% worker and 25% machine-paced processes.

Typical LR	Percent Worker-Paced	Percent Machine-Paced	Remarks
70-75%	99%	1%	Indicates fast LR. Enhancement needs that learning continues without interruption.
80%	75%	25%	Recommended for Software Development processes.
85%	50%	50%	Moderate LR.
90%	25%	75%	
98-99%	1%	99%	Shows slow LR. Improvement needs the upgrading of machinery which is dependent on the availability of finance.

Table 2. 7 Typical Learning Rates (Racoon, 1995)

Numerous studies suggest the efficacy of the Learning Curve (LC) effect in software development. In one such study, the presence of an exponentially reducing LC relationship between effort and an Integrated Computer Aided Software Engineering tool was shown in regard to the experience of a programmer (Pendharkar and Subramanian, 2006). In another study the way to utilize a proposed LC-based simulation model for software development in industry was discussed. This model suggested solutions for issues/complications which arose due to variations in the developers' productivity based on their learnings to estimate and/or forecast projects' progress (Hanakawa et al., 1998).

2.10 Chapter Review

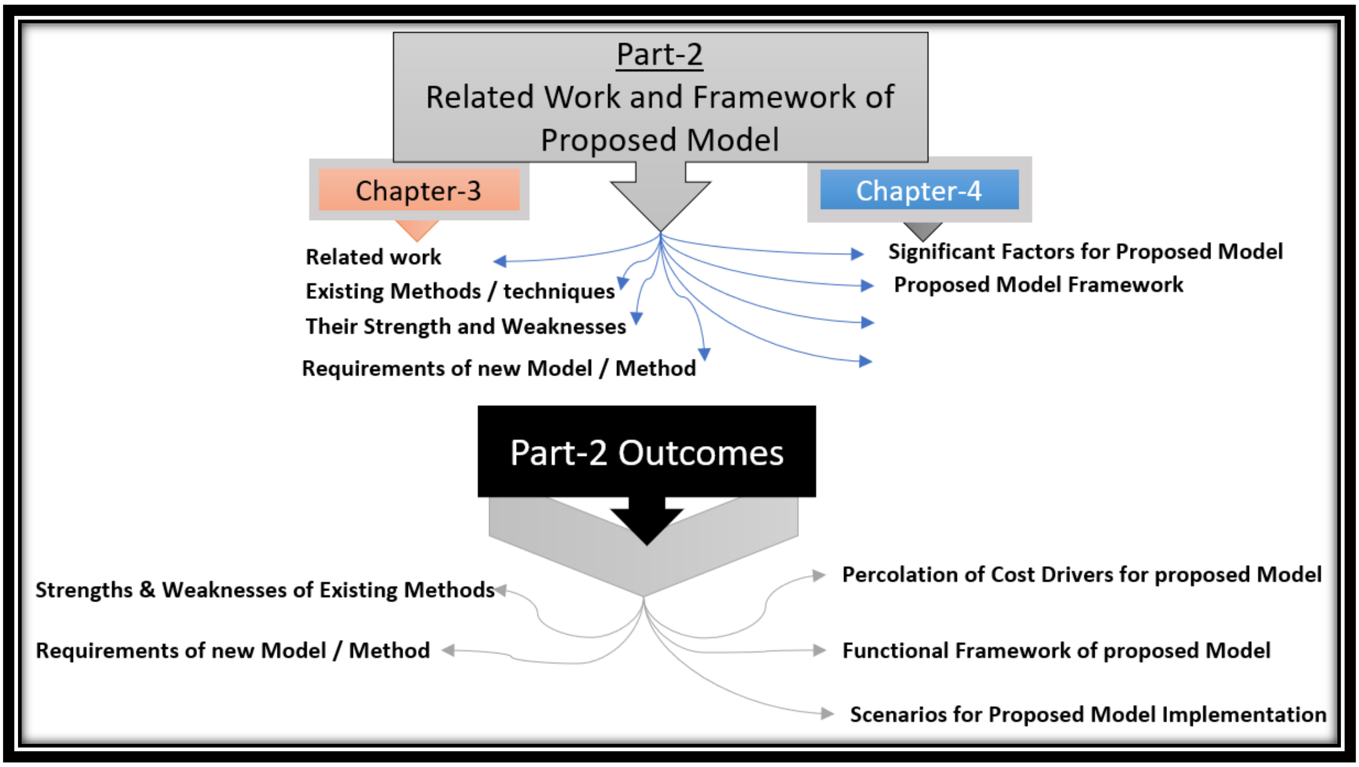
In this chapter, the specifics of various methods of software development, and particularly agile methods, were considered. Effort estimation in agile development contexts and the factors potentially significant to that estimation were discussed. Descriptions of some of the possible techniques to be used in this study were provided.

In the next chapter, contents drawn from literature reviews reflecting the perspectives of industry and academia in the domain of estimating effort, cost and scheduling are elaborated. Further, numerous methods/models for effort/cost estimation and their comparisons, along with their categorizations in algorithmic, non-algorithmic, and hybrid groupings, are discussed.

PART-2 RELATED WORK AND PROPOSED MODEL'S FRAMEWORKS

This part consists of two chapters which reviews related studies conducted in the area of the proposed research and presents comparisons of the weaknesses and strengths of some of the current and traditional methods/techniques/models used for effort estimation. Further, the theoretical and functional frameworks of the proposed model are discussed in this part.

Detailed views of the research activities with respect to the chapters within this part are presented in the figure as given below. On the top of the diagram, the chapters covered in this part are shown along with their different research endeavours as accomplished within the individual chapters, while the outcomes of Part-2 based on its individual chapters are exhibited at the bottom of the figure.



Chapter-3 Related Work / Literature Reviews

***Content Summary**: This chapter reviews the prior literature addressing software effort estimation. It covers the significant existing methods and, in particular, parametric and AI-based models used for effort estimation. Further, it compares, and contrasts proposed and existing AI-based methods and parametric models.*

3.1 RELATED WORK

Numerous studies have provided empirical evidence demonstrating that AI-based methods (e.g. Mamdani FIS (Fuzzy Inference System), Artificial Neural Network (ANN), ML (Machine Learning), Support Vector Regression (SVR), decision Trees (DT), Bayesian Networks (BN), Support Vector Regression (SVR), Genetic Algorithms/programming (GA/GP), Association Rules (AR), and others) can be used successfully in the area of software cost estimation and optimization (Arora, 2018). Therefore, previous studies on the estimation of effort, cost and/or schedule are classified on the basis of these two directions, i.e., studies using AI-based methods and studies not using them. Descriptions of the relevant studies as per the defined two categories are given in the subsequent subsections. Further, in this chapter, many of the prior evaluations of estimation method performance are presented based on the Magnitude of Relative Error (MRE) measure of accuracy/error, as it was the most used metric over the last decade or more (Usman et al., 2018). However, currently MRE is not utilized to measure estimation models' accuracy, due to criticisms of bias in numerous studies (Shepperd et al., 2000; Kitchenham et al., 2001; Foss et al., 2002; Jørgensen, 2007; Myrtveit and Stensrud, 2012; Shepperd and MacDonell, 2012). Instead, the Balanced Relative Error (BRE) and Balanced Relative Error Bias (BREbias) metrics (their formulas are presented in Figure 9.17 of Chapter-9) are applied in this study due to their balanced performance in situations of over- and under-estimation (Moløkken-Østvold et al., 2008; Mahnic et al., 2012; Usman et al., 2018).

3.1.1 STUDIES ON THE ESTIMATION WITHOUT AI-BASED METHODS

According to a 2007 systematic literature review compiled by Jørgensen and Shepperd (Table 3.1 shows the individual and collective distribution of the studies on the various estimation approaches against the period in years), the emphasis of the bulk of the research in software effort estimation had been on the introduction of new methods and on their assessment and appraisal when compared to existing methods (Jørgensen & Shepperd, 2007). Given its timing, much of that review focused on methods that had been proposed for a pre-Agile development world. In the interim there had been a growing body of research in ASD and increasing evidence that Agile Methodologies (AMs) had the potential to considerably improve the way software was produced (Vijayasarathy & Leo, 2008). Given this, contemporary forecasting methods must fit with the agile way of working (Abrahamsson & Koskela, 2004). To date, however, there had been limited empirical evidence produced in regard to the efficacy of forecasting methods in an ASD context. In addition, some of that work had perhaps misunderstood or mis-represented genuine ASD – while that might be the case, we did not make such discernment here. Rather, we accepted the authors' position that their approach was intended to fit in an agile context.

Estimation approach	-1989	1990-1999	2000-2004	Total
Rg	21 (51%)	76 (47%)	51 (51%)	148 (49%)
An	1 (2%)	15 (9%)	15 (15%)	31 (10%)
Ej	3 (7%)	22 (13%)	21 (21%)	46 (15%)
Wb	3 (7%)	5 (3%)	4 (4%)	12 (4%)
Fp	7 (17%)	47 (29%)	14 (14%)	68 (22%)
Ct	0 (0%)	5 (3%)	9 (9%)	14 (5%)
Si	2 (5%)	4 (2%)	4 (4%)	10 (3%)
Nn	0 (0%)	11 (7%)	11 (11%)	22 (7%)
Th	20 (49%)	14 (9%)	5 (5%)	39 (13%)
By	0 (0%)	1 (1%)	6 (6%)	7 (2%)
Cb	0 (0%)	3 (2%)	2 (2%)	5 (2%)
Ot	2 (5%)	7 (4%)	16 (16%)	25 (8%)
The abbreviations used are: Rg = Regression, An = Analogy, Ej = Expert judgment, Wb = Work break-down, Fp = Function Point, Ct = Classification and regression trees, Si = Simulation, Nn = Neural network, Th = Theory, By = Bayesian, Cb = Combination of estimates, Ot = Other				

Table 3.1 Estimation Approaches (Jørgensen & Shepperd, 2007)

An algorithm-based change effort estimation model said to work within a Software Development Phase-Change Impact Analysis Framework was proposed for both traditional and agile methods to estimate the cost/effort of software projects. This proposed model was evaluated by applying it on 6 real software projects in both agile and traditional contexts. The authors asserted that the modelling outcomes confirmed its suitability for use with both agile and traditional methodologies (Basri et al., 2016).

A COSMIC-based model has also shown better results for effort estimation based on story points (Hussain et al., 2013) as compared to SCRUM's native method of estimation with planning poker (Ungan et al., 2014). In a comparative analysis applying Function Point Analysis through different approaches (i.e., one, extended FPA, two, estimation in agile development and FPA, and three, agile estimation using functional metrics), the third approach outperformed the other two (Wanderley et al., 2018). In one more study, COSMIC function points (FP) and story points (SP) were compared against three ASD projects. The outcomes of this study suggested that COSMIC FP offered comparatively more objective estimates than relative SP. Further, this research claimed that COSMIC FP can surpass SP if it is used as an independent variable in a regression model (Salmanuglu et al., 2017).

While simple models have achieved encouraging results typically in one-off scenarios, other research has shown that there are many factors that can affect effort estimation while using story points in an agile development context, such as priority, story size, complexity factors and team experience (Tanveer et al., 2016; Popli et al., 2014; Mukker et al., 2014; Downey et al., 2013; Logue et al., 2007). To minimize the negative impact of these factors and to reduce uncertainty in effort estimation Zahraoui et al. (2015) proposed adjusting story-point measures by using various ranges to quantify the given factors through fuzzy scales.

As with TSD projects the estimation of the size, effort, cost and schedule of ASD projects is dependent on expert judgement and/or historical data, and in case of their non-availability various methods of estimation, such as via analogy, planning poker and so on, are not viable. Further, current agile forecasting methods provide minimal detail regarding their estimates, and many are also unable to produce them at a sufficiently early stage in development. To overcome these and similar issues some new methods for effort estimation were proposed which were said to be helpful for the assessment of cost, effort and duration for small and medium-sized ASD projects. A few of them include the following: (i) A model with the name of WebMO (i.e. Web Model) was proposed by Reifer in 2000 for estimating the effort of the web-based applications. WebMO was designed based on the concept of COCOMO-II and SoftCost-OO – a software cost-estimating model proposed by Reifer 1993 - but only suitable for web-based systems. (ii) A method suggested in which the total number of user-stories were identified as per the user-requirements and the total story-points were calculated and the initial velocity of the project determined (Popli & Chuhan, 2014). (iii) A constructive agile estimation algorithm (CAEA) based on eight significant factors, including the project domain, ease of operation, and security, and these factors were included in the estimation after assigning them some weighted values. Further, the CAEA algorithm estimates the effort in two core phases, i.e., early and iterative (BHALERAO & INGLE, 2009). In this model, however, a few significant human-centric factors, e.g., teamwork and collaboration, were not included even though these had been said to be critical in an ASD context (Munialo & Muketha, 2016). In addition, the validation of CAEA on real IT projects in an ASD context was not demonstrated. (iv) Agile method of web estimation (AgileMOW) had been suggested but its usefulness in real industry projects was unspecified (Litoriya & Kothari, 2013). This model considers 24 significant attributes (12 high- and 12 low-priority attributes) and is an extension of the COCOMO-II model and WebMO as proposed by Reifer in 2000.

Ziauddin et al. (2012) developed a simple model for estimating the effort and cost of ASD projects. This model was calibrated using empirical data from 21 software projects and the model's accuracy was evaluated based on MMRE and PRED metrics. Both of these metrics reported values within the acceptable ranges of MMRE (7.19%) and PRED (57.14%) for estimating effort with respect to time. This model proposed customized scaling for story-sizing from exceptionally large to small, and the complexity of user-stories from extremely complex to straightforward. Further, it included a few friction factors in context of team, process & environment along with nine dynamic factors to address various aspects of the project and development team (e.g., expected relocation of a team, vendors' issues, tools/technology expertise and others) was demonstrated by the model. That said, the model does have limitations: the practical efficacy of the model is not shown and only accuracy has been reported; the model suggests values for story-points (SPs) along with completion time, while in agile development contexts teams have autonomy and prefer to estimate SPs themselves; and the Velocity calculation is done based on the model's own estimated SPs divided by the time taken in a sprint, which is again an anomaly from preferred practice.

Studies have also shown the substantial benefits that can accrue when combining techniques to improve effort forecasts in software project management, e.g., expert judgement, least squares regression and case-based reasoning. Such an approach relies on the expectation that when one technique fails to forecast well then another tends to perform significantly better (MacDonell & Shepperd, 2001; Jorgensen, 2007).

3.1.1.1 CRITICAL ANALYSIS ON ESTIMATION METHODS WITHOUT AI

There are some inadequacies found in the models of the aforementioned studies which include the following: the majority of models have limited empirical evidence to show their practical efficacy with real-industry projects, e.g., the constructive agile algorithm; human factors such as teamwork, collaboration, and the like are not considered which are otherwise important in an agile environment; limitations in model use for multiple types of system, e.g., WeMo can only be used for Web-based applications; the models' cost effectiveness is unknown; models typically suggest agile team replacement; and the actual implementation of the models in the field is not demonstrated.

3.1.2 STUDIES ON THE ESTIMATION WITH AI/SOFT COMPUTING METHODS

Pospieszny et al. (2018) found that a State Vector Machine (SVM) outperformed ANN and GLM approaches when used to estimate effort for ASD projects in the ISBSG's dataset (both with and without log transformation of the variables included). The authors of this study were of the opinion that the variations in the outcomes were due to the heterogeneous nature of the ISBSG's dataset, whereas results could be different for more homogeneous datasets. In an earlier study, long short-term memory and a recurrent highway network were used to estimate effort for user stories instead of for the project as a whole. This technique was applied on 16 open-source projects which comprised 23,313 user stories in total. The outcomes of their suggested settings/models surpassed the techniques of random guessing, the median, and the mean (Choetkiertikul et al., 2015).

A model for effort estimation was proposed by Alostad et al. (2017), in which three significant factors (i.e., the team's experience, size and complexity of the story) were taken as inputs to a Mamdani Fuzzy Inference System (FIS) type Fuzzifier, generating rules to predict story points required for the user stories or issues. The suggested model resulted in higher accuracy rates with respect to MMRE and PRED; i.e., 0.28 and 50% respectively. Moreover, the model exhibited substantial improvement in accuracy with an increase in the team's experience when the model was used for more sprints (Alostad et al., 2017).

To improve on what was purported to be poor estimation practice in an agile context, an automated estimation methodology called "Auto-Estimate" was proposed by Moharreri et al. (2016) to complement the agile manual process of estimation conducted through planning poker. The proposed approach extracted features from story cards and related these to their associated records of actual effort through different methods of machine learning (ML). The generated estimates of the selected machine learning methods were said to have resulted in good performance and higher estimation accuracy as compared to planning poker (i.e., ML modelling resulted in a 91.75% MMRE whereas the comparable result was a 106.81% MMRE in the case of manual Planning Poker), especially in the advanced stages of a project (Moharreri et. al., 2016).

A hybrid Multilayer Dilation-Erosion-Linear Perceptron (MDELP) model was proposed for effort estimation by Ricardo et al. (2017). As far as can be found this model had not yet been utilized on ASD projects and, therefore, its usefulness in an ASD context is unknown. However, this hybrid model showed the best performance in terms of the MMRE (0.06) and PRED25 (95.84) measures as compared to that achieved by the existing approach. In another study, a proposed hybrid model employing a Bayesian network (BN) exhibited accuracy above 90% for all datasets when applied to estimate the effort of 160 tasks from agile development context. The authors used RMSE along with other measures of accuracy (i.e. MMRE, Pred, MAE) to check the deviations in effort estimations of their model as compared to actual effort, The proposed model revealed Pred value (m=25) equals 100% while MAE within 2.6% for the estimated effort. (Dragicevic et al., 2017).

Furthermore, one more study (Ionescu et al., 2017) showed promising results, i.e., approximately 50% better, as compared to previous empirical studies (with the ratio of 12:6) when they used ML techniques, i.e., TF-IDF (Term Frequency-Inverse Document Frequency), SVR (Support Vector Regression), and GNB (Gaussian Naive Bayes) to estimate effort versus the traditional methodologies (i.e., COCOMO).

Another BN model was proposed for use in a SCRUM environment for the purpose of estimation. The suggested model determined estimates based on two inputs, i.e., complexity and the importance of the story. The researchers used correlation tests to validate their model. Further, they were of the view that this model was helpful for teams who were new and/or planning to adopt SCRUM as their upcoming development environment (Lopez-Martinez et al., 2017). An optimized algorithm (called a Satin bowerbird) based on a hybrid fuzzy inference system was suggested in a further study. This research claimed more accurate results in effort estimation but still, its effectiveness for ASD projects was unknown due to that context's difference in input parameters that could be fed to the fuzzifier.

The proposed algorithm produced good MMRE results as compared to an existing technique, i.e., CART, when applied on ISBSG datasets (Moosavi & Bardsiri, 2017).

Wen et al. in 2012 identified approximately 84 primary studies in their systematic review proposing eight types of AI/soft computing estimation model; i.e., Case-Based Reasoning (CBR), Artificial Neural Networks (ANN), Decision Trees (DT), Bayesian Networks (BN), Support Vector Regression (SVR), Genetic Algorithms (GA), Genetic Programming (GP), and Association Rules (AR). The review concluded that the accuracy of the effort estimations and the results' acceptability of such models were much better than those from non-ML models. Further, CBR, ANN, and DT were the most frequently used ML techniques for effort estimation, and they were found in 80% of the above stated primary studies. However, in one more study, it was concluded that analogy-based software effort estimation (ASEE) had provided better results as compared to the above eight given ML techniques, especially when ASEE was used in combination with fuzzy logic or GA (Idri et al., 2015). A range of ML and other relatively complex techniques had also been used in other such studies – for instance, a General Regression Neural Network, a Probabilistic Neural Network, a Polynomial Neural Network, Cascade-Correlation Neural Network, Radial basis function neural network, General regression neural network, Cascade correlation neural network and Multilayer Perceptron had been assessed for efficacy (Panda et al., 2015; Nassif et al., 2016).

Other studies that sought to improve effort estimations by using soft computing methods in ASD include: the use of the back propagation learning algorithms on a multilayer perceptron (Agahi et al., 1998; Chulari et al., 1999), the utilization of Genetic Programming successfully on public datasets (Lefley and Shepperd, 2003), the application of three techniques (i.e., SVR, RBFNs and Linear Regression) (Oliveira, 2006), and the use of Fuzzy Decision Trees (Andreou & Papatheocharous, 2008).

3.2 Estimation Methods / Models

Effort estimation techniques / methods can be classified into various types as discussed in a range of review studies (Jorgensen, 2005; Jorgensen & Shepperd, 2007; Suri & Ranjan, 2012; Bardsiri et al., 2013). One of simpler classifications was proposed by Bardsiri et al. in 2013 in which the estimating techniques were grouped into two core types: algorithmic and non-algorithmic. Algorithmic models are normally dependent upon sets of rules that are constructed by using a knowledge base from a range of domains, e.g., Mathematics, Statistics, Arithmetic, Machine Learning, Artificial Intelligence, and the like (Biller, 2000; Li et al., 2009; Basri et al., 2016), while non-algorithmic approaches are based on analysis, reasoning, inferring, and learning (Basri et al., 2016; Saljoughinejad and Khatibi, 2018). An overview of both types is given in the subsequent subsections.

3.2.1 NON-ALGORITHMIC ESTIMATION METHODS

3.2.1.1 ANALOGY BASED ESTIMATION (ABE)

This method is used commonly for software effort / cost estimation (Shepperd et al., 1996; Shepperd & Schofield, 1997; Li et al., 2007; Suri & Ranjan, 2012) and has been seen as a longstanding alternative for traditional algorithmic estimation techniques (Idri et al., 2016; Shams et al., 2019). ABE forecasts the effort / cost of the candidate software project by using information from previously completed similar software project(s). Analogy based estimation provides an opportunity for an estimator to assess the intended project as a whole or in parts. Both options have advantages, e.g., project-level estimations provide the total cost and/or effort of the whole project while part- or component-level estimations enable more fine-grained assessments of similarities and dissimilarities between the new and completed projects (Leung & Fan 2002).

3.2.1.2 EXPERT JUDGMENT BASED ESTIMATION (EJBE)

This is a preferred and popular approach used to estimate effort/cost of software projects in an agile development context (Jorgensen, 2005; Keaveney, 2006; Popli & Chauhan, 2014; Usman et al.,2014; Shams et al., 2019) because of its simplicity and independence of any tool/technique (Huang et al., 2008). In this method, experts estimate the projects' development effort based on their gut feelings, intuition and/or their past experiences of estimating similar projects. Expert judgement can be augmented in numerous ways by using particular techniques, e.g., expert consensus through Delphi or PERT to resolve inconsistencies in the estimations.

3.2.1.3 ESTIMATION BASED ON TOP-DOWN METHOD (EBTDM)

In this meta-method, effort/cost estimation is performed for an overall project or product by using any algorithmic and/or non-algorithmic methods, e.g., estimating a project's effort / cost by comparing it to the effort/cost of similar past projects/products. The overall development effort of a project/product in EBTDM is then distributed among the core / main phases or parts. This approach considers the global properties of a project/product while estimating its effort / cost.

3.2.1.4 ESTIMATION BASED ON BOTTOM-UP METHOD (EBBUM)

This meta-method is in effect the opposite of EBTDM. In this method, a project/product is first divided into relatively small work entities or activities. The effort/cost of each of these small work units is estimated separately and later their individual effort/cost estimates are combined to calculate the effort/cost of the project/product as a whole. The breakdown of a project/product into small workable entities is critical and can only be done where tasks/components are easy to identify sufficiently early in a project and can be compared and contrasted with small entities of completed projects.

3.2.1.5 PARKINSON'S LAW

The phenomenon known as Parkinson's law was first described by G. Nortcote Parkinson. This law says, "Work expands so as to fill the time available for its completion" (Levy, 1990; Leung & Fan, 2002). To implement Parkinson's law, the cost is computed (rather than estimated) by considering the available human resources available to the intended project and therefore, no objective evaluations of the envisioned project are done (Leung & Fan, 2002). For example, if an intended software project has to be delivered in 6 months' time and 7 team members are allocated to this project for its development then as per Parkinson's law the required effort will be 42 person-months. This method is not strictly an estimating approach because this is "a philosophy of despair", but it has been used in practice (Levy, 1990); hence its brief inclusion here.

3.2.1.6 PRICE-TO-WIN

This method is normally used in estimations when a company wants to win a bid / contract for an intended software project. The bidding companies estimate the upcoming project based on the clients' budget instead of the functionalities and/or features of the intended project (or perhaps on some combination of the two). This approach is not considered a sound method of estimation because it tends to be highly optimistic; delays in the delivery of the envisioned project are common due to the unrealistic basis of these estimations (Levy, 1990; Leung & Fan, 2002).

3.2.2 ALGORITHMIC ESTIMATION METHODS

3.2.2.1 COCOMO

This model has been widely used in TSD for estimating the effort/cost of software projects (Boehm, 2000; Benediktsson et al., 2003). COCOMO, an acronym standing for Constructive Cost Model, was developed by Boehm in the 1970s for estimating project effort, cost, and schedule and was published in Boehm's seminal book titled "Software Engineering Economics" in 1981. COCOMO's initial parameters were determined by trial and error and were refined over time by fitting regression formulae using data from historical projects (i.e. 63 projects for COCOMO 81 and 163 projects for COCOMO II). The model has three levels, i.e., Basic (said to be good for estimating software projects' cost quickly with limited accuracy), Intermediate (taking into account the cost drivers that are basically attributes of the Product, Project, Hardware, and Personnel), and Detailed (the final level of the model which also considers the impacts of the individual phases of a project). In COCOMO, an effort adjustment factor (EAF) is calculated by taking the product of all 15 effort multipliers as provided by the Intermediate model. Further, the effort of the intended software project is estimated by using a given formula depending upon the matching type of the project from three available types (to reflect the deployment context).

The cogency and accuracy of the Intermediate COCOMO are in question for today's complex and dynamic software development environment because the model's cost drivers and their values were originally dependent upon the data of software projects completed in the 1960s and 1970s (Hu et al., 1998).

3.2.2.2 FUNCTION-POINTS ANALYSIS (FPA) METHOD

FPA is another approach that was particularly popular for estimating the effort / cost of TSD software projects during the 1980s and 1990s (Lubashevsky, 1996; Yinhuan et al., 2009). This approach surpassed the practice of software project size measurement as compared to the traditional approach of counting lines of the code; rather, it sought to assess the functional scale of a software project/application (Meli & Santillo, 1999). However, to apply this technique effectively, a comprehensive descriptive level document of requirements/design is expected. Furthermore, users' specific logical functions of the intended software project (divided further into two data and three transactional function types respectively) are grouped into added, changed, or deleted sets of functions, and merged with a Value Adjustment Factor (VAF) to get the final value/number of FP. Moreover, a unique formula is utilized for each of two count types, i.e., for an Application / Development or an Enhancement project.

3.2.2.3 USE CASE POINTS (UCP) METHOD

UCP is another TSD approach for estimating the effort/cost of software projects (Braz & Vergilio, 2006; Ochodek et al., 2011), developed by Karner in 1993. This is one of the more extensively used models to estimate effort for software projects (Usman et al., 2014; Kirmani and Wahid, 2015; Ribu, 2008) and is said to be appropriate for both short (up to one month) and longer duration projects (that is, above one year). Further, this model is also helpful to estimate the size of a software project in their early stages of development (Azzeh &Nassif, 2016). The working concept of UCP is based on as system's use case(s), i.e., a description of steps/actions between a user and a system under discussion (Tariq & Shoba, 2011) which are core to the set of UML modelling techniques. This method determines the size and then the human effort of the software project based on calculations of the elements as stated (Tariq & Shoba, 2011):

a. Unadjusted Use Case Weight (UUCW): The size in points of the software project is calculated based on the number of identified complexities (i.e., simple, average, and complex) of all the use cases of the intended system.
b. Unadjusted Actor Weight (UAW): The size in points of the project is also calculated based on the number of identified complexities (i.e., simple, average, and complex) of all the actors of the projected system.
c. Unadjusted Use Case Point (UUCP): UUCP is computed by adding UUCW and UAW.
d. Technical Complexity Factor (TCF): The technical factors of the intended project are identified from a list of 13 factors as provided by the UCP model.

e. Environmental Complexity Factor (ECF): Similarly, the environmental factors that may impact or influence the anticipated project from a list of eight environmental factors are identified, as provided by the UCP model.
f. Use Case Points (UCP): UCP is determined by taking the product of UUCP, TCF, and ECF which is later transformed into human effort estimates if UCP is multiplied by the effective hours (i.e., the time in hours taken by a development team to implement one UCP).

This model has some restrictions that may impact estimation accuracy. The core limitation is related to the somewhat arbitrary criteria that are defined by UCP, which state that a use case can be declared simple, average, or complex based on the number of transactions in it; e.g., a use case is declared simple with 3 transactions, but it becomes an average use case by just adding one more transaction. Further, as per UCP measures, a project with 10 average-sized use cases is double in the size of a project having the same number of simple-sized use cases. Such changes and differences have been described as impractical (Nassif et al., 2016). Furthermore, the TCF and ECF factors of UCP are dependent on the experience of the estimator (Morgenshtern et al., 2007).

3.2.2.4 HYBRID METHODS / MODELS

Hybrid models or methods are modified versions of the approaches as discussed in the preceding sections combined with one or more soft computing techniques/methods e.g. Artificial Intelligence, Machine / Deep Learning, Genetic Algorithm, Neural/Fuzzy networks, particle swarm optimization, grey relational analysis, outlier elimination technique, principal component analysis, regression & rough set theory, and others, to improve the effort/cost estimations of software projects (Ruhe, 2007; Bardsiri et al., 2013). Among multiple ML techniques, numerous studies have shown the efficacy of Neural Network models in the area of effort estimation (Finnie et al., 1997; Samson et al, 1997; Tadayon, 2005; Idri et al, 2006; Park and Beak, 2008; Tronto et al, 2008; Iwata et al, 2009; Balich and Martin, 2010; Reddy et al, 2010; Attarzadeh and Ow, 2010; Shepperd and MacDonell, 2012; López-Martín, 2014) but still, more research work is needed to conclude the most feasible model for the anticipation of effort estimation in terms of precision (Kamlesh, 2019). Details of various hybrid methods and models are already covered in section 3.1.2 of this chapter.

3.3 ESTIMATION TECHNIQUES IN ASD

Agile Software Development (ASD) has fundamentally changed the software life cycle. The emphasis of ASD is on the iterative and continuous delivery of working software features, replacing the formerly large-scale (and often document-heavy) phases of TSD, including Vision, Analysis, Design, and Test (Usman et al., 2017). Moreover, just as ASD introduced the disruptive concept of agility in every step of the software life cycle, it has also introduced a range of new and unique methods for sizing and cost estimation. Some of the more frequently reported techniques used for estimation in ASD are explained in the subsequent subparagraphs.

3.3.1 Planning poker (PP)

Planning Poker is asserted to be one of the more popular sizing estimation techniques in the agile development context (Usman, et al., 2017, Grenning, 2002). It is a consensus-based technique of estimation, the details of which are already covered in section 2 of Chapter 2. Agile teams typically use the Planning Poker method to estimate the relative scale of their user stories in story points by utilizing the Fibonacci series (or similar).

3.3.2 Affinity Grouping

Affinity grouping is said to be very simple but effective when there are a large number of items to be estimated. In this approach, team members are required to group similar sized items (Sliger, 2012) as shown in Figure 3.1.

Figure 3.1 Affinity Grouping Sample (Sliger, 2012)

Affinity grouping also employs the idea of relative size thinking; any item from the list is picked and read out in the presence of the development team and placed on the wall/discussion-board. Then all other items are taken out from the list one by one and placed on the wall/discussion-board as per their sizes after comparing the size of the current item with the size(s) of the items already placed on the wall/discussion-board, in groups of different sizes. The simple steps of this method are shown in Figure 3.2.

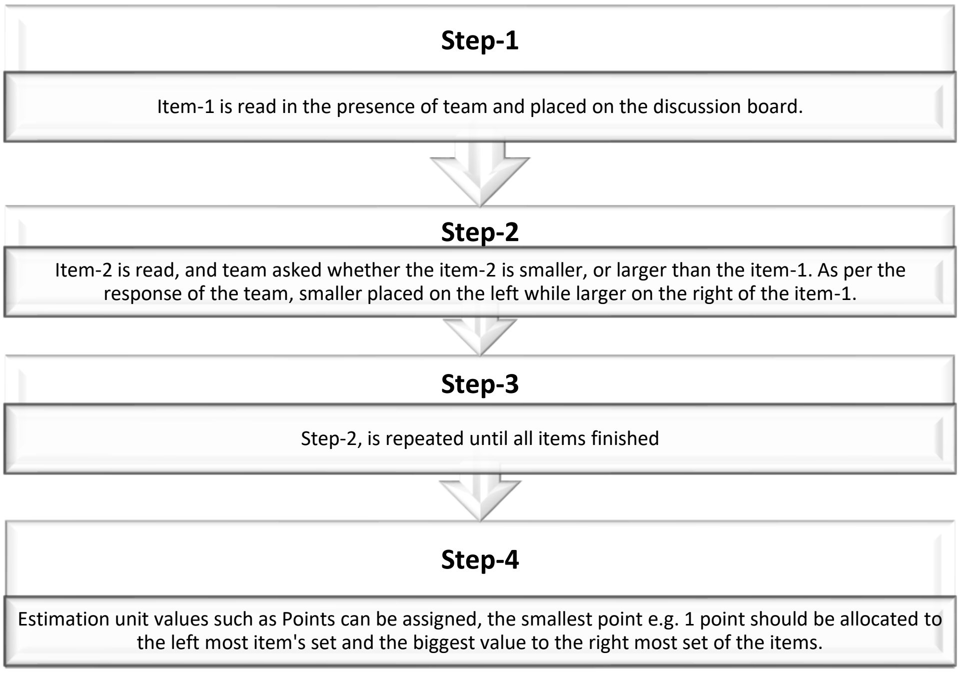


Figure 3.2 Simple Steps of Applying Affinity Grouping

Affinity grouping can also be applied on the items by using the T-shirt sizing technique, another methods of estimation in ASD (Sliger, 2012: Raslan et al., 2015; Alostad et al., 2017). This method utilizes the idea of applying different T-shirt sizes, i.e., Extra-Small (XS), Small (S), Medium (M), Large (L), and Extra-large (XL), to like-sized sets of the items, as shown in Figure 3.3.

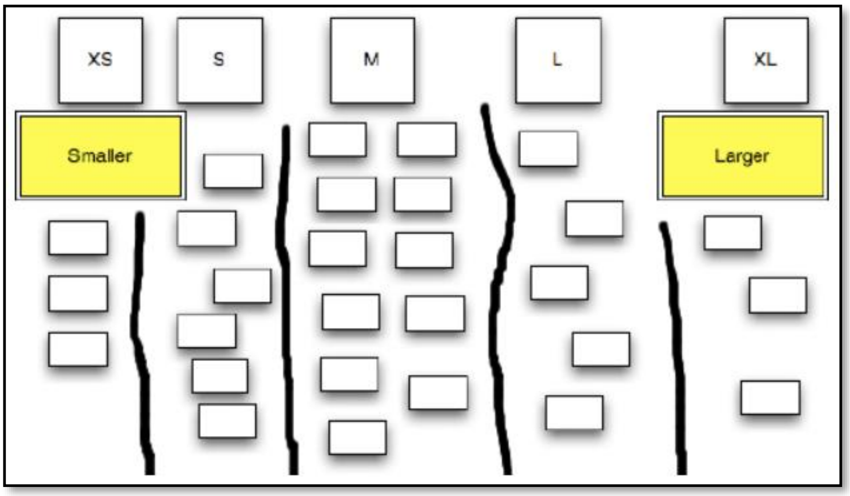


Figure 3.3 Affinity Grouping using T-shirt Sizing (Sliger 2012)

3.3.3 The Bucket System

The bucket system is said to be a particularly efficient technique when a sizable number of items /stories are to be estimated by a larger number of participants. In this method, various buckets of different sizes are created, e.g., 0, 1, 2, 3, 4, 5, 6, 7, 8, 13, 20, 50, 100, 200, and then the like-sized items are placed in the relevant buckets of the matching sizes, as exhibited in the Figure 3.4.

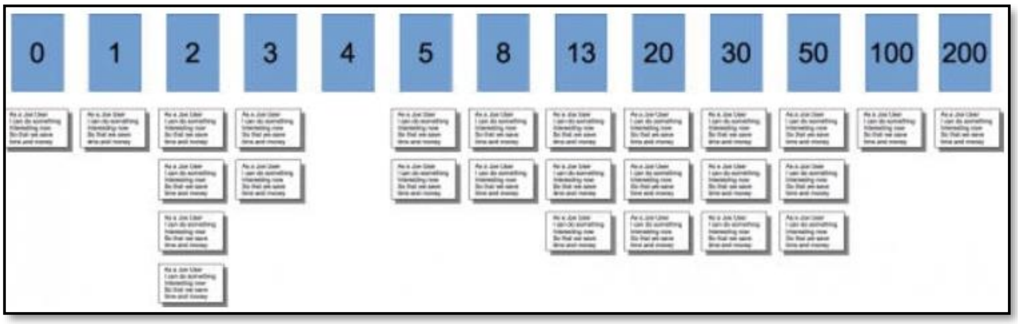


Figure 3.4 The Bucket System Sample (Py, 2019)

3.4 ESTIMATION METHODS COMPARISON

The above review highlights the diversity of a subset of the approaches that have been taken by researchers, and sometimes practitioners, when seeking to estimate aspects of software projects. Table 3.2 summarizes the strengths and weaknesses of the major method classes.

Method / Category	Strengths / Merits	Weaknesses / Demerits
Analogy Method	• Based on the realistic experience of estimators for past projects / products (Shepperd et al., 1996) • Tendency to account for all software factors • Easy to put into practice and superior than some other methods (Shepperd et al., 1996; Mendes et al., 2003-4)	• Relies on the historical data availability and the experience of the estimator (Shepperd et al., 1996) • In some situations, experience / data of a similar project is unavailable • Less robust especially when dealing with the heterogeneous data sets (Briand et al., 1999)
Expert Judgment	• Commonly used approach in software industry (Jørgensen and Sjoberg, 2006; Jørgensen, 2004; Shekhar and Kumar, 2016; Ivan & Despa, 2016; Damm, & Börstler, 2018; FERNÁNDEZ-DIEGO et al., 2020) • Accuracy expected due to shared estimates • Comparatively quick prediction (Cuadrado-Gallego et al., 2010) • Simple to apply (Cuadrado-Gallego et al., 2010)	• Dependent on experts and hence no standard method is being followed (Jørgensen, 2005). • Estimates cannot be justified with verifiable evidence and can exhibit bias because this relies on expert's past experience on similar projects (Khuat and Le, 2016, Cuadrado-Gallego et al., 2010)

Top-Down	• Requires only limited project details (Moløkken, 2002). • Fast & easy to implement / use	• Could be unstable if the estimators have no or limited experience of very similar projects (Jørgensen, 2003)
Bottom-Up	• Utilizes more project details which improves project insight/awareness (Jørgensen, 2003) • More stable & accurate (Moløkken, 2002)	• More effort / time consumed in its practice (Jørgensen, 2003) • System / project level cost ignored
Algorithmic Traditional (COCOMO, UCP, FP, etc)	• Possible to calibrate on the available data of the past completed projects (Keung, 2009) • Developers can be trained, and methods can be used by advanced or novice teams • Objective based, analyzable formula and can be replicated • Additional expertise is not required to put into practice because of simple linear / mathematical models	• Adjustments of the models' formulas are only possible with past completed projects (Prakash and Viswanathan, 2017) and for future projects there is no guarantee of suitable calibration • Not suitable for all types of projects (Ren and Yun, 2013). • Not frequently used for agile development context (FERNÁNDEZ-DIEGO et al., 2020) • Some of the models require additional data as compulsory input to make them operational, which are sometimes unavailable or require more time to prepare or gather
Algorithmic Non-traditional (Proposed Simple algorithm-based Model for Agile Development)	• Suitable for agile development context (Beck et al., 2015; Huang and Chiu, 2007, Ziauddin et al., 2012; Schmietendorf et al., 2008) • Objective based, analysable formula and can be replicated • Possible to calibrate on the available data of the past completed projects • Developers can be trained, and methods can be used by advanced or novice teams • Additional expertise is not required to put into practice because of simple linear / mathematical models	• Requires empirical validation because they are new in the industry • A few models do not take into consideration some of the critical factors, e.g., culture, work environment, project's risks, sprint, technique, which otherwise can impact a project's success / team performance / productivity • Some of the models require additional data as compulsory input to make them operational, which are sometimes unavailable or require more time to prepare or gather.

AI/Machine Learning Based (New Models, or Traditional Models modified by using AI / Machine Learning / Technology)	• Relatively more accurate as shown by some empirical evidence (Mahajan et al., 2011; Kaur et al., 2012) • Combination of two approaches therefore less chance of estimation failure (Ren and Yun, 2013; Jørgensen et al., 2009)	• Cost effectiveness is unknown. • Complex to put into practice • Additional expertise required to use them effectively

Table 3. 2 General Comparison of Estimation Methods

The weaknesses of these models or methods as reported in the above table necessitate the construction of a new model that should exhibit features and characteristics so that most of the above stated shortcomings are avoided or resolved. Those features and characteristics should mean the new model is: dynamic in nature, i.e., capable of adjusting the team's effort estimates for agile and non-agile projects; able to provide estimates on its own for ASD projects, e.g., in a situation where a team is new and/or migrating to agile development; configurable based on improvements or changes in team expertise, experience, relationship, culture and/or team reflections per sprints; using fewer cost drivers as compared to prior models so that the team may readily complete the model's data entry requirements; based on simple and transparent rules; using cost drivers relevant to agile development; supporting the team in their estimation activities without replacing them; easy to understand and implement; and could provide effort estimates for multiple types of systems.

3.5 Chapter Review

In this chapter, contents drawn from literature reviews reflecting the perspectives of industry and academia in the domain of estimating effort, cost and scheduling were elaborated. Further, numerous methods/models for effort/cost estimation and their comparisons, along with their categorizations in algorithmic, non-algorithmic, and hybrid groupings, were discussed.

In next chapter, conceptual, and functional views of COEEMO, the Constructive Effort Estimation Model, are discussed. Moreover, the model's various components along with their detailed descriptions, prototype instances, and a number of scenarios for its implementation, are presented.

Chapter-4 Proposed Parametric Model

***Content Summary**: This chapter describes the conceptual and functional views of the proposed model. It further encompasses explanations of the essential components of the intended model along with aspects of its proposed implementation and calculations of various components. Moreover, prototype's description for the proposed model along with the various scenarios of its implementation are provided.*

The previous chapter established the need for a new model of effort estimation that is suitable for use in agile software development (ASD) contexts. The framework of this model is based on the factors that were discussed in Chapter-2 under the core categories of Human, Organization, System, Sprint, and Risk. The simple functional view of this model is shown in Figure 4.1 within the SCRUM framework but in principle it could be used by non-agile teams who are using RUP or any other development methodology, with or without a sprint-based development approach.

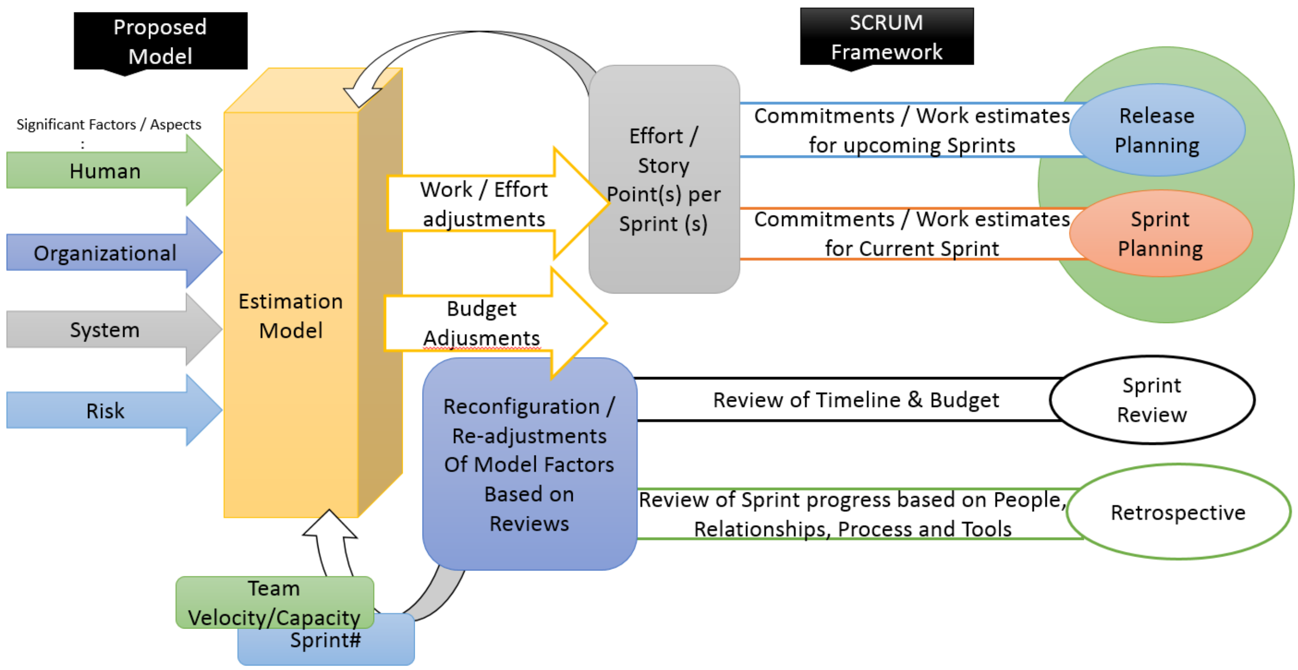


Figure 4.1 Conceptual/Functional Schematic View of the COEEMO aligned with the SCRUM Framework

The proposed model in the figure is represented by a box that needs to be configured based on the categories of factors as shown prior to being utilized for estimating effort in a sprint / iteration. On the other side of the figure a SCRUM framework is shown where the team follows their existing estimation activities and concludes the number of stories and total story-points to be implemented in the upcoming sprint / iteration. The number of story-points arrived at by the team is then input to the proposed model, which then provides suggested adjustments to the story-points based on the data that the team has already set for the critical factors considered under each category of this model. Finally, the adjusted story-points are reported to the team for the purpose of implementing their upcoming sprint / iteration.

The categories as shown in the conceptual view are suggested because each has a distinctive single or group of critical factors that could impact the accuracy of effort estimates, as revealed by numerous prior studies (and detailed in previous chapters). Some of these factors are considered critical for the proposed model and, therefore, form the basic components of the proposed model. These factors are subdivided into 3-4 levels based on subjective criteria and then each level comprises a set of ratings based on a scale of five-point fuzzy values ranging from "Very High" to "Very Low". The core objective is to create various levels of each factor to accommodate multiple scenarios of a team (e.g., their relationships, experiences, and expertise), the organization's culture and working environments, the project's risks, systems' / sprints' complexities, and stakeholder coordination. Further, five fuzzy scale ratings are used to make the model more flexible and practicable, enabling practitioners to assign each factor with a sufficiently granular set of values from "Very High" to "Very Low". Some of the previous studies have used fuzzy sets for constructing their estimation models to improve estimation accuracy in ASD contexts (Ziauddin et al., 2013; Abeer, 2014; Raslan et al., 2015) because through this approach, it is possible to capture the interval details of ordinal variables in an open response format, helping to reduce information loss and/or its distortion during measurement (Li, 2013).

The initial framework of the proposed model is designed based on the outcomes of the prior state-of-art and state-of-practice. Further, the initial values of the ratings for fuzzy scales are given intuitively based on insights from some of the previous non-algorithmic models, e.g., COCOMO, UCP, Function Point Analysis, and the like. However, they are validated and modified later based on the significance of the envisioned factors as conveyed by 21 IT practitioners during interviews (and where ratings for all factors are provided in Chapter-8).

The model proposed here is referred to as the **Co**nstructive **E**ffort **E**stimation **Mo**del **(COEEMO).** This model estimates effort for ASD projects by considering a range of parameters (stated in section 4.1 of this chapter) supported by a number of traditional and contemporary estimation techniques, i.e., Expert judgement, PERT, Top-Down, Bottom-up, and planning poker. Detailed significant factors, features and scenario-based implementations of COEEMO are given in the subsequent sections.

4.1 Significant Factors for COEEMO

The potentially significant factors considered for COEEMO are reviewed in the following subsections under the main categories and subcategories as depicted in Figure 2.2 of Chapter-2. Each has its own proposed description and scale(s) to capture responses for its quantification. To validate the proposed descriptions, their various levels along with associated scales, interviews with IT practitioners are conducted (as reported in Chapter 8).

4.1.1 Human Context

Numerous factors under this category are discussed in previous chapters which might impact estimations; however, the following four human aspects are selected as being particularly significant based on the sources as explained in section 2.1 and the list of significant factors as shown in figure 2.2 of Chapter-2.

4.1.1.1 Team Members' Expertise

Wright & Bolderm (1992) describe the understanding of expertise as one of the significant goals for cognitive science in the context of both theoretical and practical motives. It is closely related to the structure of differences in knowledge, demonstration, decision-making, and a variety of other cognitive competencies; in practice this underpins an ability to classify and use experts in a wide range of real-world scenarios (e.g., an expert may be used to predict project scheduling).

In this proposed model, expertise means knowledge or skill that is acquired through practice irrespective of its duration. It is suggested to be measured across four levels of expertise, which are given below, based on the team member's domain-specific knowledge (meaning the demonstrable academic and/or practical and/or application knowledge in a relevant area of Information Technology, ICT, Computer Science, Software Engineering or equivalent), proficiency with development tools (i.e. how many relevant development tools and technologies they have used), and expertise to estimate/develop software projects (i.e., how many software projects have they estimated/developed). The detailed qualification / description for each level of expertise is stated as follows:

i. High Expertise: To be eligible for this level a person should have established sufficient theoretical and practical skills in Software development (e.g., desktop / mobile / web / cloud-based platform and others) and should also be proficient in a minimum of 8-10 development / design tools / technologies (e.g., Dot Net, Java, and others). Further, s/he should have been involved in the estimation and/or development of more than 10 software projects / products in agile and/or non-agile context(s).

ii. Medium expertise: To be eligible for this level a person should have established sufficient theoretical and practical skills in Software development (e.g., desktop / mobile / web / cloud-based platform and others) and should also be proficient in a minimum of 4-7 development / design tools / technologies (e.g., Dot Net, Java, and others). Further, s/he should have been involved in the estimation and/or development of 5-10 software projects/products in agile and/or non-agile context(s).

iii. Adequate expertise: To be eligible for this level a person should have established sufficient theoretical and practical skills in the Software development (e.g., desktop / mobile / web / cloud-based platform and others) and should also be proficient in a minimum of 1-3 development / design tools / technologies (e.g., Dot Net, Java, and others). Further, s/he should have been involved in the estimation and/or development of 1-4 software projects / products in agile and/or non-agile context(s).

iv. Inadequate expertise: To qualify for this level a person should have established necessary theoretical and/or practical skills in the Software systems development (desktop / mobile / web / cloud-based platform and others).

Note: The ratings (from Very High to Very Low) in all the "Team Member's Expertise" levels are depended upon the level of their practical skills, proficiency in the number of the development tools, and number of the estimated and/or developed software projects/products e.g. "Very High" rating is assigned when a team member has demonstrated an exceptionally high level of expertise in all the above stated areas for the relevant selected level / subcategory.

4.1.1.2 Team Members' Experience

In a study the correlation between teams' experience and the total estimated cost of a software project was considered (Macdonald, 2005). The analysis indicated that such estimates are dependent upon two aspects of team experience: (1) the average experience of the members of each team, and (2) whether team members have or have not had 'similar' project experience – though a concept such as 'similar' may be challenging to define in operational terms. Further, one more study found that the type of experience of a team member may be more important than the total amount of their experience. Moreover, each developer's knowledge and experience are among the significant factors that may affect the magnitude as well as the accuracy of the estimation (Tanveer et al., 2017).

For this proposed model, experience means knowledge or skill that is acquired or practiced over a number of years. It is suggested to be measured across four levels as demonstrated below based on the team member's type and length (duration) of experience. Type of experience includes knowledge and hands-on working in any one or more of the activities of a software project: requirements' engineering, designing, architecting, code development, testing, quality assurance, system support, maintenance, estimation and scheduling. The detailed description of experience for each level/subcategory is given as follows:

i. Proficient: To qualify for this level a person should have working experience of 10 years or more in estimation and/or scheduling of software project(s) and/or in other software development activities which include but are not limited to: requirements' engineering, designing, architecting, code development, testing, quality assurance, system support, and/or maintenance.

ii. Experienced: To qualify for this level a person should have working experience of 6-9 years in estimation and/or scheduling of software project(s) and/or in other software development activities which include but are not limited to: requirements' engineering, designing, architecting, code development, testing, quality assurance, system support, and/or maintenance.

iii. Intermediate: To qualify for this level a person should have working experience of 2-5 years in estimation and/or scheduling of software project(s) and/or in other software development activities which include but are not limited to: requirements' engineering, designing, architecting, code development, testing, quality assurance, system support, and/or maintenance.

iv. Beginner: To qualify for this level a person should have working experience of 0-1 years in estimation and/or scheduling of the software project(s) and/or in other software development activities which include but are not limited to: requirements' engineering, designing, architecting, code development, testing, quality assurance, system support, and/or maintenance.

Note: The ratings (from Very High to Very Low) in all the "Team Member's Experience" levels are depended upon the number of years working and the number of estimated and/or developed software projects/products e.g., "Very High" rating is assigned when a team member has exhibited an exceptionally high level of the experience in all the above stated domains for the relevant selected subcat / level.

4.1.1.3 Team relationship and spirit

The Oxford English Dictionary defines team spirit as "feelings of camaraderie among members of a group, enabling them to cooperate and work well together". Kozlowski & Bell (2003) describe teams as "(a) composed of two or more individuals, (b) who exist to perform organizationally relevant tasks, (c) share one or more common goals, (d) interact socially, (e) exhibit task interdependencies (i.e., workflow, goals, outcomes), (f) maintain and manage boundaries, and (g) are embedded in an organizational context that sets boundaries, constrains the team, and influences exchanges with other units in the broader entity".

According to Stavru (2014) agile development approaches are in use in 50% of software projects world-wide and therefore the effective teamwork of agile teams has much, and growing, significance (Chow & Cao, 2008) and is of genuine interest to practitioners (Gregory et al., 2015). However, conflicting views exist regarding how effective teamwork can be best achieved in an agile context. For example, one view is that effective teamwork can be achieved through the adoption of agile practices whereas another view states that practice adoption alone is not sufficient to enable effective teamwork (Strode, 2015). To address these different perspectives, it is important to have a clear idea of the teamwork theories (e.g., Dickinson & McIntyre's theory, Team Mental Model theory, Big Five theory and others) that are appropriate in an agile development context. For instance, Moe & Dingsøyr in 2008 applied the well-grounded Big Five teamwork theory (BFTT) in their study and found all concepts of BFTT suitable for agile projects with the exception of only one concept “i.e., team leadership” due to the self-organised nature of agile teams.

The Big Five teamwork theory (BFTT) framework was established by Salas et al. in 2005. Strode (2015) relabelled one of the concepts from “team leadership’ to “shared team leadership” because some research has indicated that the “team leadership” concept of BFTT is not applicable in a self-organised agile software development context. Overall Strode concludes that the existing and adapted BFTT theory is applicable in some cases but is not universally applicable due to the variable adoption of agile practices (or potentially other factors). Having said that, Strode (2015) supports the selection of the BFTT framework over others due to the extensive underpinning review of literature covering 20 years of research and involving the analysis of 121 references from multiple contexts, its prior use in agile contexts, and its positive impact in the literature (Google Scholar has shown 2095 citations (March 2021)).

For this intended model, the modified BFTT framework as presented by Strode (2015) is used to measure the effectiveness of teamwork. It is measured across five levels (i.e., Excellent, Very Good, Good, Satisfactory, and Poor / conflicting / challenging) based on each of the concepts of BFTT as described in Table 4.2.

Traits	Concepts	Description
1	Adaptability	Ability to adjust strategies based on information gathered from the environment through the use of backup behavior and reallocation of intra-team resources. Altering a course of action or team repertoire in response to changing conditions (internal or external).
2	Backup behavior	Ability to anticipate other team members’ needs through accurate knowledge about their responsibilities. This includes the ability to shift workload among members to achieve balance during high periods of workload or pressure.
3	Closed-loop communication	The exchange of information between a sender and a receiver irrespective of the medium.
4	Mutual performance monitoring	The ability to develop common understandings of the team environment and apply appropriate task strategies to accurately monitor teammate performance.
5	Mutual trust	The shared belief that team members will perform their roles and protect the interests of their teammates.
6	Shared mental models	An organizing knowledge structure of the relationships among the task the team is engaged in and how the team members will interact.
7	Team leadership (relabeled Shared Team leadership in Strode (2015))	Ability to direct and coordinate the activities of other team members, assess team performance, assign tasks, develop team knowledge, skills, and abilities, motivate team members, plan and organize, and establish a positive atmosphere.
8	Team orientation	Tendency to take other’s behavior into account during group interaction and the belief in the importance of team goal’s over individual members’ goals.

Table 4.2 Big Five Team Theory Framework (Salas et al., 2005)

According to the above given table, BFTT has uses eight concepts (with their descriptions) to measure team effectiveness. For COEEMO, these eight concepts are treated as eight traits of a team, and they are ordered from trait 1-8. The concept given in the first cell of the above table represents trait-1 of the team whereas the second cell represents trait-2 of the team, and so on. Based on the number of given traits a team can be assigned different levels / subcats (i.e. Excellent, Very Good, Good, Poor / struggling / challenging) while the level of compliance against the given traits will decide the rating level (i.e. Very High to Very Low) within the assigned subcats. For example, if a team has all eight traits as shown in Table 4.2, then the team can be given “Excellent” subcat but the ratings (i.e. from Very High to Very Low) within the “Excellent” subcat will depend on the level of compliance of the team to the exhibited traits (i.e. for a very high level of compliance of the team for all traits, the team can be assigned a “Very High” rating within the “Excellent” subcat or “Very Low” otherwise in the same if the team has variations in the compliance with the given traits).

Keeping in view these eight traits team effectiveness is categorized into four levels, which are explained as follows:

i. Excellent: A team is given “Excellent” level if the team shows all eight traits.
ii. Very Good: A team is allocated “Very Good” level if the team indicates all eight traits with slight deviations in trait 6 or 7.
iii. Good: A team is assigned “Good” level if the team shows any 6 traits without any deviations in traits 1, 2, 4, 5, and 8.
iv. Poor / Conflicting / Challenging: A team is assigned “Poor” level if the team indicates that it is lacking in any 3-4 traits and/or missing traits i.e., 1, 2, and 5.

Note: The ratings (from Very High to Very Low) in all the “Team Relationship & Spirit” subcats are depended upon on the level of the compliance which the team demonstrates against the given traits.

4.1.1.4 Stakeholder/User Collaboration

Agile practices lend importance to direct and regular contact between the clients (and/or their representatives) and the development team. These forms of contact (e.g., Daily stand-up meetings, release and iterative planning) are said to be of help in building a positive relationship between client and team. More concretely, this relationship gives an opportunity to the team to discuss user requirements and project progress with the client and to get the client’s feedback and/or evaluate their success (Tripp et al., 2016).

For this intended model, Stakeholder/User collaboration refers to the relationship between team and client and it is measured across four levels (Fully, Partially, Occasionally, and Poorly) based on the type of contact (e.g., one to one, through email, Skype or other channels), frequency of visits/interactions, and the feedback level (e.g., detailed or incomplete) of the client stakeholders. The details of each level for stakeholder / user collaboration are given as follows:

i. Fully: This level is designated if a stakeholder / user collaborates fully with the development team by attending all the scheduled meetings and also respects completely to other agreed upon means of correspondence / communication for the intended project. Moreover, s/he provides thorough feedback as and when is required.

ii. Partially: This level is assigned if a stakeholder / user collaborates partially with the development team by attending some of the scheduled meetings without missing the important ones (e.g., kick-off, release planning.) and also respects completely the other agreed upon means of correspondence / communication for the intended project. Moreover, s/he provides thorough feedback as and when is required.

iii. Occasionally: This level is allocated if a stakeholder / user collaborates occasionally with the development team by attending some of the scheduled meetings but also disregards some of the important ones (e.g., kick-off meeting, release planning). Further, s/he sometimes respects the other agreed upon means of correspondence / communication for the intended project and also provides delayed or on time and/or thorough or incomplete feedback frequently or occasionally.

iv. Poorly: This level is assigned if a stakeholder / user does not attend any of the scheduled meetings or attends some occasions but also disregards most of the important ones (e.g., kick-off meeting, release planning) and does not respect other agreed upon means of correspondence / communication for the intended project. Moreover, s/he provides delayed or on time and/or thorough or incomplete feedback frequently or occasionally.

Note: The ratings (from Very High to Very Low) in all the “Stakeholder / User Collaboration” subcats are depended upon on the level of the collaboration, response, and feedback of the stakeholder for the intended project(s).

4.1.2 Systems Context

This factor is intended to capture the challenges encountered due to the implementation of leading technologies and/or complex functionalities. Evidence suggests that military and mission-critical systems are typically more complex than general purpose information-based systems, and hence require relatively more effort and cost to implement (Popli & Chauhan, 2014) – all other things being equal. This aspect is subdivided into different subcats as shown below. These subcats cover the various types of system to gauge their complexity levels (i.e., from Very High to Very Low) with respect to their implementation:

a. Sys-1 (Defence / Military / Mission Critical Systems)
b. Sys-2 (Decision Support / EIS / Enterprise Systems / Embedded Systems / IoT / Machine Learning / Data Analytics)
c. Sys-3 (Financial / Banking Systems)
d. Sys -4 (General Purpose Systems other than financials)

Note: All the above stated subcats include the multiple platforms of systems e.g., Web, Mobile, Cloud, Desktop and others.

4.1.3 Organization Context

Organization structure and culture both affect capability to accomplish project success. The culture of an organization is a set of shared values, behaviours, and assumptions that describes its functional view comprising 10 characteristics: member identity, group emphasis, people focus, unit integration, control, risk tolerance, reward criteria, conflict tolerance, means-ends orientation, and open-systems focus. To capture the organization context, three important aspects, i.e., organization type, culture, and working environment, are considered by the COEEMO. Their details are given in the following subsections.

4.1.3.1 Organization Type

It seems plausible that organization type, including government, private, and joint, could influence a project directly or indirectly; e.g., in a government organization the strong bureaucracy may restrict frequent meetings with project stakeholders, or top management may be unwilling to support any specific project or team over concerns regarding probity. The organization type is divided into three subcategories which includes: Government / Public sector (e.g., government departments or ministries, public limited company and corporations), Private sector (private limited companies, individually owned organizations), and Joint ventures (e.g., public and private entrepreneur).

The ratings (i.e., from Very High to Very Low) for the different types of organization will depend on the structure, control, policies and similar parameters that are implemented in the organizations to execute their day-to-day operations e.g., if an organization belongs to the “Private” category but it has long bureaucratic channels, centralized control and strict policies just like a government organization, then it may be assigned a “Very Low” rating.

Note: The ratings from “Very High” to “Very Low” can be assigned against the above given subcats of the organization depending upon the organization structure (i.e., from big to small), policies (i.e., from strict to soft), procedures (i.e., from long to short), control mechanisms (i.e., from centralized to distributed) and similar parameters. For example, the rating “Very High” can be assigned to a small, structured organization that has soft procedures for approvals of its projects’ / teams’ requirements, and/or has a distributed control mechanism.

4.1.3.2 Organization Culture

Organizations may have established values and beliefs that in turn can have positive or negative impacts on teams and their projects. Teams may seek to adhere to, or work against, strong organizational values. The organization culture is suggested to be measured across four levels (i.e., Strong, Semi-Strong, Fragile, Challenging /Toxic). The details for each level are stated below:

i. Strong: This level is assigned when the organization follows an open and robust culture where values and beliefs are strong and there is freedom of expression and other positive traits i.e., team cohesiveness, mutual trust, team autonomy (Sharma & Gupta, 2012).
ii. Semi-Strong: This level is allotted when most of the qualitative traits as given in the "Strong" culture category are being followed with some exceptions.
iii. Fragile: This level is assigned when the organization follows a culture where values and beliefs are weak; there is a lack of mutual trust, and restricted freedom of expression.
iv. Challenging/Toxic: This level is appropriate when the organization follows a culture where values and beliefs are very weak; the workplace is afflicted by fighting and/or drama; there is an extreme lack of mutual trust; no freedom of expression; creating unhappy employees.

Note: The ratings (from Very High to Very Low) in all the "Culture" categories can be assigned "Very High" if the organization culture in that specific level is inclined enormously towards the good as compared to the bad traits and so on.

4.1.3.3 Work Environment

The work environment may have direct and indirect impacts on project outcomes; e.g., it can directly influence team performance progression, and job continuity, while quality and development time can be impacted indirectly (Schwalbe, 2014; Sharma & Gupta, 2012). The work environment is suggested to be measured across four levels (i.e., Best, Semi-Best, Good, and Poor). The details for each level are stated below:

i. Best: This level of the work environment is allotted when the work environment is characterised by traits such as these: people operate as individuals, but still focus on succeeding as a team; honest feedback is favoured; it helps to keep the team composed; it offers employees flexibility to customize their own working styles; everything may be questioned because questions are encouraged (Wroblewski, 2020; B, 2021).

ii. Semi-Best: This level of the work environment is assigned when most of the qualitative traits as given in the "Best Work Environment" are being followed, with a few exceptions.

iii. Good: This level of the work environment is allocated when some of the qualitative traits as given in the “Best Work Environment” are being followed.

iv. Poor: This level is assigned when the work environment exhibits traits such as: having strict timetables, dress codes, protocols, and operations; it restricts the creativity and independence of team members; forces individuals into silos; there is nothing except success or failure; rewarding good behaviour but punishing bad behaviour; bosses cannot be asked, leaders cannot be confronted (Williams, 2017; Ishak, 2016).

Note: The ratings (from Very High to Very Low) in all the “Work Environment” levels can be assigned “Very High” if the “Work Environment” of the organization in that specific level is inclined immensely towards the good as compared to the bad traits and so on.

4.1.4 Iteration / Sprint Context

It seems intuitive to expect that estimates and plans will be generally less accurate early in projects when information and certainty are at a premium, and that accuracy should improve as projects progress (all other things being equal). This aspect is also considered to be potentially important for the proposed model of estimation. It is assumed that the first sprint of any new project will need more effort than the succeeding sprints due to the longer time consumed by the team in initial learning and understanding of the user requirements and also in developing the first blueprint of the basic architecture of the system. The sprint context is suggested to be measured across the three subcategories (i.e., Spt1Cpxy, Spt2Cpxy, and SptNCpxy). The details for each subcat are stated below:

i. Spt1Cpxy – Spt2Cpxy: This subcat is used to assign the complexity level for Sprint-1 to Sprint-2 from the 5 given ratings (i.e., “Very High” to “Very Low” complexity).

ii. SptNCpxy: This subcat is used to assign the complexity level for Sprint-3 or more from the 5 given ratings (i.e., “Very High” to “Very Low” complexity). The sprint’s complexity for the 3rd or later sprints can be calculated by the formula as given in the section 4.4.5.1 of this Chapter. Further, if a project / product develops in only 3 sprints then the complexity rating for its last sprint is suggested to be selected from the SptNCpxy subcategory.

4.1.5 Risk Context

Unmanaged risks are one of the (albeit numerous) reasons for the failure of software projects. Some of the risks linked with project failures are uncertainty regarding project complexity, project goals, personality differences, and others. This aspect also has the potential to impact on estimation. It is assumed that some risks are known and so they will be mitigated during the life of the project. However, some risks are unknown and therefore cannot be mitigated in advance, including the resignation of an active member of the team, long leave due to illness, and so on. This aspect is measured in 3 subcats (i.e., Planned Risks, Unplanned Risks, and Unknown Risks). The impact ratings for all these subcats of the risks on the intended project(s) can be selected from “Very High” to “Very Low”.

4.2 COEEMO’s Conceptual view

COEEMO’s conceptual view is very simple, and it provides a value to adjust the effort estimations of the team for the upcoming sprint. This value is calculated by the model based on the 10 significant factors which are covered in the preceding sections of this chapter and on the selection of the estimation techniques, e.g., Expert judgement, PERT, Top-Down, and Bottom-up. These factors are divided into five core categories/contexts: i.e., Human, System, Organization, Sprint, and Risk as explained in the previous section. A complete list of the categories along with their detailed ratings and weighted values are provided in Table 4.1.

4.3 COEEMO’s Functional view

The core formula of the proposed model, significant keywords, sub-parts, and the abbreviations used to exhibit the formula are all described in this section. Moreover, the abbreviation “wt.” is used frequently for the word “weighted” during the functional description of the COEEMO.

4.4 COEEMO’s Components

4.4.1 COEEMO Effort Adjustment Factor (CEAF)

CEAF is the sum of the effort factor (EF) and the learning factor (LF). This is the final value that the model computes for the estimator. The estimator then calculates the Adjusted Effort Estimation (AEE) by taking the product of CEAF and Estimated Effort (EE). CEAF is a multiplier (for ASD) / divisor (for TSD) and does not represent any unit. Therefore, the unit of the estimation (e.g., story points, or ideal person day, or hours, and so on) depends upon the priority of the estimator. Here EE is the estimation of Story points, or ideal days, or hours that the team estimates to start their development activities (normally called project / sprint development schedule).

4.4.2 Technical Factor (TF)

The model computes TF by considering the selected complexities of the product / system (SysCpxy), sprint (SprintCpxy) and risks. Three types of risks are used to compute risk complexity i.e., Planned risk (PlanR), unplanned risk (UplanR), and unaware risk (UnR). The selected wt. values of all these parameters are used in the following base formula to calculate the TF:

$TF = 0.7 + 0.02 * ((SysCpxy) + (\sum_{i=1}^{j} \frac{PR_i}{j} + \sum_{k=1}^{l} \frac{UR_l}{l} + \sum_{m=1}^{n} \frac{UnR_m}{n}))$ --------- Equation-1

Where:

- TF is the Technical Factor
- SysCpxy is the System Complexity

- $\sum_{i=1}^{j} \frac{PR_i}{j}$ is an average rating for all the Planned Risks (i.e., Known mitigated Risks) of the sprint / project from $i\ to\ j$
- $\sum_{k=1}^{l} \frac{UR_k}{l}$ is an average rating for all the Unplanned Risks (i.e., Known unmitigated Risks) of the sprint / project from $k\ to\ l$
- $\sum_{m=1}^{n} \frac{UnR_m}{n}$ is an average rating for all the Unaware Risks (i.e., Unknown Risks) of the sprint / project from $m\ to\ n$

Note: The values i.e., 0.7 and 0.02 in the equation are used for model fitting.

4.4.3 Environmental Factor (EnvF)

EnvF is calculated by using the formula as stated in Equation-2. This can be used in situations where estimations are required for the upcoming sprint / projects i.e., in a sprint planning meeting for agile and/or non-agile projects (with or without sprints/cycles e.g., RUP or waterfall) where iterations/sprints/cycles are considered for the software development work e.g., RUP. It may also be applied in a situation where initial effort estimation is required for the ASD project for the purpose of bidding and/or to communicate approximate likely timelines to the customer. Further, this formula can also be utilized for projects that typically do not follow development in iterations/sprints/cycles e.g., waterfall. In the both equations (i.e., Equation-1 and -2), EF is computed by considering the selected wt. values of the factors that include sprint complexity (SprintCpxy) when development work is in iterations/sprints, team expertise (TExpert), team experience (TExp), team relationship (TR), stakeholder coordination (StkCoord), organization type (OrgT), environment (OrgEnv), and culture (OrgCul). Moreover, some intermediate values are also computed based on the above stated factors that are utilized later to compute the value for the Environmental Factor. These intermediate values are teams' capability (TCap), productivity (TPrd), human impact (HI), and the organization efficiency index (OEI). The base formulas to calculate the Environmental factor are given as follows:

EnvF= 1.3 + (-(0.05) *(Tcap + TPrd + OEI + HI)) --------------------------- Equation-2

Where:

- HI is the sum of the team expertise (TExpert), team experience (TExp), stakeholder coordination (StkCoord), and the team relationship (TR) which are calculated by the following formulas:
    - TExpert = (∑wt. values of each team member expertise) / Total no. of team members
    - TExp = (∑wt. values of each team member) / Total no. of team members
    - StkCoord = (∑wt. values of each stakeholder coordination) / Total no. of stakeholder
    - TR = Selected wt. value by the team and/or team Lead for the team spirit and relationship

- Tcap is computed by this formula: TCap = (0.3 * TExpert + 0.5 * TExp) * TR
- OEI is calculated by taking the sum of the selected wt. values of the organization type, culture, and environment.
- TPrd is computed by this stated formula: TPrd =0.25*(TCap + OEI)

Note: The values i.e., 1.3 and 0.05 in the equation are used for model fitting.

4.4.4 Effort Factor (EF)

Effort Factor is the product of the Technical and Environmental factors (i.e., TF * EvnF). This factor is used as an intermediate value in the Learning Factor (LF) to calculate the COEEMO Effort Adjustment Factor (CEAF).

4.4.5 Learning Factor (LF)

Learning Factor for agile and non-agile projects is computed by the formulas as stated in the successive sections. The details and efficacy of the learning curve are already covered in Chapter 2 (section 2.9.5).

4.4.5.1 LF Computation for ASD projects

The formula as demonstrated in equation-3 is used to compute LF for ASD projects.

$LF = (1 - EF) * (SptCpxy) * N^b$ ----------------------------- Equation-3

Where:

- EF is the Effort Factor
- b is the slope of LC and is calculated by the formula: Log (Learning Rate) / Log_2
- N is the value calculated by dividing the total number of expected sprints by the current sprint
- SptCpxy: This is the rating as selected by the team keeping in view the complexity of a sprint. The SptCpxy's ratings are given in Table 4.1 for the first two sprints under the sprint category but if the project / product has more than two expected sprints then the formula as stated below will be employed to calculate the complexity of the third and further sprints:

SptCpxy = Spt2Cpxy + ((SptNCpxy- Spt2Cpxy) / (Expected Sprints/ Current Sprint))

Notes:

- Current sprint means the upcoming sprint for which the team intends to generate estimations. (The expected and current sprints must be given as integer values, e.g., in 10/4, 10 represents the total number of the expected sprints while 4 is the current sprint number for which the team wants to estimate.)
- Spt2Cpxy and SptNCpxy are the ratings as given in Table 4.1 under the five scales (i.e., Very High, High, Moderate, Low, and Very Low). These ratings can be chosen by the team based on the complexity of the current sprint. If the team selects SptNCpxy rating from the "Very High" scale then they should choose the corresponding value of the rating for the Spt2Cpxy from the same scale if they do not remember the complexity rating for the sprint 2.

- The value for the Learning Rate (LR) can be considered from 80% to 90% depending upon the learning capability (i.e., fast or slow) of the team. However, the recommended LR for software development processes is 80% (Racoon, 1995). The typical LRs for most of industrial processes are exhibited in Table 2.8.

4.4.5.2 LF Computation for Non-ASD projects

The formula presented in equation-4 is applied to calculate LF for Non-ASD projects.

$LF = (1 - EF) * M^b$ -------------------------------------- Equation-4

Where:

- EF is the Effort Factor
- b is the slope of LC and is calculated by the formula: Log (Learning Rate) / Log_2
- M is the value calculated by dividing the expected duration of the project in Person-Hours by the work hours in a day (i.e., normally 8 hours are contemplated as work hours by most organizations, but this can be modified as per the need of an organization) for TSD projects developed without sprints.

Notes:

The value for the Learning Rate (LR) can be considered from 80% to 90% depending upon the learning capability (i.e., fast or slow) of the team. However, the recommended LR for software development processes is 80% (Racoon, 1995).

	COEEMO's Contexts, Categories, Subcategories, Fuzzy Scales/Ratings					
Contexts	**Category /Subcategory/Factors**	**Fuzzy Scale / Ratings**				
Human		**Very High**	**High**	**Moderate**	**Low**	**Very Low**
	Team Expertise					
	High expertise	0.3	0.34	0.42	0.52	0.64
	Adequate expertise	0.7	0.74	0.82	0.92	1.04
	Medium expertise	1.4	1.44	1.48	1.58	1.7
	Inadequate expertise	1.95	1.99	2.07	2.17	2.20
	Team Experience					
	Proficient	0.2	0.25	0.35	0.5	0.7
	Experienced	0.8	0.85	0.95	1.1	1.3
	Intermediate	1.55	1.6	1.7	1.85	2.05
	Beginner	2.1	2.15	2.25	2.4	2.6
	Team Relationship & Spirit					
	Excellent	1.03	1.06	1.09	1.12	1.15
	Very Good	1.18	1.21	1.24	1.27	1.3
	Good	1.33	1.36	1.39	1.42	1.45
	Poor / Struggling / Conflicting	1.48	1.51	1.54	1.57	1.6
	Stakeholder / User Collaboration					
	Fully	0.25	0.3	0.35	0.4	0.45
	Partially	0.6	0.65	0.7	0.75	0.8
	Occasionally	1	1.05	1.1	1.15	1.2
	Poorly	1.5	1.55	1.6	1.65	1.7
System	**System Types' Complexity**					
	Sys-1 (Defense/Military/ Mission critical System)	0.1	0.25	0.4	0.55	0.7
	Sys-2 (DSS / EIS/ Enterprise/ Embedded/ IoT / Machine Learning / Data analytics)	0.8	0.95	1.1	1.25	1.4
	Sys-3 (Financial/Banking (Mobile & Desktop)	1.5	1.65	1.8	1.95	2.1
	Sys-4 (General Purpose systems other than financials)	2.15	2.25	2.35	2.45	2.5
Org.	**Organization types**					
	*Public / Government Sector	0.9	0.95	1	1.05	1.1
	**Joint Sector organizations	0.6	0.65	0.7	0.75	0.8
	Private Sector	0.3	0.35	0.4	0.45	0.5
	Organization Culture					
	Strong	0.15	0.25	0.35	0.45	0.55
	Semi Strong	0.65	0.75	0.85	0.95	1.05
	Fragile	1.15	1.25	1.35	1.45	1.55
	Toxic / Challenging	1.65	1.75	1.85	1.95	2.05
	Work Environment					
	Best Work Environment	0.25	0.3	0.35	0.4	0.45
	Semi best work Environment	0.6	0.65	0.7	0.75	0.8
	Good work Environment	0.95	1	1.05	1.10	1.15
	Poor work Environment	1.3	1.35	1.4	1.45	1.5
Sprint No.	**Sprint**					
	Spt1Cpxy	0.1	0.12	0.15	0.18	0.2
	Spt2Cpxy	0.3	0.4	0.45	0.5	0.55
	Spt3Cpxy / SptNCpxy	0.75	0.85	0.9	0.94	0.98
Risk	**Risk**					
	Known Mitigated Risks	2.1	2.3	2.5	2.7	2.9
	Known non-mitigated Risks	-0.5	-0.4	-0.3	-0.2	-0.1
	Unknown Risks	-1.5	-1.3	-1.1	-0.9	-0.7

* e.g. Govt., Dept., Ministry and/or public Ltd. Company or corporation, etc

** e.g. Public and entrepreneurs, etc. (private Ltd. Companies and/or individual owned organization)

Table 4.1 COEEMO's Contexts, Categories, Subcategories, Fuzzy Scales /Ratings

4.5 COEEMO's Prototype

A prototype of the model was developed by using VBA in Microsoft Excel. Key screen shots of the prototype are provided in Appendix-A while a soft copy of this program was provided to the team who applied COEEMO during the experimentation part of this study. This application helped the team in collecting data regarding their team and projects.

4.6 COEEMO's Scenario-based Usages & Implementations

In Table 4.1, the complete list of the COEEMO contexts, categories, subcategories and significant factors, along with their ratings, are provided. The complete method of the calculations per single category / context and their use in COEEMO is explained in this section. Further, to illustrate the use of these significant factors and the calculations made using COEEMO's various formulas, and to estimate human effort at the end, example activities are presented here. Before the explanation of the calculating activities and various scenarios, the following concepts are essential to know for more understanding about the various computations of COEEMO:

a. COEEMO's Contexts: For COEEMO, five main contexts (CC) are discovered which impact or influence the human effort estimation for a software project undertaken in an agile development. They are Human, Organization, System, Sprint/iteration number, and Risk.
b. COEEMO's Categories: CC comprises 10 main categories (CCat) which are Team expertise, experience, relationship & spirit, stakeholder coordination, system complexity, organization type, culture, work environment, sprint number, known-mitigated risks, known but unmitigated risks, and unknown (not mitigated) risks. For the purpose of the calculations all these CCat are utilized by the COEEMO model.
c. COEEMO'S's Subcategories: CCat have 37 different subcategories (CSCat) which are considered critical, and their list is given in Table 4.1. The relevant CSCat are identified within each CCat while using COEEMO model.
d. COEEMO''s Ratings: While estimating the effort, each identified CSCat is assigned a single value from the ratings based on the different fuzzy scales / levels i.e., Very High, High, Moderate, Low, Very Low.

4.6.1 COEEMO's Calculations

4.6.1.1 How to calculate Team expertise?

This factor captures the expertise of a team involved in the development of a software project. The method of the expertise computation of a team is very simple and explained in easy steps as given below:

Step-1: Team members (TMs) are assigned their relevant CSCat for team expertise from the given four CSCat of expertise i.e., High, Adequate, Medium, and Inadequate (the qualifications of each expertise level of the CSCat are given in subsection 4.1.1.1 of this chapter). For example, if you have seven team members and you want to calculate the team expertise based on the aggregate expertise of these individual members of the team then they are required to assign / allocate the relevant CSCat of expertise to each of the team members based on their individual qualifications matching to the given criterion of the expertise's CSCat. The allocation of expertise based on the seven individual team members of the current example is shown in table 4.3.

Number of Team Members (TMs) (a)	Allocation CSCat of expertise (b)
1	High expertise
2	Adequate expertise
3	Medium expertise
1	Inadequate expertise

Table 4.1 Allocation of the relevant CSCat of expertise and the no. of TMs

Step-2: The team is then required to select the fuzzy scale against each selected CSCat and the relevant ratings against each chosen fuzzy scale. To illustrate Step-2, table 4.3 is updated by the two additional new columns as shown in table 4.4:

No. of Team Members (a)	Allocations		
	CSCat of Expertise (b)	Grading / Scale (c)	Ratings (d)
1	High expertise	Very High	0.30
2	Adequate expertise	Moderate	0.82
3	Medium expertise	High	1.44
1	Inadequate expertise	Very Low	2.20

Table 4.2 Allocations of relevant Gradings/ratings against selected CSCat of expertise

Finally, team expertise is calculated by multiplying the column (a) and column (d) and then the all product values of the column (e) are added, and further their average is taken as shown in table 4.5. This average value is noted as the team's expertise for the planned / upcoming sprint of the seven members of the team and is used further in COEEMO for the various calculations.

No. of Team Members (a)	Allocations			Team Expertise (e = a*d)
	CSCat of Expertise (b)	Grading / Scale (c)	Ratings (d)	
1	High expertise	Very High	0.30	0.3 * 1 = 0.3
2	Adequate expertise	Moderate	0.82	0.82 * 2 = 1.64
3	Medium expertise	High	1.44	1.44 * 3 = 4.32
1	Inadequate expertise	Very Low	2.20	2.20 * 1 = 2.20
Total Team Expertise= Sum of CSCat expertise/Total number of TMs				8.46/7 = 1.21
Final Value of the Team Expertise for the planned sprint/project				1.21

Table 4.3 Final value calculations of the team expertise

4.6.1.2 How to calculate Team experience?

This factor reflects the experience of a team involved in the development of a software project. The method of the experience computation of a team is very simple and explained as follows:

Step-1: Team members (TMs) are assigned their relevant CSCat for team experience from the given four CSCat i.e., Proficient, Experienced, Intermediate, and Beginner (the qualifications of each experience level are given in subsection 4.1.1.2 of this chapter). For example, for the same seven team members the team is required to assign the relevant CSCat of experience to each of them based on their individual qualifications matching to the given criterion of the experience's CSCat. The allocation of experience based on the individual team members of the current example is shown in table 4.6.

Number of Team Members (TMs) (a)	**Allocation CSCat of experience (b)**
1	Proficient
2	Experienced
3	Intermediate
1	Beginner

Table 4.4 Allocation of the relevant CSCat of experience and the no. of TMs

Step-2: The team is then required to select the fuzzy scale along with the relevant ratings against each chosen fuzzy scale. To illustrate Step-2, table 4.6 is updated with two additional columns as shown in table 4.7.

No. of Team Members (a)	**Allocations**		
	USCat Team experience (b)	**Grading / Scale (c)**	**Ratings (d)**
1	Proficient	Very High	0.2
2	Experienced	Moderate	0.95
3	Intermediate	High	1.6
1	Beginner	Low	2.4

Table 4.5 Allocations of the relevant Gradings/ratings against selected CSCat of the experience

Finally, team experience is calculated by multiplying column (a) and column (d) and then all product values of column (e) are added, and further their average is taken as shown in table 4.8. This average value is recorded as the team's experience for the planned / upcoming sprint of the seven members of the team and is used further in COEEMO for the various calculations.

No. of Team Members (a)	Allocations			Team Expertise (a*d)
	USCat Team experience (b)	Grading / Scale (c)	Ratings (d)	
1	High	Very High	0.2	0.2 * 1 =0.2
2	Adequate	Moderate	0.95	0.95 * 2 = 1.9
3	Medium	High	1.6	1.6 * 3 = 4.8
1	Inadequate	Low	2.4	2.4 * 1 = 2.4
Total Team Expertise= Sum of CSCat team experience/Total number of TMs				9.3/7 = 1.33
Final Value of the Team Experience for the planned sprint / project				1.33

Table 4.6 Final value calculations of the team experience

4.6.1.3 How to calculate Stakeholder Coordination/collaboration?

This factor captures the coordination/collaboration level of the stakeholder(s) of the project.

Step-1: Stakeholder(s) are assigned their relevant CSCat from the given four options of stakeholder coordination (StkCoord) i.e., Fully, Partially, Casually, and Poorly (where the qualifications of each CSCat of StkCoord are given in the subsection 4.1.1.4 of this chapter). For example, if you have three stakeholders of your project and you want to calculate their coordination level then you are required to allocate the relevant CSCat of StkCoord to each of the stakeholders based on their individual qualifications matching to the given criteria. The allocation of coordination level based on the individual stakeholders of the current example is shown in table 4.9.

Number of Stakeholder(s) (a)	Allocation CSCat of StkCoord (b)
1	Fully
1	Partially
1	Poorly

Table 4.7 Allocation of related CSCat of the stkcoord and no. of Stakeholders

Step-2: You are then required to select the fuzzy scale and ratings against each selected CSCat of the StkCoord. To illustrate Step-2, table 4.9 is updated by two additional columns as shown in table 4.10.

No. of Stakeholder(s) (a)	Allocations		
	CSCat of StkCoord (b)	Grading / Scale (c)	Ratings (d)
1	Fully	Very High	0.25
1	Partially	Moderate	0.70
1	Poorly	High	1.55

Table 4.8 Allocations of the relevant Gradings/ratings against selected CSCat of the StkCoord

Finally, Stakeholder coordination/collaboration is calculated by multiplying column (a) and column (d) and then all product values of column (e) are added, and further their average is taken as shown in table 4.11. This average value is the StkCoord for the planned / upcoming sprint and is used further in COEEMO for the various calculations.

No. of Stakeholder(s) (a)	Allocations			StkCoord (a*d)
	CSCat of StkCoord (b)	Grading / Scale (c)	Ratings (d)	
1	Fully	Very High	0.25	0.25 * 1 =0.25
1	Partially	Moderate	0.70	0.70 * 1 = 0.70
1	Poorly	High	1.55	1.55 * 1 = 1.55
Total StkCoord= Sum of CSCat StkCoord/Total number of Stakeholders				2.5/3 = 0.83
Final Value of the StkCoord for the planned sprint / project				0.83

Table 4.9 Final value calculations of the Stakeholder Coordination/Collaboration

4.6.1.4 How to calculate Risks' Impact?

This factor captures the impact of risks on the project/sprint. The method to calculate the impact of risks is very simple and is explained in the steps given below. Further, in case no risk is identified for the planned sprint/project against any of the CSCat of the risk then the "Very Low" rating is suggested to be chosen for that specific CSCat.

Step-1: Risk is assigned their relevant CSCat from the given three CSCat of the risk, i.e., known mitigated, known non-mitigated, and unknown risk (where the qualifications of each CSCat of risk are given in subsection 4.1.5 of this chapter). For example, if a team has six risks for their sprint/project and they want to calculate the overall impact of these risks then the team is required to assign the relevant CSCat of the risk based on risks' individual qualifications matching to the given criterion of the CSCat of that risk. The allocation of the number of risks based on their CSCat of the current example is shown in Table 4.12.

Number of Risks (a)	Allocation CSCat of Risk (b)
3	Known mitigated
2	Known non-mitigated
1	Unknown

Table 4.10 Allocation of related CSCat of the risk and no. of risks

Step-2: The team is then required to select the fuzzy scale along with the relevant ratings against each chosen fuzzy scale. To show Step-2, Table 4.12 is updated with two additional columns as shown in Table 4.13.

Number of Risks (a)	Allocations		
	CSCat of Risk (b)	Grading / Scale (c)	Ratings (d)
3	Known mitigated	Moderate	2.5
2	Known non-mitigated	High	-0.4
1	Unknown	Very High	-1.5

Table 4.11 Allocations of the relevant Gradings/ratings against selected CSCat of the Risk

Finally, risk impact is calculated by multiplying column (a) and column (d) and then all product values of column (e) are added, and further their average is taken as presented in Table 4.14. This average value is noted as the risk impact for the planned sprint / project and is used further in COEEMO for the various calculations.

Number of Risks (a)	Allocations			Risk Impact (a*d)
	CSCat of Risk (b)	Grading / Scale (c)	Ratings (d)	
3	Known mitigated	Moderate	2.5	2.5 * 3 = 7.5
2	Known non-mitigated	High	-0.4	-0.4 * 2 = -0.8
3	Unknown	Very High	-1.5	-1.5 * 1 = -1.5
Total StkCoord= Sum of CSCat StkCoord/Total number of Stakeholders				5.2/6 = 0.86
Final Value of the Risk impact for the planned sprint / project				0.86

Table 4.12 Final value calculations of the Risk impact

4.6.1.5 How to calculate complexities/impacts of the factors excluding expertise/experience, stakeholder coordination, and risk?

No calculations are required for the allocation of values/ratings of all other COEEMO factors: Team Relationship & Spirit, System complexity, impacts of the Organization type, culture, work environment, and the sprint. The ratings of these factors can be chosen directly from Table 4.1 depending upon the selection of the CSCat of that factor and then further selection of the relevant fuzzy scaling and the rating(s) within the chosen CSCat.

For example, if a team considers that their relationship and spirit fall in the “Excellent” CSCat then they need to choose a fuzzy scaling from the five given scales, i.e., from “Very High” to “Very Low” within the Excellent CSCat. If a team decides to select “Very High” fuzzy scaling, then they need to pick the related rating of this scale which in this case is 1.03. A similar process will be followed for all the above said factors to determine their ratings, except the factors that are explained from section 4.6.1.1 to 4.6.1.4.

4.6.1.6 How to calculate the COEEMO Effort Adjustment Factor (CEAF)?

COEEMO includes CEAF computation as the sum of the Effort Factor (EF) and the Learning Factor (LF). Detailed descriptions of CEAF, EF, and LF are presented in section 4.3. CEAF is the final value that the model computes for the estimator. The estimator then calculates the Adjusted Effort

Estimation (AEE) by taking the product of CEAF and the Estimated Effort (EE) for the ASD projects, while the AEE for the non-agile projects can be computed by dividing the EE by the CEAF.

This activity is completed in four steps. Step-1 includes the complete method of technical factor calculation, while the computation of environmental, learning, and effort factors are demonstrated from steps 2 to 4. The data for this activity are drawn from the sample data as shown in Table 4.15. Moreover, the data for team expertise, experience, stakeholder collaboration, and risks are used from the activities as explained in subsections 4.6.1.1 to 4.6.1.4. The other data items are taken from Table 4.1 directly as per the selection of the category/subcategory of the factors for the sample project and are used as sample data for the calculations of CEAF as shown in Table 4.15. Further, to illustrate the computation of sprint complexity, this sample project is assumed to be implemented in four sprints. The ratings along with scales for the complexity of sprints from 1-2 are selected from Table 4.1 while the complexity for sprint 3 and 4 is computed by the formula as provided in subsection 4.4.5.1. The calculations of the complexities for sprint 3 and 4 are demonstrated as follows. The selected complexity scale for the SptNCpxy is “High” while its rating is “0.85”. To calculate the complexity for sprint-3 and 4, the complexity rating of sprint-2 is given to the formula as stated below:

SptCpxy = Spt2Cpxy + ((SptNCpxy – Spt2Cpxy)/ (Expected Sprints / Current Sprint))

SptCpxy for the Sprint-3 = 0.45 + ((0.85-0.45) / (4/3))
= 0.45 + (0.4/1.33)
= 0.45 + 0.3 = **0.75**

SptCpxy for the Sprint-4 = 0.45 +((0.85-0.45) / (4/4))
= 0.45 + (0.4/1)
= 0.45 + 0.4 = **0.85**

The calculated values of the complexities for sprints 3 & 4 are also shown in Table 4.15.

S/No	Factors’ Category	Factors’ Subcategory	Selected	
			Grade / Scale	Ratings
1	Team Relationship & Spirit	Good	High	1.36
2	System Type	Sys-3 (Financial/Banking)	Very High	1.5
3	Organization Type	Joint Sector	Moderate	0.7
4	Organization Culture	Semi-Strong	High	0.75
5	Work Environment	Best work environment	High	0.3
6	Sprint Number (This is 4 sprints project)	Spt1Cpxy	High	0.12
		Spt2Cpxy	Moderate	0.45
		Spt3Cpxy	High	0.75
		Spt4Cpxy		0.85

Table 4.13 Sample Project / Data Used to Exhibit CEAF Calculations

Step-1: Calculation of the Technical Factor (TF) of COEEMO

EF is the product of technical and environmental factors. COEEMO computes the technical factor (TF) by considering selected complexities of the product / system (SysCpxy), and risks. Three types of risks are used to compute risk complexity, i.e., planned risk (PlanR), unplanned risk (UplanR), and unaware risk (UnR). The system complexity from the sample data (as given in Table 4.15) and the risk impact (as shown in subsection 4.6.1.4) are placed in the following base formula (where the details are covered in section 4.4.2) to calculate the TF:

$TF = 0.7 + 0.02 * ((SysCpxy) + (\sum_{i=1}^{j} \frac{PR_i}{j} + \sum_{k=1}^{l} \frac{UR_l}{l} + \sum_{m=1}^{n} \frac{UnR_m}{n}))$ --------- **TF's base formula**

TF = 0.7 + 0.02 * (1.5 + 0.86)

TF = 0.7 + 0.02 *(1.5 + 0.86)

TF = 0.7 + 0.05 = **0.75**

The final value of the technical factor (TF) is in this case 0.75.

Step-2: Calculation of the Environmental Factor (EnvF) of COEEMO

The details of EnvF are given in subsection 4.4.3. The base formula to calculate the environmental factor is given as follows:

EnvF= 1.3 +(-0.05*(Tcap + TPrd + OEI + HI)) -------------- **EnvF's base formula**

Where:

HI: is the sum of the team expertise (TExpert), team experience (TExp), stakeholder coordination (StkCoord), and the team relationship (TR) which are calculated by the following formulas (EnvF's sub-base formula-1):

HI = TExpert + TExp + StkCoord + TR ----------------- **EnvF's sub-base formula-1**

TExpert: Sum of the ratings of each team member's expertise / Total no. of team members
(As already shown in activity-1 of the current section, in which Team expertise is calculated for the seven team members and the final value is 1.21)

TExp: Sum of the ratings of each team member's experience / Total no. of team members
(As already presented in activity-2 of the current section, in which Team experience is calculated for seven team members and the final value is 1.33)

StkCoord: Sum of the ratings of each stakeholder coordination / Total no. of stakeholders

(As already exhibited in activity-3 of the current section, in which Stakeholder coordination is calculated for the three stakeholders and the final value is 0.83)

TR: The selected ratings by the team from Table 4.15.
TCap: is computed by the given formula (EnvF's sub-base formula-2):

TCap = (0.3 * TExpert + 0.5 * TExp) * TR --------------- **EnvF's sub-base formula-2**

OEI: Calculated by taking the sum of the selected wt. values of the organization type, culture, and environment.

OEI: Sum of Ratings of (Organization Type + Culture + Work environment)

----- **EnvF's sub-base formula-3**

TPrd: Computed by the following formula (EnvF's sub-base formula-4):

TPrd =0.25 * (TCap + OEI) ----------------- **EnvF's sub-base formula-4**

Now, by putting the values from Table 4.15 and activities from 1-4 in EnvF's sub-base formulas 1-4, the values of Human Index (HI), Team Capability (TCap), Organization Efficiency index (OEI), and Team Productivity (TPrd) can be calculated as shown:

HI = TExpert + TExp + StkCoord + TR
= 1.21 + 1.33 + 0.83 + 1.36
HI = **4.73**

TCap = (0.3 * TExpert + 0.5 * TExp) * TR
= (0.3 * 1.21 + 0.5 * 1.33) * 1.36
= (0.36 + 0.67) * 1.36
TCap = **1.40**

OEI = Sum of Ratings of (Organization Type + Culture + Work environment)
OEI = 0.7 + 0.75 + 0.3
OEI = **1.75**

TPrd = 0.25*(TCap + OEI)
= 0.25 * (1.4 + 1.75)
TPrd = **0.79**

Further, by putting the above computed values of HI, TCap, OEI, and TPrd in the base formula of EnvF, the final value of the Environmental Factor (EnvF) can be calculated as:

EnvF = 1.3 +(-0.05* (Tcap + TPrd + OEI + HI))

= 1.3 + (-0.05 * (1.40 + 0.79 + 1.75 + 4.73))

= 1.3 + (- 0.05 * 8.67)

= 1.3 – 0.44

EnvF = **0.87**

Step-3: Calculation of the Effort Factor (EF) of COEEMO

EF is the product of technical factor (TF) and environmental factor (EnvF); therefore, the current EF is 0.65 (by taking the product of both factors' computed values, i.e., 0.75 and 0.87, as calculated in the preceding steps).

Step-4: Calculation of the Learning Factor (LF) of COEEMO

LFs for agile and non-agile projects are computed by the formulas as indicated below:

$LF\ = (1 - EF) * (SptCpxy) * N^{b}$ ----------------------- LF base formula for agile projects

$LF\ = (1 - EF) * M^{b}$ --------------------------------------- LF base formula for non-agile projects

Where:

- EF is effort factor
- b is the slope of LC and is calculated by the formula: Log (Learning Rate) / Log_2
- N is the value calculated by dividing the total number of expected sprints by the current sprint in agile
- M is the value calculated by dividing the expected duration of the project in Person-hours by the work hours in a day (i.e., normally 8 hours are contemplated as work hours by most organizations, but this can be modified as per the needs of a specific organization) for TSD projects developed without sprints
- SptCpxy: This is the rating as selected by the team keeping in view the complexity of a sprint. The SptCpxy's ratings are given in Table 4.1 for the first two sprints under the sprint category but if the project / product has more than two expected sprints then the formula as stated below will be employed to calculate the complexity of the third and later sprints:

SptCpxy = Spt2Cpxy + ((SptNCpxy- Spt2Cpxy) / (Expected Sprints/ Current Sprint))

Notes:

- Current sprint means the upcoming sprint that the team intends to estimate (the expected and current sprints must be given as integer values, e.g., in 10/4, 10 represents the total number of the expected sprints while 4 is exhibiting the current sprint number for which the team wants estimation).

- Spt2Cpxy and SptNCpxy are the ratings as given in Table 4.1 under the five scales (i.e., Very High, High, Moderate, Low, and Very Low). These ratings can be chosen by the team based on the complexity of current sprint. If the team selects SptNCpxy rating from the "Very High" scale then they should choose the corresponding value of the rating for the Spt2Cpxy from the same scale, if they do not remember the actual complexity rating for sprint 2.
- The value for the learning rate (LR) can be considered from 80% to 90% depending upon the learning capability (i.e., fast or slow) of the team. However, the recommended LR for software development processes is 80% (Racoon, 1995). The typical LRs for most of the industrial processes are exhibited in Table 2.8.

The sample project as shown above is an agile project which is supposed to be implemented in four sprints. Therefore, LFs for four sprints are calculated by using the aforesaid formula for agile projects. The complexities of sprints 1-4 as taken from the sample Table 4.15 (i.e., 0.12, 0.45, 0.75, and 0.85) are placed in the LF formula to calculate the LFs for the four sprints. The sprint-wise computed LF along with input values are presented in Table 4.16.

SPT#	SPT Cpxy (a)	EF (b)	1-EF (c)	(TSPT/CSPT) (d)	LR (e)	Log (LR) / Log_2 (f)	LF = c * a * d ^ f
1	0.12	0.65	0.35	4	0.8	-0.32	0.03
2	0.45	0.65		2			0.13
3	0.75	0.65		1.34			0.24
4	0.85	0.65		1			0.30

SPT#: Sprint No. EF: Effort Factor TSPT: Total Expected Sprints CSPT: Current Sprint
LR: Learning Rate LF: Learning Factor Log_2: Logarithm base 2 SPT Cpxy: Sprint Complexity

Table 4.14 Sprint-wise Computed LF for Sample ASD project / Data

So finally, the value of CEAF can be calculated after the completion of step-4. To illustrate the CEAF computation, one more column is added in Table 4.16 while CEAFs for all the sprints are shown in Table 4.17. These are the final values which the estimator / team can use to adjust their effort estimates, by taking the product of CEAF for the relevant sprint and the team-provided estimates, e.g., if the team's sprint-wise total estimated story-points (TESPs) are 35, 45, 50, and 60 for sprint-1, 2, 3, and 4 respectively, then their adjusted effort estimates (AEE) or adjusted story-points (ASPs) after multiplying them with the CEAFs of the relevant sprints (as shown in Table 4.17) are 24, 35, 46, and 57.

SPT#	SPT Cpxy (a)	EF (b)	1-EF (c)	(TSPT/CSPT) (d)	LR (e)	Log (LR) / Log_2 (f)	LF = c * a * d ^ f	CEAF= EF + LF
1	0.12	0.65	0.35	4	0.8	-0.32	0.03	0.68
2	0.45	0.65		2			0.13	0.78
3	0.75	0.65		1.34			0.24	0.89
4	0.85	0.65		1			0.30	0.95
SPT#: Sprint No. EF: Effort Factor TSPT: Total Expected Sprints CSPT: Current Sprint SPT Cpxy: Sprint Complexity LR: Learning Rate LF: Learning Factor Log2: Logarithm base 2 CEAF: COEEMO Effort Adjustment Factor								

Table 4.15 Sprint-wise Computed CEAF for the Sample ASD project / Data

These ASPs are the final story-point values that the team will seek to implement in their upcoming sprints. COEEMO's usage in other scenarios, e.g., if the team does not provide TESPs or estimates or if the estimates are presented by using T-shirt sizing and not story points, and COEEMO utilization in a non-agile environment, are covered in the subsequent sections.

4.6.2 COEEMO's Usage in Various Scenarios

4.6.2.1 How to Use COEEMO if the team doesn't provide ESPs or estimates for all or a single sprint(s)/iteration(s) for ASD Projects?

COEEMO can work in situations where the team is experienced or new or in a transitional phase to an agile methodology, or in scenarios where the team desires that COEEMO should provide estimates for a single and/or multiple upcoming sprint(s) of software development project(s). To apply COEEMO effectively in these specific situations, the team is required to follow two core steps. In step-1, the team needs to provide the actual team capacity (ActTeamCap) based on the template given as Appendix-B (The detailed instructions are provided in the attached template to calculate the actual team capacity in points). After getting the actual team capacity in points, the given formula is used to calculate the projected team capacity (PrjTeamCap) for the upcoming sprint:

PrjTeamCap = ActTeamCap + DTLF ---------------- Projected Team Capacity base formula

Where:

- DTLF is the difference of Team LF (TLF) and the actual capacity of the team (ActTeamCap) while TLF can be calculated by the given formula: TLF= ActTeamCap $* (SPT + 1)^b$
- b is the slope of LC and is calculated by the formula: Log (Learning Rate) / Log_2
- SPT is the current sprint number for which the team intends to get estimations

The team / product owner needs to list down the stories of the project/product based on the referenced sizes as presented in Table 4.18. The given sizes of the stories can be associated by the team based on their own benchmarked stories' sizes, e.g., if the team benchmarked story has the size 0.5 or 1.5, then the initial size of the XSS can be reset to 0.5 or 1.5 and all other subsequent sizes of the stories can be readjusted accordingly. Further, if the team suggests the size of any story is bigger than 13 then the breakdown of that story is recommended into any size(s) as are demonstrated in Table 4.18.

S#	Size Description of the Story	Size of the Story (Story-Points)
1	Extra-small size story (XSS)	1
2	Small size story (SS)	3
3	Medium size story (MS)	5
4	Large size story (LS)	8
5	Extra-large size story (XLS)	13

Table 4.16 Recommend sizes of the stories

To plan and estimate the upcoming sprint, the team is required to pick the number of stories from their product backlog based on their own/product priorities or business value but the sum of the story-points for their chosen number of stories must not be more than the projected team capacity (PrjTeamCap). For step-2, the team is required to follow the activities reported in section 4.6.1 to calculate CEAF, EF, TF, and EnvF.

To illustrate the above scenario, a sample product backlog as presented in Table 4.19 is utilized. There are nine stories to be implemented in this envisioned project/product. These stories are categorized based on their priority. Further, by using the specifications as given in Table 4.18, the team / product owner has classified the stories based on their sizes.

ToDo List				
IDs	**Story's Description**	**Priority**	**Story-Size Description**	**Story-Size Points**
ST-01	As a visitor, I want to register myself into the site by providing my name and email address so that I can participate in the discussions' forums (DFs) of the site.	1	XS	1
ST-02	As a member, I want to read the different discussions' blogs on the home page of the site so that I can contribute in the discussion blogs of my interest.	2	MS	5
ST-04	As an administrator, I want to remove the unethical discussions blogs to establish a pleasant discussion environment among the site members.	3	LS	8
ST-03	As a member, I want to subscribe for an RSS feed of news to keep myself updated.	4	MS	5
ST-07	As an administrator, I want to remove the members from the list of the site members on their request so that they can feel liberty	5	LS	8
ST-05	As a member, I want to read Frequently Asked Questions related to the ethics policies for the participation in the discussions' blogs so that I can keep myself within the defined policies of the ethics.	6	LS	8
ST-06	As a visitor, I want to read the diverse DFs so that I can join my interesting forum.	7	SS	3
ST-08	As a member, I want to share my DFs with non-members (my friends or family) of this site so that they can also get benefits of the DFs.	8	MS	5
ST-09	As a member, I want to view the similar blogs of the discussions on the other sites so that I can improve my knowledge base.	9	LS	8

Table 4.17 Sample Product Backlog

To decide how many stories and total story-points the team can undertake in sprint-1, for example, if the total expected sprints are 3, what is the procedure? In step-1, the team is required to provide the actual team capacity (i.e., 15 points in this case) to the above stated formula to compute the projected team capacity of the team for sprint-1. If the actual capacity of the team remains the same for all three expected sprints while sprints' complexities are "Moderate" for sprint-1, 2 and 3 respectively, then the projected team capacity (PrjTeamCap) values for all the three sprints are shown in Table 4.20. As the table shows, the sprint-wise projected team capacities are 18, 20, and 21 points for sprint-1, 2, and 3, respectively, which means that the team can undertake the number of stories in each sprint if their sum of story-points are equal or close to the projected team capacities (where the acceptable difference is within 1 to 2 story-points). Consequently, in sprint-1, the team can consider only four stories (i.e., story IDs ST-01, ST-02, ST-04 and ST-03) to implement because the sum of their Story-points (19) is slightly above the projected team capacity. Similarly, three (story IDs ST-07, ST-05, ST-06) and two (story IDs ST-08, ST-09) stories can be implemented in sprint-2 and 3. However, the final decision regarding the implementation of the actual number of stories per sprint can be taken by the team after the completion of step-2 where the team is required to calculate CEAF, EF, TF, and EnvF, as shown in subsection 4.6.1.

Sprint#	ActTeamCap (Points)	b Log (LR) / Log2	SPT	PrjTeamCap (Points)
1	15	-0.32	1	18
2	15	-0.32	2	19
3	15	-0.32	3	20
LR: Learning Rate i.e. 80% SPT: Current sprint number for which the team is interested to estimate				

Table 4.18 Sprint-wise Projected Team Capacity

4.6.2.2 How to Use COEEMO for TSD Projects where development is done without sprints / cycles?

COEEMO can also be utilized for non-ASD projects where the product / project development is followed without sprints or cycles. To illustrate its usage, sample data for three non-ASD projects are reported in Table 4.21 (a) while their data regarding the COEEMO factors are presented in Table 4.21(b).

S#	Project Name	Estimated Effort (Man-Hours)	Team Members
1	Project-A	2,112	4
2	Project-B	3,168	6
3	Project-C	4,224	8

Table 4.19(a) Sample Data for Non-ASD Projects

These values are employed to compute the CEAF, EF, EnvF, and LF along with some intermediate factors' values (i.e., Human Impact, Team Capability, Team Productivity, and the organization Efficiency Index) for the aforementioned non-ASD projects by following the methods as demonstrated in subsection 4.6.1.

S#	COEEMO Factors	Computed / Selected Ratings		
		Project-A	Project-B	Project-C
1	Team Expertise	0.3	0.82	1.48
2	Team Experience	0.85	0.35	1.7
3	Team Relationship & Spirit	1.18	1.03	1.33
4	Stakeholder / User Collaboration	0.7	1	0.25
5	System Type Complexity	0.1	1.65	2.35
6	Organization Type	1	0.7	0.4
7	Organization Culture	0.75	0.95	0.35
8	Work Environment	0.3	0.65	1.05
9	Risks			
	Known Mitigated Risks	2.1	2.7	2.9
	Known non-mitigated Risks	-0.5	-0.3	-0.1
	Unaware Risks	-1.5	-1.1	-0.7

Table 4.20 (b) Sample Data of the COEEMO Factors for Non-ASD Projects (Project-A, B, and C)

All the computed COEEMO factors' values and the adjusted estimated Effort (AEE) for the three projects are presented in Table 4.23(a) and Table 4.23(b) respectively.

Project Name	HI	Tcap	OEI	Tprd	TF	EF	LF	CEAF
Project-A	3.03	0.61	2.05	0.66	0.70	0.98	0	0.98
Project-B	3.20	0.43	2.30	0.68	0.76	0.97	0	0.97
Project-C	4.76	1.72	1.80	0.88	0.79	0.84	0.02	0.86

Table 4.23 (a) COEEMO Factors Values Computations for Non-ASD Projects (Project-A, B, and C)

AEE is computed by dividing the Estimated Effort of the three projects by the CEAF values as shown in Table 4.23(b).

S#	Project Name	Estimated Effort (Man-Hours)	Team Members	AEE (Man-Hours)
1	Project-A	2,112	4	2,155
2	Project-B	3,168	6	3,266
3	Project-C	4,224	8	4,912
AEE: Adjusted Estimated Effort				

Table 4.21 (b) Adjusted Estimated Effort for Non-ASD Projects (Project-A, B, and C)

4.7 COEEMO's Summary

4.7.1 Model's INPUTs

COEEMO considers five core contexts (Human, Organization, Sprint, System, and Risk) to compute their impact on the project/sprint.

4.7.2 Model's Metrics

For the purpose of the effort estimations, model provides numerous intermediate metrics which are ultimately used to calculate the COEEMO Effort Adjustment Factor (CEAF). These metrics are stated as follows:

a. Team Capability (Tcap): Details are given in sections 4.4.3, 4.6.1.1 & 4.6.1.6 (Step-2)
b. Team Productivity (Tprd): Details are provided in section 4.4.3, 4.6.1.2 & 4.6.1.6 (Step-1 to 2)
c. Organization Efficiency Index (OEI): Details are presented in sections 4.4.3 & 4.6.1.6 (Step-2)
d. Human Impact (HI): Details are exhibited in sections 4.4.3 & 4.6.1.6 (Step-2)

4.7.3 Model's Assessments

The model provides, the final assessments:

a. Technical Factor (TF): Details are given in sections 4.4.2 & 4.6.1.6 (Step-1)
b. Environmental Factors (EnvF): Details are exhibited in sections 4.4.3 & 4.6.1.6 (Step-2)
c. Effort Factor (EF): Details are provided in sections 4.4.4 & 4.6.1.6 (Step-3)
d. Learning Factor (LF): Details are demonstrated in sections 4.4.5 & 4.6.1.6 (Step-4)
e. COEEMO Effort Adjustment Factor (CEAF): Details are presented in sections 4.4.1 & 4.6.1.6

4.8 Chapter Review

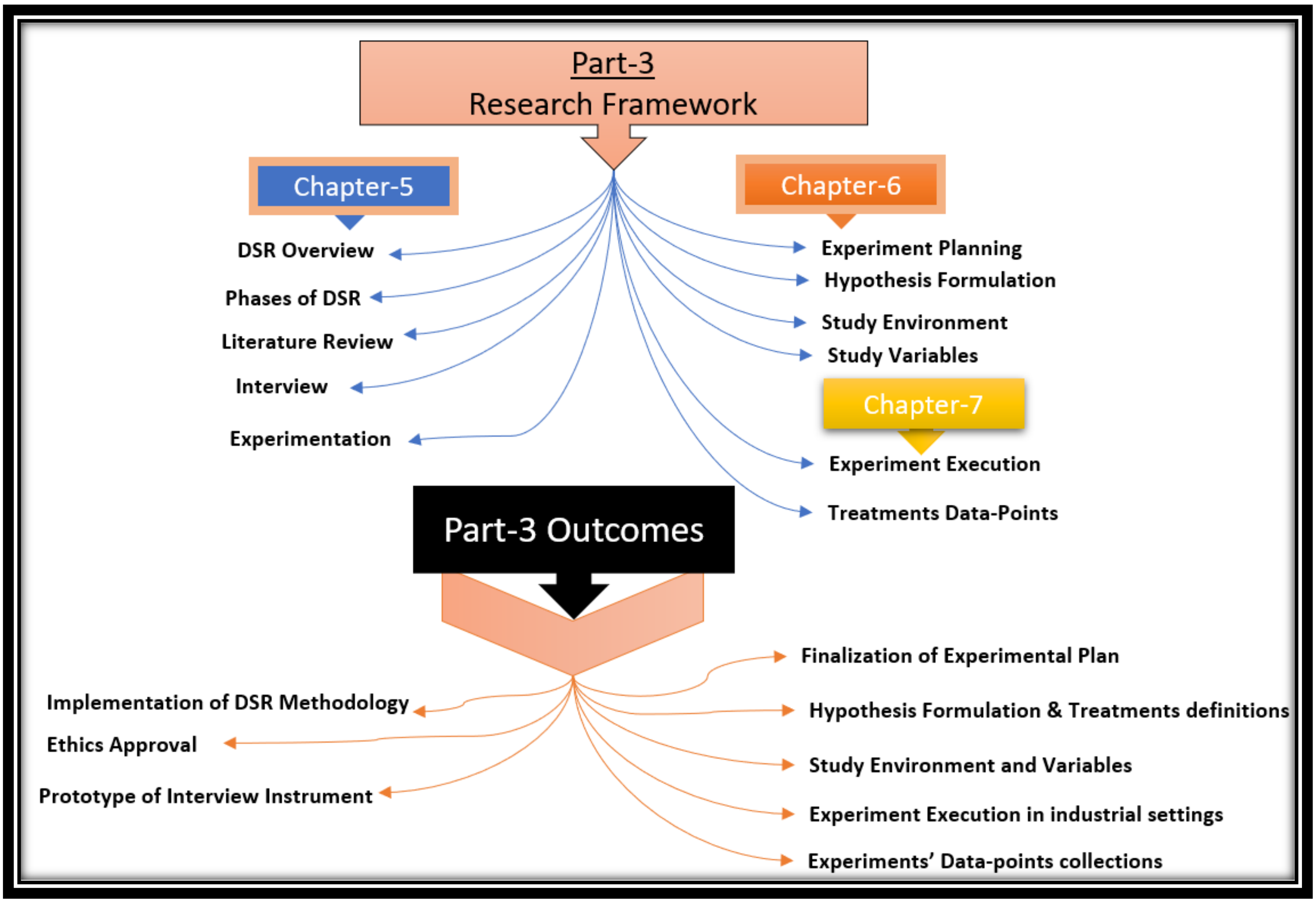
In this chapter conceptual and functional views of the COEEMO (Constructive Effort Estimation Model) were discussed. Moreover, the model's various components along with their detailed descriptions and a number of scenarios for implementation were presented.

In next chapter, the significance of the Design Science Research (DSR) methodology and its projected relevance to the intended study is described. Further, DSR's suggested three phases and their alignment with this study are presented. The PhD study plan involving the use of a variety of methods in each DSR phase, along with their justification and importance, are discussed. At the end of this chapter, data collection strategies and methods along with the ethics approval sought from the AUT ethics committee (AUTEC) to conduct this intended study are described.

PART-3 RESEARCH FRAMEWORK

This part consists of three chapters which describes the detailed research methodologies that are adopted in this study along with the ethics approval from AUTEC. This part also discusses detailed plans of the experimentation and its execution along with collection of experimental data.

Detailed views of the research activities with respect to the chapters within this part are presented in the figure as given below. On the top of the diagram, the chapters covered in this part are shown along with their different research endeavours as accomplished within the individual chapters, while the outcomes of Part-3 based on its individual chapters are exhibited at the bottom of the figure.



Chapter-5 Research Methodology and Ethics Approval

***Content Summary**: This chapter describes the research methodology, the relevance of its selection, the mapping of the methodology with the research questions, research goals, and PhD study plan. The last section of this chapter discusses the use and importance of mixed-methods research and the methods of data collection employed, along with ethics approval from the AUT ethics committee (AUTEC) for the proposed study.*

As already described briefly in Chapter-1, the Design Science Research (DSR) methodology as per the guidance provided by Offermann et al. (2009) is employed to conduct this intended study in order to answer the four research questions as stated in that same chapter. DSR has undergone continuous refinement over more than 40 years (Cross, 2007) and has gained positive recognition in the discipline of Information Systems (Nunamaker et al., 2013). Numerous researchers have not only informed the multiple frameworks of DSR but also shared their benefits while employing DSR as a methodology in conducting and reporting research in the IS domain (Gregor and Hevner, 2013; Nunamaker and Briggs, 2011, Peffers et al.,2007; Hevner et al., 2004; Nunamaker et al., 1991). The DSR process as proposed by Offermann et al. (2009) is shown in Figure 1.1 of Chapter 1. This process suggests three core phases, i.e., Problem identification, Solution design & development and Evaluation, further divided into various steps as demonstrated in the above stated figure.

5.1 Design Science Research

Design Science Research (DSR) is a methodology that seeks to generate new knowledge through design and recommends ways through which research and design can be practically joined (Cross, 2007). This methodology has been employed by IS researchers for more than two decades for the purposes of exploration and clarification for their in-hand phenomena of interest (Peffers et al., 2006). In early 1990s DSR was suggested to be used for Information System and/or its allied domains (March & Smith, 1995; Walls et al., 1992; Nunamaker et al., 1990). Since then, numerous studies (Vaishnavi & Kuechler, 2015; Gregor & Hevner, 2013; Hevner & Chatterjee, 2010; Hevner, 2007; Alan et al., 2004) have encouraged IS or similar fields' researchers to apply DSR in conducting artefact-related research, in order to produce generally useful and rigorous research outcomes (Rosemann & Vessey, 2008; Alan et al., 2004; Benbasat & Zmud, 1999) in the domains of interest.

The reasons why DSR has gained recognition in the IS discipline, which were relevant to it being used in this intended study, include: it has suggested clear directions along with a roadmap for research in Information System through the lens of design science (Offermann et al., 2009); it motivates the construction of artefacts for resolving real-life issues or problems (Alan et al., 2004); it provides opportunities to conduct research of an applied nature which not only contributes to producing general insights about the problem in-hand but also contributes to practice (Chamberlain et al., 2007; Smaczny, 2001); it proposes comprehensive directions about artefacts, their roles and the methods to produce them as research outcomes which are not only rigorous but also relevant (Rosemann & Vessey, 2008; Alan et al., 2004; Benbasat & Zmud, 1999).

This study followed the DSR methodology as per the guidelines provided by Offermann et al. (2009). This comprised three phases, i.e., problem identification, solution design & development, and evaluation. Further, these three phases were interconnected, meaning that after the completion of each phase the previous phases were revisited for the purpose of improvements, modifications, revalidation and re-evaluation. Each phase was considered with respect to the current study as follows.

5.1.1 Problem identification Phase

Problem identification is the first phase of DSR, in which the area of research is chosen based on the interests of the researcher and the significance of the practical and knowledge requirements of the selected research area. The literature review is an important technique to get an overview of current knowledge related to the area of research prior to initiating further investigation. DSR advocates this type of activity in the problem identification phase. Such an endeavour in research is referred to as a secondary study because it does not have the requirements for collection or analysis of primary data.

In Software Engineering a more systematic approach has been taken to such almost for more than a decade (Kitchenham, 2004a: Kitchenham et al., 2004). Further, two methods are being followed in SE to execute secondary studies: the Systematic Literature Review (SLR) and the Systematic Mapping (SM). Kitchenham et al. (2007) note that through such approaches research gaps and opportunities may be identified for further improvement.

The area of effort estimation in agile development is the identified domain in which this research was being conducted. This phase had two steps to determine the current state of effort estimation in agile development contexts. In step-1, the underlying problems and issues in this area were surfaced by literature reviews in addition to three SLR studies in the field published between Sep 2014 and Dec 2015 (Ahimbisibwe et al., 2015; Usman et al., 2015; Usman et al., 2014). To search the relevant literature, both manual and automatic searches were performed on the basis of relevant keywords (e.g., Effort estimation, agile based estimation, Gaps/Limitations in Agile methods, Critical factors for effort estimations, Significant factors for software development projects, Cost drivers for software projects, Estimation techniques/Methods, Traditional and non-traditional methods of effort estimations, Weaknesses of traditional methods, COCOM, UCP, FPA). Snowballing techniques were also utilized to search for relevant papers from the aforementioned SLR studies. Initially, 550 relevant papers were gathered from four digital libraries, i.e., IEEE Xplore, ACM, SCOPUS, and Google. Later, after applying the exclusion criteria, 350 relevant papers were finalized to include in this study.

In the second step, practitioners' interviews were conducted to support identification and validation of the related problems in that area of research. Interviews are a primary method to be used in qualitative research (Doody & Noonan, 2013; Schultze and Avital, 2011). Further, they are said to be one of the best means in qualitative research to determine others' perceptions (Cisneros-Puebla, Faux and Mey, 2004; Patton and Michael, 2002).

In this study face-to-face interviews were planned through physical contact or over phone / skype / whatsapp. In the last three decades interviews over phone as well as other modes, e.g., skype, VoIP (voice over internet protocol) and others, have gained popularity, given evidence that they produce similar outcomes as face-to-face interviewing (Miller, 1995; Holt, 2010; Sturges and Hanrahan, 2004; Opdenakker, 2006; and Vogl, 2013). The Daily Interpretive Analysis (DIA) technique was utilized to assemble and interpret the information as collected from the interviewees at the end of every day of interviewing. Further, audio recording and note taking were also utilized for the interviews. Suitably qualified and experienced participants were recruited and also provided with pre-interview guidelines one week prior to the planned meeting/interview.

The outcomes of the problem identification phase of DSR are the input and motivations to initiate the next phases, with the intent to deliver new knowledge / artefacts being useful in resolving the problems/issues and for advancement in research and practice.

5.1.2 Solution Design and Development Phase

During the execution of this phase a solution needs to be designed and employed while keeping in view the requirements of the academia and practice (Offermann et al., 2009). For artefact design there is no single strategy provided in the IS literature, but this can be thought as an innovative engineering process (Offermann et al., 2009).

Some studies have published general approaches, i.e., prevailing solution and state-of-the-art is deliberated (Eder et al., 2007; Simon, 1996). Walls et al. (1992) suggested that an Information Systems Design Theory (ISDT) should be "a prescriptive theory which integrates normative and descriptive theories into design paths intended to produce more effective information systems." March and Smith (1995) proposed artefacts segregation into four groups, i.e., models, methods, constructs, and instantiations.

To undertake the solution design and development phase for this study, the framework as suggested by Venable (2006) was followed. This framework states that design theory should follow a format of utility theories ─ this relates to probable enhancements gained by employing a specific single or multiple types of technology to address a particular issue or a problem. An activity framework as shown in Figure 5.1 was adapted from Venable (2006) for this study. This figure shows a framework comprising a novel model of estimation along with two existing methods of estimation, i.e., PERT and Expert Judgement based on a Bottom-Up approach.

During this phase a novel model for estimation was designed and developed based on theories, constructs, assumptions, and factors as indicated in the problem identification phase. The conceptual functional framework of this model was constructed along with the running prototype of this model. Two techniques, i.e., PERT and Expert Judgement based on a Bottom-Up approach, were suggested to be utilized to resolve the underlying issues and problems as surfaced in the problem identification phase. The current PERT technique was modified (reported in Chapter 6) and its two variants were devised to analyse their effectiveness in effort estimation. There are therefore three artefacts developed during this phase that were inducted into the next phase, i.e., Evaluation, to analyse their efficacies as compared to existing methods of effort estimation, to resolve the identified gaps and limitations as conceived from the problem identification phase to improve ASD estimation accuracy.

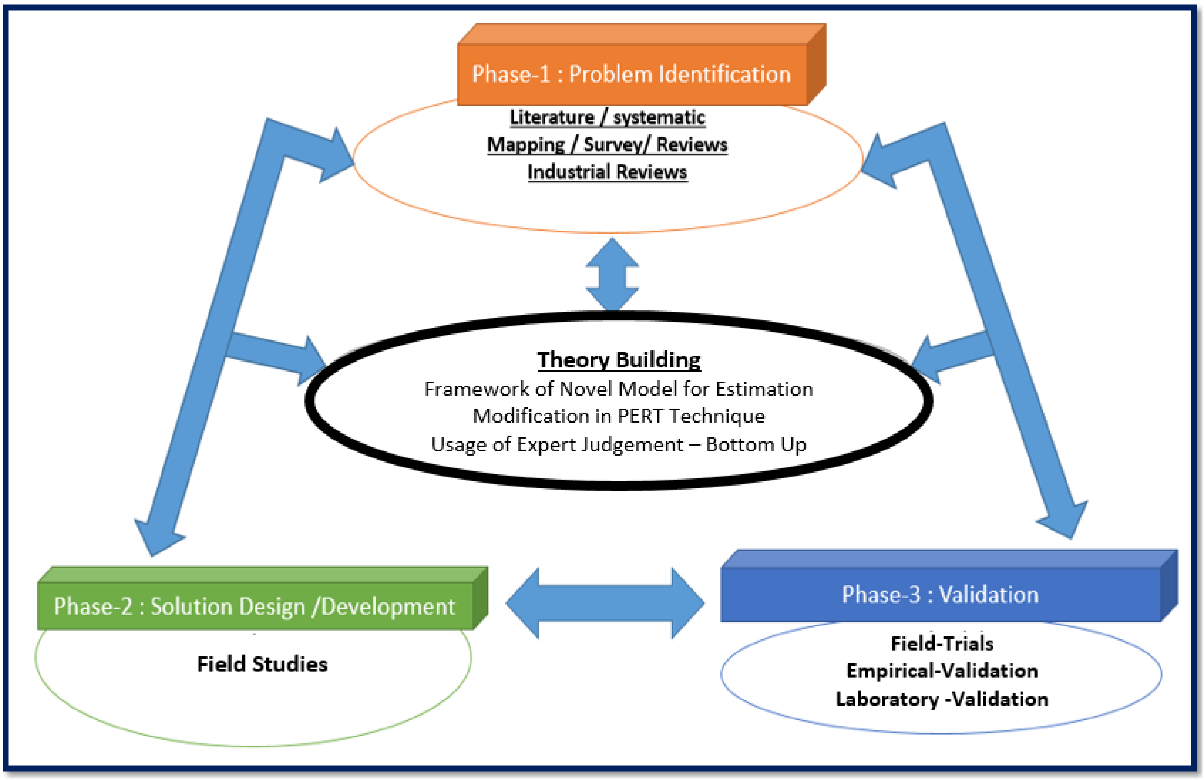


Figure 5.1 DSR Implementation for the Intended Study

5.1.3 Evaluation Phase

This phase is important to corroborate the expectations of the artefacts as developed in the solution design phase in terms of actually being a solution for the problems and issues identified (Offermann et al., 2009). Further, there is also a possibility to revisit the design of the artefacts (being completed in the solution phase) and/or the problem identification phase, depending upon study requirements (Vaishnavi and Kuechler, 2004). The validation activity may be accomplished by using one or many techniques that include but are not limited to field's trials, laboratory or field experiments, case studies and experts' surveys.

In the evaluation phase of this study, in order to validate artefacts three distinct perspectives were employed, i.e., field, empirical and laboratory. In the first, artefacts were applied in field settings on real industry data and their results were compared with those obtained from existing methods. The second perspective, i.e., empirical, was accomplished by applying artefacts on the data of already completed ASD projects collected from different sources: published studies and the IT industry. For the context of laboratory validation, the artefacts were implemented using a well-established fuzzy inference model (i.e., the Mamdani model in the MATLAB toolbox) and applied to the already completed projects, and their results were compared.

In all three validation perspectives, two metrics were used to measure the accuracy rate (BRE) and the direction of any error (BREbias) for each effort estimation. A model/method resulting in a lesser value of BRE compared to another such models was considered more accurate. The core objective of evaluation to establish that artefacts offer a practicable solution for the in-hand problem/issue.

At the end of research, the findings and conclusions are reported through various forms, typically conference or journal articles or a thesis (Offermann et al., 2009). In this research, the whole research endeavour along with its outcomes are published in the form of this thesis. However, some specific features of this research will also be published separately: the framework of the proposed novel model for effort estimation, an industrial review publication reporting the outcomes of the practitioners' interviews, and the findings/outcomes detailing the field, laboratory and empirical studies.

Figure 5.2 presents the PhD research plan employed during this intended study. In this figure, all phases of the intended research are shown along with the study year. Moreover, every phase output is shown by the cloud callout symbols. In the first year of study, initial searches of the literature along with the prevailing methods of estimation were conducted. The cloud callout of the first year shows the various outputs: a list of issues/gaps in effort estimation, critical factors impacting estimations, and others. Likewise, every phase of DSR aligned with this intended study year wise. In year-3, the envisioned artefacts were validated and then finalized.

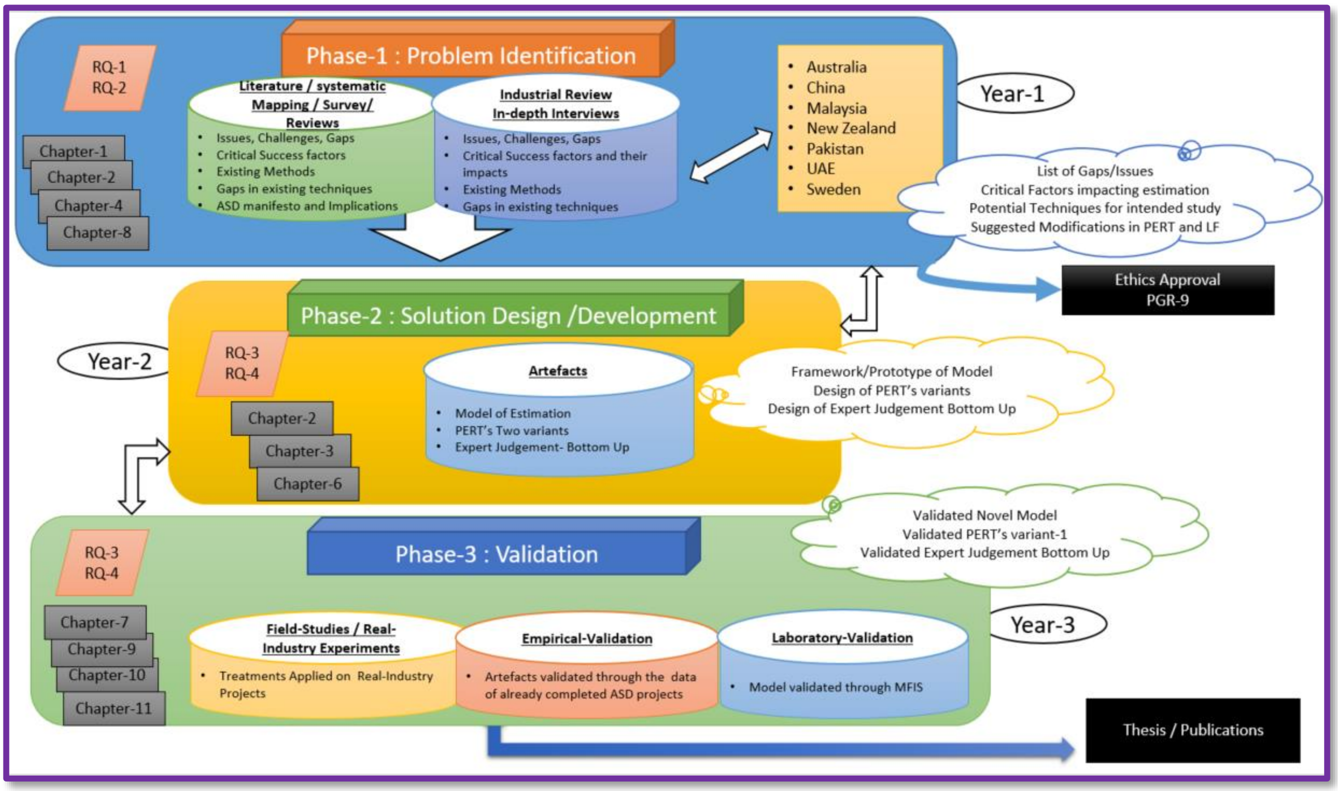


Figure 5.2 Mapping of PhD study plan with the intended study

Figure 5.2 shows three perspectives were planned to be inducted to validate the artefacts that were developed under the solution design and development phase.

5.2 Mixed Research Methods for the Intended Study

During this study various types of methods as explained in the subsequent paragraphs are used. A mixed-methods approach to research is very effective for in-depth comprehension of a theory or phenomenon of interest (Venkatesh et al., 2013) while offering opportunities for a variety of interpretations (Teddlie and Tashakkori, 2009). For example, in the problem identification phase, both qualitative and quantitative methods were used while analysing the industrial reviews regarding the research problem. Numerous studies have advocated the usage of a variety of methods in information systems research (Lee 1999; Robey 1996; Sidorova et al. 2008). However, their usage has been limited: only 5 percent of the empirical studies published between 2001 and 2007 in the six major journals of Information System had engaged mixed methods (Venkatesh et al., 2013). Creswell and Clark (2007) proposed mixed methods' designs of four key types that include: triangulation — this unifies information of a qualitative and quantitative nature to comprehend a problem/issue of research; embedded — responding to a research question by utilizing information either quantitative or qualitative in nature; explanatory — employs data qualitatively to illustrate quantitative outcomes of a research problem/issue; and exploratory — enlightening and evaluating an association established in qualitative information through quantitative data.

5.2.1 Individual in-depth interviews

Interviews were conducted individually with industry practitioners to collect data relating to their perceptions of the factors that were thought to be relevant to estimation. Audio-recording along with note taking were used to collect interview data. Daily Interpretive Analysis (DIA) was utilized to assemble and interpret data as collected from the interviewees at the end of every day of interviewing. Suitably qualified and experienced participants were recruited, and they were given pre-interview guidelines and an interview schedule one week prior to the planned meeting. Further, thematic analysis by following deductive approach was utilized to identify themes from the interview data (Braun &Clarke, 2006).

5.2.1.1 Participants' Selection

Candidate organisations engaged in agile software development were identified from public sources such as the Agile Auckland LinkedIn group and the indexNZ sector listing for software firms, or through the researcher's personal industry contacts. The CIOs of the chosen organisations were contacted by the applicant and primary researcher via email. The message described the nature and purpose of the research, the role of the researcher and our expectations of organisations and participants.

Participants were considered eligible to take part in interviews if they had knowledge and experience working for at least 3 years in an agile context in any of the activities of software development, such as requirements engineering, design, architecture, code development, testing, quality assurance, maintenance, and IT management.

Further, organisations and teams sought to participate in the experimentation phase could be eligible only if they were using an agile methodology in the development of their software products and services. Organisations that responded positively to the request were asked to confirm their participation in the research. Moreover, a Consent form was shared with the participants to secure agreement regarding data and information being used later for the research. Participants had the choice to withdraw at any stage of the interview.

Once confirmation was received the primary researcher proceeded. The primary researcher asked the first contact of the targeted organisations to forward an email invitation to suitable participants who would then be asked to contact the researcher directly. In some cases, the researcher might contact candidate organisations or research participants through phone calls or by face-to-face meetings. In total, 21 participants from 7 different countries were selected for face-to-face interviews. Their details are covered in Chapter 8 of this thesis.

5.2.2 Experimentation

Quasi-experimentation was planned. Projects were used as objects of the experiment while teams who had completed or worked on ASD projects were considered as the subjects of the experiments.

Full details about treatments, hypothesis formulation, and research design for this intended experiment are reported in Chapter 6. Moreover, details of the participating organization, their three industry projects and the associated teams are covered in Chapter 7.

5.2.3 Empirical and Laboratory Validation

Empirical evaluation for the intended study was accomplished by applying anticipated artefacts on the data of already completed ASD projects collected from different sources. For laboratory validation, the proposed model of estimation was implemented by using a well-established fuzzy inference model - MFIS (i.e., the Mamdani model in the MATLAB toolbox) and applied to the already completed projects. In both methods of validation, the outcomes were compared to the existing methods of estimation. The details of the ASD projects that were collected to validate artefacts empirically are provided in Chapter 10. Further, the laboratory validation to compare the results of the proposed estimation model with MFIS are presented in the same above stated chapter.

5.3 Data Collection Methods for the Intended Study

In this envisaged study multiple sources were traversed to gather data in order to limit the effects of constrained interpretation due to the availability of a single data source. Data collected from various software development and management artefacts (both electronic and non-electronic), participant interviews and discussions, as well as quasi-experiments in the fields and validation in laboratory, which in turn were subject to the analysis.

The techniques used for data collection in this study fall under the three broad categories defined by (Lethbridge et al, 2005): a) Direct methods, real time data collected directly from subjects through interviews, meetings and the like; b) indirect methods, which involve collection of raw data without actually interacting with the subjects during the data collection, such as are available via software tools (company's project databank), or observations; and also by c) examining already available compiled or un-compiled data regarding projects (such as documented records of already completed ASD projects, archival data, repositories, logs).

5.4 Ethics Approval

A fundamental principle for all research is to seek ethics approval prior to the start of that research, if human participants are directly or indirectly involved in or impacted by the data collection (Gelling, 2015). For this intended research, human participants were involved at two stages: one at the stage of validation of the identified influential factors, and on a second occasion while conducting experiments in the actual setting of the IT industry. Therefore, the researcher submitted a research proposal to the Auckland University of Technology Ethics Committee (AUTEC) to seek their approval in relation to the conduct of interviews with IT practitioners and for the execution of the planned experiments within the IT industry. In addition, the following documents were also submitted as pre-requisites to AUTEC for their approval:

a. The participant's information sheet (Appendix-I) which described the purpose of the research, its benefits, privacy protection of the participants, and other items of relevance to potential participants. This information sheet was shared with the participants well before their active involvement in the research.
b. Two consent forms seeking agreement to participate in an interview and/or experimentation (Appendix-G & H). These forms were shared with the participants before the conduct of the interview and experimentation sessions. Willing participants were required to place their signatures on the provided space of the forms and email the signed forms back to the researcher before the start of the interview/experiments.
c. An interview guide (Appendix-K), which included a list of some open- and closed-ended questions that were to be put to the participants, and details of the anticipated duration of the interview. This guide was sent to the participants four weeks in advance prior to the actual date of the interview session.
d. The email text that was used to invite participants via email (Appendix-J)

AUTEC approved the above submitted case on 24 April 2018 and allotted reference number 18/153 to the case for future reference. As stated above, relevant documents were shared with the participants well before the time of their engagement in the research. All participants were assured of privacy protection of their projects, data, organizations, and their own identities. Further, they were also advised of their right to withdraw from participations at any time, or from any stage/phase of the research.

5.5 Chapter Review

In this chapter, the significance of the Design Science Research (DSR) methodology and its projected relevance to the intended study was described. Further, DSR's suggested three phases and their alignment with this study were presented. The PhD study plan involving the use of a variety of methods in each DSR phase, along with their justification and importance, were discussed. At the end of this chapter, data collection strategies and methods along with the ethics approval sought from the AUT ethics committee (AUTEC) to conduct this intended study were described.

In the next chapter, the specifics of the experimental planning including the research hypothesis, the various treatments along with details of the study environment, the dependent and independent variables, and hypothesis testing strategies are all covered.

Chapter-6 Experimental Planning

***Content Summary**: This chapter describes the formulation of the research hypothesis and its intended testing based on quasi-experiments. It also encompasses details of all the treatments to be applied to the objects of interest along with the suggested study environments, dependent & independent variables.*

The core empirical work in this thesis was conducted via quasi-experimental methods. The purpose of conducting a series of experiments was to systematically evaluate the impact of a range of forecasting techniques and their variants (Top-Down, Bottom-Up and PERT), as stated in RQ-4. This enabled the comparative analysis of existing methods of estimation along with the novel estimation model proposed here. The model was applied to and calibrated on a number of projects using appropriate regression methods (linear, log-transformed, robust or trimmed) and model performance was assessed using best practice error measures, Balanced Relative Error (BRE) and BREbias, due to their balanced performance in situations of over- and under-estimation (Mahnic et al., 2012; Usman et al., 2018). Moreover, two further measures, Standardized Accuracy (SA) and Effect size (Glass delta – Δ) as recommended by Shepperd and MacDonell (2012), were also utilized to test whether the prediction error of any proposed model or method was equivalent to random guessing, and further, how much improvement over random guessing was attained. Details of the experimental plan are given in the succeeding sections.

6.1 Hypothesis Formulation

To conduct the required experiments a standard Randomized Complete Block Design (RCBD) was proposed. This design had been chosen because it was expected that each subject (Team) would apply all treatments (forecasting methods) on one or more single objects (ASD projects) to test one factor with restrictions on randomization, and RCBD was the most suitable in such a context. It is one of the experimental designs most commonly used in software engineering. The selection of quasi-experiments was made to compare the effects of multiple treatments (Wohlin et al., 2000). To implement RCBD for this experiment, six treatments with one factor were chosen and their details are given as follows:

- **Treatment-1**: Effort estimation through existing methodology for an agile software development project ($Effort_{ExMethod}$): -
  Under this treatment the baseline estimation methods followed within a typical SCRUM framework were Planning Poker and Expert Judgement. Developers would estimate stories based on gut feelings, analogies, previous experiences of similar or other projects and/or a top-down approach but they were not allowed to employ a Bottom-Up method for their estimations. During the estimation process / sprint planning, story estimations were done in story points based on the development complexity of the reference story.

- **Treatment-2**: Effort estimation through existing methodology for an agile software development project extended with PERT (Program Evaluation Review Technique) ($Effort_{ExtPERT}$): -

  Under this treatment the existing methodology was extended with PERT along with its two variants (i.e., Treatments T2a and T2b).

  To apply PERT as a treatment a modification in the existing PERT are suggested. The three points-based estimates of time are replaced by estimates of size in story points and so the new three points-based scale is based on the optimistic size of a story, the pessimistic size of a story and the most likely size of a story. The average size of a story is calculated by the following formula:

  $$E_s = (O_s + 4M_s + P_s) / 6$$

  Where

  a) E_s (Estimated size of the story): is the average size of the story
  b) O_s (Optimistic size of the story): is the minimum value picked from the disclosed values of the team at the time of estimating the size of a particular story before consensus and / or discussion is made among the team
  c) M_s (Most likely size of the story): is the value in-between Os and Ps of a particular story picked from the disclosed values of the team at the time of estimating the size of a particular story before consensus and / or discussion is made among the team. If more than one value is disclosed in-between Os and Ps, then the average of those values is selected.
  d) P_s (Pessimistic size of the story): is the maximum possible value which is picked from the disclosed values of the team at the time of estimating the size of a particular story before consensus and / or discussion is made among the team.

- **Treatment-3**: Effort estimation through existing methodology for an agile software development project extended with expert judgment based on Bottom-Up approach ($Effort_{ExtBU}$): -

  Under this treatment the suggested modifications to the existing methodology were applied. The team was given two options to apply this treatment and a summary of these options is given in the succeeding paragraphs while details can be viewed in Appendix-C. The team had chosen option-1, for two reasons: one, to save time because this treatment was applied on the actual industry project, and two, to reduce the confounding impact of one technique on the other while estimating.

  Option-1:

The team applied bottom-up approach (BUA) on alternate sprints e.g., if they applied existing approach on sprint-1 then BUA was followed in sprint-2, and so on. In BUA, the team had followed work break down structure (WBS) technique in which each member of the team divided development task of the candidate estimating story into various activities/tasks i.e., design, coding, testing, integration, etc. and then shared his / her estimates in the estimation activities.

Option-2:
Single agile team proposed to be divided into two subsets of the small teams e.g., if there were 6 members in a team, they were divided in two subsets (subset-A and B) in equal numbers. In case of 05 members, subset-A or B might have 03 or 02 members each. One subset of team was suggested to apply existing methodology / Top-Down and other BUA simultaneously in a single sprint/iteration. Team applied BUA through WBS as explained in option-1. Further, the final size of the story taken by averaging the two values as decided by the two subsets (subset-A & B) of the team.

- **Treatment-4**: Treatment-3 extended with PERT ($Effort_{ExtBU\&PERT}$)
  Under this treatment, the treatment-3 was extended with PERT's two variants (i.e., Treatments T4a, and T4b).

- **Treatment-5**: Under this treatment, the treatment T1 extended with the proposed Model ($Effort_{ExtProModel}$): -

- **Treatment-6**: Under this treatment, the treatment T3 extended with the proposed Model ($Effort_{ExtProModel}$): -

- Subjects: Team(s) in Software Industry using Agile Methodology

- Objects: Agile Software Development Project(s)

All treatments were applied on the selected projects while estimating the sizes of the stories In the remainder of this section Treatments T1, T2a-b, T3, T4a-b, T5 and T6 will be used in place of $Effort_{ExMethod}$, $Effort_{ExtPERT}$, $Effort_{ExtBU}$, $Effort_{ExtBU\&PERT}$, and $Effort_{ExtProModel}$.

6.1.1 Null Hypothesis (H0)

The null hypothesis is accepted if the same estimation result (i.e., μ_s for all treatments are equal) is returned by all of the treatments:

H_0: $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5 = \mu_6$ where mean is the same for all treatments

6.1.2 Alternative Hypothesis (H1)

On the rejection of H_0 the alternative hypothesis is accepted, which means that the results of the treatments are not the same:

H_1: $\mu_1 \neq \mu_2 \neq \mu_3 \neq \mu_4 \neq \mu_5 \neq \mu_6$ for at least one pair

Appropriate statistical tests were used to test the above hypotheses. They included ANOVA (a parametric test that makes assumptions about the distribution of the population from which data are drawn) and/or Kruskal-Wallis & Chi-squared tests (non-parametric tests that make no such assumptions). The ANOVA (Analysis of Variance) test compares variabilities due to treatment and random error as well as the samples of the same mean values. A summary of the test is presented in the following Table 6.1:

Input	"b" Samples of Treatments from 1-6
H0	$\mu_{b1} = \mu_{b2} = \mu_{b3} = \mu_{b4} = \mu_{b5} = \mu_{b6}$ Where mean is same for all treatments
Calculations	Sum of Square (SS_z) = $\sum_{i=1}^{b} \sum_{j=1}^{n_j} b^2_{ij} - b^2_{...} / N$ *where N is the total number of measurements and a dot Index denotes a sum over the dotted index.* $SS_{Treatment} = \sum_{i=1}^{b} b^2_i/n_i - b^2_{...}/ N$ $SS_{Error} = SS_z - SS_{Treatment}$ Mean Square ($MS_{Treatment}$) = $SS_{Treatment}$ / (b-1) Mean Square (MS_{Error}) = SS_{Error} / (N-b) $F_0 = MS_{Treatment} / MS_{Error}$
Criteria	Reject H_0 if $F_0 > F_\infty$, b-1, N-b. Here, F_∞, f_1, f_2 is the upper ∞ percentage point of the "F" distribution with f_1 and f_2 degrees of freedom

Table 6.1 ANOVA (One Factor with multiple Treatments)

6.2 Study Environments

The experiments were undertaken in sequence in two environments:

a) Environment-A: Industry Environment

In this environment all treatments were applied in the real-world setting of the IT industry; development teams applied all the suggested treatments on ongoing and new ASD projects.

b) Environment-B: Lab (AUT SERL) Environment

To validate and measure the impact of the proposed model on estimations, two activities were undertaken by the researcher in the Lab environment:

i. The proposed model of estimation was applied on 10 projects (i.e., 85 sprints / iterations) developed in agile/SCRUM environments. The data for these projects were collected from prior published studies and from the IT industry.
ii. The proposed model was implemented in a Mamdani Fuzzy model using Matlab toolbox.

As we were relying on the goodwill of volunteers to provide data and access to projects, we used a convenience sampling approach. Personal contacts and professional networks were trawled to seek willing contributors to the research. Teams and projects for which the above data were available, or could be easily collected, were considered for inclusion. While this potentially introduced a bias in the projects, we worked to mitigate this by disclosing as many details as possible and by including the data so that our experiments could be verified.

6.3 Study Variables

6.3.1 Dependent Variables

Effort estimation by the existing (baseline) and proposed methods are the dependent variables of the treatments and hence their responses were evaluated.

6.3.2 Independent Variables

Independent variables are those that differ per sprint, per project, per team and so on. We were interested in assessing the extent to which those differences impact on the dependent variables. We were also interested in the effects of how the inclusion of additional variables influenced the forecasting outcomes.

6.3.2.1 Independent Variables for Treatment-1

Independent variables for Treatment-1 were:

a. Story Size

   Every ASD project comprises a number of stories or epics that are to be delivered through a number of features. The size of each should be evaluated in terms of story points (or similar). The total number of story points is calculated within a sprint, which the team then commits to develop or implement.

b. Number of Sprints

   Every project is developed over some number of sprints or iterations. Each iteration has particular development and release goals which the team strive to achieve before starting the next iteration.

c. Team Velocity / Capacity

   This represents the typical pace of work of a team in a sprint, calculated by considering the numbers of features or story points that are implemented over a collection of iterations.

d. Actual Effort

   Person-hours required to implement the completed / finished story-points in a sprint / project.

e. Estimated Effort

   Person-hours required to implement the estimated story-points in a sprint / project.

6.3.2.2 Independent Variables for Treatment-2 with Variant-1 (PERT-1)

The independent variables of Treatment-1 were augmented by the following in Treatment-2:

a) Es (Estimated size of the story): It is the average size of the story
b) Os (Optimistic size of the story): The value whose frequency is highest in between the values

of the minimum and maximum frequencies is picked from the disclosed values of the team. If there is more than one such value and they have the same frequencies, then the smaller value will be chosen among them. However, if there is no value in between Os and Ps, and overall, only two different values exist, then zero will be considered as the Os value.

c) Ms (Most likely size of the story): The value with the highest frequency is picked from the disclosed values of the team. However, if more than one such value has the same highest frequencies, then the team will follow the steps as explained in the following two scenarios:

   Scenario-1: Consider the situation in which the team discloses their estimated values and out of those, three different values have the same highest frequencies then the team needs to use these values – largest as Ms, large as Os, and small as Ps.

   Sceraio-2: In another situation, if the team has two different values of the same highest frequencies, then the team should choose these two values – largest as Ms, and smallest as Ps.

d) Ps (Pessimistic size of the story): The value with the minimum frequency is picked from the disclosed different values of the team. However, if more than one values are minimum and have the same frequencies then the largest value among them will be picked as pessimistic while the smallest will be assigned to Os, if the Os value is not yet decided.

Note: In case only one value is disclosed as the size of the story by the whole team before discussion or consensus is achieved among the team then the proposed PERT extension will not be applicable.

6.3.2.3 Independent Variables for Treatment-2 with Variant-2 (PERT-2)

The independent variables of Treatment-1 were augmented by the following in Treatment-2:

a) Es (Estimated size of the story): It is the average size of the story
b) Os (Optimistic size of the story): It is the minimum value picked from the disclosed values of the team but if there are only two different values then zero value will be considered as optimistic value.
c) Ms (Most likely size of the story): It is the highest value picked from the disclosed values of the team in between Os and Ps. However, if there is no value in between Os and Ps, except the overall two different values then the smallest one will be considered as Ms.
d) Ps (Pessimistic size of the story): It is the maximum possible value which is picked from the disclosed values of the team.

Note: The following suggestions are applicable when the given situation or scenario emerges while applying PERT's variants, but to select a value for PERT, it is required to choose the values at the time of the estimation from the disclosed values of the team before the team's discussion or consensus is made:

- In case, the only one value disclosed as a size of the story then the proposed PERT is not applicable for that particular story.
- If the two or more different values have the same frequencies, then the greater, smaller, and the value in between greater and smaller are considered for Ps, Os, and Ms respectively.

6.3.2.4 Independent Variables for Treatment-3

The independent variables of Treatment-1 and Treatment-3 were the same if this treatment was applied by the team using option-1 but for option-2, the following independent variables were also considered:

a) Size of the particular story from the subset of the team using existing approach ($Size_T$)
b) Size of the particular story from the subset of the team using BUA ($Size_B$)
c) Average size of the story calculated by sum of $Size_T$ and $Size_B$ and dividing it by two.

6.3.2.5 Independent Variables for Treatment-4

All the independent variables of Treatments-1, 2 and 3 were also considered independent variables of Treatment-4 with minor modifications in the variables/calculations as follows:

a) Estimated size (Es) based on optimistic (Os), most likely (Ms) and pessimistic size (Ps) of the particular story is calculated as per the given criteria under Treatment-2 (for both variants PERT-1 & 2) exclusively for the two subsets of the same team as per the given option-2 of Treatment-3
b) The average of both estimated sizes of the story from the two subsets of the team is considered as the final estimated size (Es) of each story if the team selects option-2 for the application of Treatment-3.

6.3.2.6 Independent Variables for Treatment-5 & 6

The Independent variables of Treatments T1, T3 and the variables that are the constituents of the proposed dynamic model of estimation were considered as the independent variables of Treatment-5 and 6. The list of those factors is as follows:

1. Human context
    a. Team Members' expertise
    b. Team Members' experience
    c. Team relationship and spirit
    d. Stakeholder / User collaboration
2. Systems complexity
3. Organization context

   a. Organization culture
   b. Organization type
   c. Working environment
4. Sprint context
   a. Sprint number
5. Risk context
   a. Known mitigated risks
   b. Known unmitigated risks
   c. Unknown risks

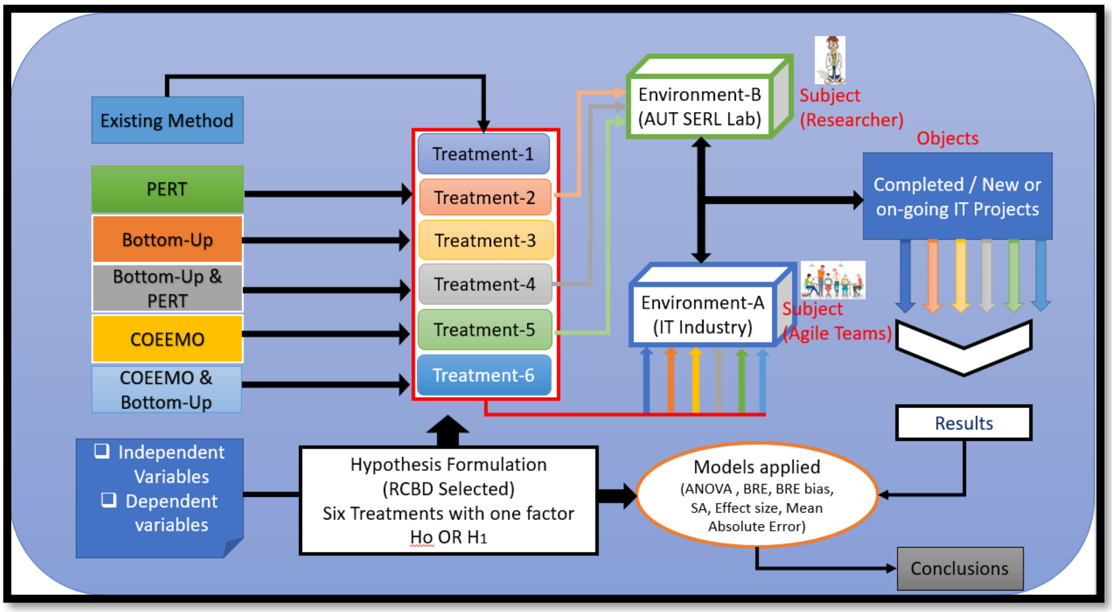


Figure 6.1 Block Diagram —— Experiments Implementation

6.4 Chapter Review

In this chapter, the specifics of the experimental planning including the research hypothesis, the various treatments along with details of the study environment, the dependent and independent variables, and hypothesis testing strategies were all covered.

In the next chapter, the settings for the real industry projects in which the proposed experiments are conducted are described. The chapter also discusses the methods through which the planned treatments are applied to the industry projects. Moreover, various aspects of the experiments including the details of subjects, objects, and important data points of each treatment, are presented.

Chapter-7 Experiments' Execution and Data Collection

***Content Summary**: This chapter covers the details of the environments in which the experiments were conducted. It also discusses the methods through which the planned treatments were applied to IT projects in industry. The various aspects of the experiments are also discussed in the last sections of this chapter, including the subjects, objects, and significant data points of each treatment.*

7.1 Experimental Settings

The experiments were conducted in the real-world environment of an IT company that is itself a subsidiary of an Australian-based international company dealing in building products and construction. As per AUT ethics approval, the actual name of this company cannot be disclosed. Therefore, the fictitious name AUSIT is used to refer to the experiment host organization. AUSIT is involved in the development of software projects ranging from Mobile/Web applications to ERP solutions for their clients. AUSIT follows an agile/scrum framework for the development of their IT-related projects. The proposed treatments were applied on three real AUSIT projects (i.e., the objects of the experiment), which are named as Xproject, Yproject, and Zproject throughout this experimentation. These projects were developed by two different AUSIT development teams (i.e., the subjects of the experiment) who are referred to in this experiment as Team-A and Team-B. Further details of the teams, treatments, and projects are given in the subsequent sections.

The data collected through the experiments was shared on the Google/Excel sheets with the researcher (templates of Excel sheets used for the collection of the data for the different treatments are attached at Appendix-C). These templates were developed, refined and finalized collectively by the researcher and the Agile coach / Capability lead of the teams. As per the agreement with the company, the Capability lead of the AUSIT teams was responsible to enter the required data on the given templates and to share them with the researcher on the same day as completed, or at least before the next meeting, i.e., the next sprint planning or retrospective meeting. Other interactions were common - the researcher was available via SMS/phone and 44 times chats/calls were exchanged, 5 video meetings were conducted over Skype, and around 150 Whatsapp chats/calls occurred. This enabled the researcher to answer any queries of the team/Capability Lead and to provide clarity on the applications of the proposed treatments.

7.1.1 Xproject Details

Xproject centred around an ERP system and was an on-going project of AUSIT when it was included in the proposed experiment. Development and testing of this product started in 2017 and was completed in September 2018. This project was developed by Team-A, which consisted of 10 developers. Further, the team worked strictly using the agile/scrum framework for their development and testing activities throughout the Xproject.

This project was completed in 34 sprints where the duration of each sprint was 2 weeks. Team-A started to apply the proposed treatments on this on-going project from Sprint#27-32. Team-A applied three treatments (i.e., Treatment-1, 2, and 5) on the Xproject. As per the plan and agreement of the researcher with the team/agile coach, it was decided to apply those treatments (i.e., 1, 2 and 5) and not treatments 3 and 4 on the Xproject, as this would not require much time or training of the team thus ensuring that the on-going project could be delivered on time. Instructions to apply the proposed treatments on the Xproject were provided to Team-A, as placed at Appendix-C. The data collected from the Xproject is shared in Table 7.1.

7.1.2 Yproject and Zproject Details

Yproject and Zproject were new, small-scale AUSIT projects which were planned to include in this proposed study. In these projects, the team had to develop and test two Web systems along with their mobile interfaces. Both projects were undertaken in 2019 by Team-B of AUSIT, which comprised five developers. The projects were completed within four months in a total 12 sprints (i.e., six sprints in each of the projects while each sprint's duration was 2 weeks). Team-B followed strictly the agile/scrum framework for their development and testing activities.

Team-B applied the proposed treatments on alternate sprints of the two projects; i.e., in the Yproject treatments 1, 2, and 5 were applied on Sprint #1, 3 and 5 while treatments 3, 4 and 5 were applied on Sprint # 2, 4, and 6); and in the Zproject treatments 1, 2, and 5 were applied on Sprint # 2, 4 and 5 while treatments 3, 4, and 5 were applied on Sprint # 1, 3, and 6. Instructions on how to apply the proposed treatments on the Yproject and Zproject were provided to the team, as placed in Appendix-C. In addition to the instructions, the agile coach / Capability Lead of the team also conducted training sessions with the team on applying the Bottom-Up approach in their estimation activities, because this was the only technique that the team was not practicing in their previous projects for the purpose of estimating effort. The data collected from the Yproject and Zproject are shared in Table 7.1.

S#	SPT #	US	T1 TESPs	T2a (PERT-1) TESPs	T2b (PERT-2) TESPs	T3 TESPs	T4a (PERT-1) TESPs	T4b (PERT-2) TESPs	T5 TESPs	T6 TESPs	AFSPs
Experimental Data: (Proposed subjects are Team-A and Team-B, Agile coach / Capability Lead, and the Researcher, while objects are Xproject, Yproject & Zproject)											
Xproject Data: - Subjects/Treatments: Team-A, Agile coach, Researcher / T1, T2, T5											
1	27	10	37	35.3	34	-	-	-	33.3	-	32
2	28	8	45	45.8	45	-	-	-	40.7	-	42
3	29	8	49	41.8	41	-	-	-	44.5	-	44
4	30	12	78	71.2	65.2	-	-	-	71.3	-	57
5	31	8	77	67.2	58.2	-	-	-	70.8	-	77
6	32	12	62	54.7	47.7	-	-	-	57.3	-	49
Yproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / T1, T2, T3, T4, T5, T6											
7	1	5	39	28.5	22.5	-	-	-	29.9	-	31
8	2	5	-	-	-	34	37.7	32.7	-	27.9	31
9	3	5	47	34.5	30.5	-	-	-	41.1	-	39
10	4	5	-	-	-	42	45.2	40.2	-	38	37
11	5	5	39	30.7	26.2	-	-	-	36.4	-	39
12	6	5	-	-	-	32	31.2	27.2	-	30.8	32
Zproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / T1, T2, T3, T4, T5, T6											
13	1	5	-	-		34	30	26.5	-	26.1	31
14	2	5	42	37.7	32.7	-	-		34.5	-	37
15	3	5	-	-	-	45	37.3	30.8	-	39.4	40
16	4	5	52	45.2	40.2	-	-	-	47	-	44
17	5	5	37	30	26.5	-	-	-	34.5	-	34
18	6	5	-	-	-	32	31.2	27.2	-	30.8	32
S#: Sample number **SPT#:** Sprint number **US:** User stories **T1, T2, T3, T4, T5, T6:** Treatments 1, 2, 3, 4, 5, 6 **TESPs:** Total Estimated Story-Points per sprint **AFSPs:** Actual Finished SPs **PERT-1 & 2:** Variant-1 & 2 of PERT											

Table 7.1 Experimental Data of AUSIT's Projects X, Y and Z

7.2 Treatments as Applied to AUSIT's Projects

The six different treatments T1 to T6 were applied across the three AUSIT projects (i.e., Xproject, Yproject, and Zproject). The estimation data per sprint was recorded in story points (SP) by the agile coach on the agreed upon templates of the Excel sheets. The data collection sheets for all treatments and projects are given at Appendix-D. The exact dates of the sprints and other project/team identifying information are removed from the data to comply with the ethics protocol of the AUT ethics committee. The methodology of each treatment with respect to the project is explained in the subsequent subsections.

7.2.1 Treatment-1 and Important Data Points (DPs)

Treatment-1 – a blend of Planning Poker and Expert judgment techniques – was applied on all sprints of Xproject and alternate sprints of Yproject (i.e., Sprint# 1, 3, and 5) and Zproject (i.e., Sprint# 2, 4, and 5) by the development teams of AUSIT. During this treatment the developers estimated stories by employing their normal methodology that was based on gut feelings, analogies, previous experiences of similar or other projects and/or a top-down approach but they were not allowed to use a Bottom-Up method for their estimations. Table 7.2 demonstrates the important DPs that the agile coach was required to record on the provided Excel sheets were the total number of stories estimated per sprint, agreed size/estimate of the story within the team in SP against each story, Sprint number, finished SP (i.e., the story points which were completed by the team as the respective sprint concludes) and unfinished SP (i.e., story points which were not completed by the team at the conclusion of the corresponding sprint).

Sample Number	Sprint No.	No. of User Stories	Agreed SPs per Story / Sprint	Treatment-1 Agreed Team Estimated Sprint's Total SP	Finished Sprint's SP	Unfinished Sprint's SP
Xproject:						
1	27	10	5, 3, 8, 5, 5, 3, 3, 3, 1, 1	37	32	5
2	28	8	5, 3, 8, 5, 5, 8, 8, 3	45	42	3
3	29	8	21, 13, 5, 5, 2, 1, 1, 1	49	44	5
4	30	12	2, 13, 2, 13, 3, 5, 3, 5, 3, 13, 13, 3	78	57	21
5	31	8	21, 13, 13, 13, 8, 5, 2, 2	77	77	0
6	32	12	8, 8, 5, 5, 5, 8, 5, 3, 3, 8, 2, 2	62	49	13
Yproject:						
7	1	5	8, 13, 8, 5, 5	39	31	8
9	3		13, 8, 5, 13, 8	47	39	8
11	5		8, 13, 8, 5, 5	39	39	0
Zproject:						
14	2	5	13, 13, 8, 3, 5	42	37	5
16	4		13, 13, 13, 5, 8	52	44	8
17	5		8, 13, 8, 3, 5	37	34	3

Table 7.2 Treatment-1's Important DPs of Xproject, Yproject and Zproject

7.2.2 Treatment-2 and Important Data Points (DPs)

Treatment-2 – a combination of Planning poker, Expert Judgement, and PERT with two variants i.e., PERT-1 & PERT-2 – was applied on all sprints of Xproject and alternate sprints of Yproject (i.e., Sprint# 1, 3, and 5) and Zproject (i.e., Sprint# 2, 4, and 5). The agile coach of the team had recorded the estimates of each individual team member against each story per sprint on the provided template of the Excel sheet.

Treatment-2 (Yproject & Zproject)				
Project Name:		Yproject		
Sprint No:		1	3	5
No of Stories:		03		
Sprint's SPs Estimated by Team:		39	47	39
Sprint's SPs Estimated by PERT-1:		28.5	34.5	30.7
Sprint's SPs Estimated by PERT-2:		22.5	30.5	26.17
Team Member wise SPs against individual Story (5 Developers)	**S-1**	5, 5, 8, 8, 5	13, 13, 8, 8, 8	5, 5, 8, 8, 5
	S-2	13, 13, 8, 13, 13	8, 8, 5, 5, 8	13, 13, 8, 13, 13
	S-3	8, 5, 8, 8, 8	3, 3, 5, 3, 3	8, 5, 5, 8, 13
	S-4	5, 3, 5, 5, 3	8, 8, 13, 13, 13	5, 3, 5, 5, 3
	S-5	5, 3, 3, 5, 5	8, 8, 8, 8, 8	5, 3, 3, 5, 5
Project Name:		Zproject		
Sprint No:		2	4	5
No of Stories:		03		
Sprint's SPs Estimated by Team:		42	52	37
Sprint's SPs Estimated by PERT-1:		37.7	45.2	30
Sprint's SPs Estimated by PERT-2:		32.7	40.2	26.5
Team Member wise SPs against individual Story (5 Developers)	**S-1**	13, 13, 13, 8, 8	13, 13, 8, 8, 13	5, 5, 5, 8, 8
	S-2	13, 13, 13, 13, 13	13, 13, 13, 13, 13	13, 13, 8, 13, 13
	S-3	5, 8, 5, 8, 8	13, 13, 13, 8, 8	8, 5, 5, 13, 8
	S-4	5, 5, 5, 8, 8	5, 5, 8, 5, 8	5, 3, 3, 2, 3
	S-5	5, 3, 5, 5, 3	13, 8, 8, 13, 8	5, 3, 5, 3, 5

Table 7.3 Treatment-2's Important DPs of Projects X & Y

Table 7.3 and 7.4 present the important DPs that the agile coach was required to record on the provided Excel sheets, similar as given in Treatment-1 but in addition the story size/estimate of each story in SP against each individual member of the team was recorded as revealed during the sprint planning meetings.

Treatment-2 (Xproject)							
Project Name:		Xproject					
Sprint No:		27	28	29	30	31	32
No of Stories:		10	8	8	12	8	12
Sprint's SPs Estimated by Team:		37	45	49	78	77	62
Sprint's SPs Estimated by PERT-1:		35.3	45.83	41.8	71.2	67.2	54.7
Sprint's SPs Estimated by PERT-2:		34	45	41	65.2	58.2	47.7
Team Member wise SPs against individual Story (10 Developers) (Note: **S-1** represents Story-1 and so on)	**S-1**	5, 5, 5, 3, 3, 8, 8, 3, 5, 5	3, 3, 8, 5, 5, 5, 3, 5, 5, 5	21, 21, 13, 13, 21, 13, 21, 21, 21, 13	2, 2, 2, 1, 1, 3, 0.5, 1, 2, 2	21, 21, 13, 13, 21, 13, 21, 21, 21, 13	8, 8, 8, 8, 5, 5, 5, 8, 8, 5
	S-2	3, 3, 3, 3, 2, 2, 1, 3, 3, 1	2, 2, 3, 3, 3, 3, 2, 3, 3, 3	8, 13, 13, 8, 5, 13, 13, 13, 8, 13	8, 8, 13, 13, 13, 13, 8, 13, 8, 13	8, 8, 13, 13, 13, 13, 8, 21, 13, 13	8, 8, 5, 5, 5, 5, 8, 8, 8, 8
	S-3	8, 8, 13, 13, 8, 8, 5, 8, 5, 8	5, 5, 8, 8, 8, 13, 5, 13, 8, 8	5, 5, 5, 8, 5, 8, 8, 5, 5, 8	1, 1, 2, 2, 2, 2, 2, 2, 1, 2	13, 8, 13, 13, 13, 13, 8, 8, 13, 13	5, 5, 5, 5, 3, 3, 8, 5, 5, 3
	S-4	5, 5, 5, 5, 5, 3, 8, 3, 3, 5	5, 5, 3, 8, 5, 5, 5, 3, 5, 5	5, 5, 5, 3, 3, 5, 8, 3, 5, 5	13, 13, 13, 21, 13, 21, 13, 13, 13, 13	8, 13, 13, 13, 13, 13, 13, 13, 13, 13	3, 3, 5, 5, 5, 5, 8, 5, 5, 3
	S-5	5, 5, 5, 5, 5, 3, 5, 3, 3, 5	3, 3, 5, 8, 8, 5, 5, 5, 3, 5	1, 1, 2, 2, 2, 3, 1, 2, 2, 3	1, 1, 2, 3, 3, 3, 3, 2, 3, 3	8, 8, 8, 5, 13, 8, 5, 13, 8, 8	5, 5, 5, 3, 3, 3, 5, 5, 5, 5,
	S-6	3, 3, 3, 3, 2, 2, 5, 3, 5, 5	8, 8, 13, 13, 13, 8, 8, 8, 8, 5	0.5, 0.5, 0.5, 1, 1, 1, 1, 0.5, 1, 1,	8, 5, 5, 5, 3, 8, 5, 5, 5, 5	8, 5, 5, 5, 5, 3, 3, 5, 5, 5	5, 8, 8, 8, 8, 8, 5, 5, 8, 8
	S-7	3, 3, 3, 3, 2, 2, 1, 3, 3, 1	5, 5, 8, 8, 8, 13, 8, 8, 8, 5	1, 1, 1, 1, 1, 1, 1, 0.5, 2, 1	5, 3, 3, 5, 5, 3, 3, 3, 3, 3	3, 1, 1, 1, 2, 2, 2, 2, 2, 1	5, 5, 5, 5, 3, 3, 5, 8, 5, 5
	S-8	3, 5, 3, 5, 2, 2, 3, 3, 3, 3	2, 1, 1, 3, 3, 5, 3, 3, 2, 3	2, 2, 1, 1, 1, 1, 0.5, 1 0.5, 0.5	5, 5, 5, 8, 8, 3, 5, 5, 5, 5	3, 1, 2, 2, 2, 2, 2, 3, 1, 3	2, 2, 1, 3, 3, 3, 3, 3, 5, 3
	S-9	0.5, 0.5, 0.5, 1, 1, 1, 1, 1, 1, 0.5			2, 3, 3, 3, 3, 5, 2, 3, 3, 5		3, 3, 3, 3, 3, 2, 2, 5, 3, 3
	S-10	1, 1, 1, 1, 1, 0.5, 1, 0.5, 1, 0.5			13, 13, 13, 13, 8, 21, 13, 8, 13, 13		5, 5, 8, 8, 8, 8, 8, 8, 5, 8
	S-11				13, 13, 13, 13, 8, 8, 13, 13, 13, 8		1, 2, 2, 2, 2, 3, 5, 2, 2, 1
	S-12				5, 5, 5, 3, 3, 2, 3, 3, 3, 3		1, 1, 1, 2, 2, 2, 2, 2, 3, 2

Table 7.4 Treatment-2's Important DPs of Project X

7.2.3 Treatment-3 and Important Data Points (DPs)

Treatment-3 – a blend of Planning poker and Expert Judgement in a strictly Bottom-Up approach – was applied on the alternate sprints of two AUSIT projects i.e., Yproject (on Sprint #2, 4 and 6) and Zproject (on Sprint #1, 3, and 6). The development teams had conducted their estimation activities in all sprints' planning meetings by following their normal method of planning poker and estimated the stories in story points (SP) through an expert Bottom-Up approach based on the work break down structure (WBS) of the complexities perceived in the development and testing activities that were to be carried out during the actual implementation of the estimated story.

Table 7.5 exhibits the important DPs that the agile coach was required to record on the provided Excel sheets, the same as given in Treatment-1.

Sample Number	Sprint No.	No. of User Stories	Agreed SPs per Story / Sprint	Treatment-3 Agreed Team Estimated Sprint's Total SP	Finished Sprint's SP	Unfinished Sprint's SP
Yproject:						
8	2	5	8, 13, 5, 5, 3	34	31	3
10	4		8, 13, 8, 5, 8	42	37	5
12	6		8, 8, 8, 5, 3	32	32	0
Zproject:						
13	1	5	5, 13, 8, 3, 5	34	31	3
15	3		13, 8, 3, 13, 8	45	40	5
18	6		8, 8, 8, 5, 3	32	32	0

Table 7.5 Treatment-3's Important DPs of Projects Y & Z

7.2.4 Treatment-4 and Important Data Points (DPs)

Treatment-4 – a mixture of Planning poker and Expert Judgement in a strictly Bottom-Up approach, and PERT with two variants PERT-1 & PERT-2 – was applied on the alternate sprints of the AUSIT's latter on two smaller projects, i.e., Yproject (on Sprint #2, 4, and 6) and Zproject (on Sprint #1, 3, and 6). The development teams had conducted their estimation activities in all sprints' planning meetings by following their normal method of planning poker. The team estimated the stories in story points (SP) through an expert Bottom-Up approach based on the work breakdown structure (WBS) of the complexities perceived in different development and testing activities that were to be carried out during the actual implementation of the estimated story. The agile coach of the team recorded the estimates of each individual team members against each story of the alternate sprints on the provided template of the Excel sheet. Table 7.6 shows that the important DPs that the agile coach was required to record on the provided Excel sheets were that same as those given in Treatment-1 and Treatment-2.

Treatment-4				
Project Name:		Yproject		
Sprint No:		2	4	6
No of Stories:		03		
Sprint's SPs Estimated by team:		34	42	32
Sprint's SPs Estimated by PERT-1:		37.7	45.2	31.2
Sprint's SPs Estimated by PERT-2:		32.7	40.2	27.2
Team Member wise SPs against individual Story (05 Developers)	**S-1**	5, 5, 8, 8, 5	5, 5, 8, 8, 5	13, 13, 8, 8, 13
	S-2	13, 13, 13, 13, 13	13, 13, 13, 13, 13	13, 8, 8, 8, 13
	S-3	5, 5, 8, 8, 8	13, 13, 8, 8, 13	5, 8, 8, 8, 8
	S-4	5, 5, 8, 8, 5	5, 5, 5, 8, 8	5, 5, 8, 8, 5
	S-5	5, 3, 3, 5, 5	13, 13, 8, 8, 8	5, 3, 3, 3, 5
Project Name:		Zproject		
Sprint No:		1	3	6
No of Stories:		03		
Sprint's SPs Estimated by team:		34	45	32
Sprint's SPs Estimated by PERT-1:		30	37.33	31.2
Sprint's SPs Estimated by PERT-2:		26.5	30.83	27.2
Team Member wise SPs against individual Story (05 Developers)	**S-1**	5, 8, 5, 8, 5	13, 13, 13, 13, 8	13, 8, 13, 8, 13
	S-2	13, 13, 13, 8, 13	8, 8, 5, 8, 5	13, 8, 8, 8, 13
	S-3	8, 5, 8, 5, 13	3, 3, 2, 5, 3	8, 5, 8, 8, 8
	S-4	5, 3, 3, 2, 3	8, 13, 8, 13, 13	5, 5, 8, 5, 8
	S-5	5, 3, 5, 3, 5	8, 8, 8, 8, 8	5, 3, 3, 5, 3

Table 7.6 Treatment-4's Important DPs of Projects Y & Z

7.2.5 Treatment-5 and Important Data Points (DPs)

This treatment – the proposed model of estimation (COEEMO) along with the combination of other techniques: planning poker, and Expert Judgement – was applied on all sprints of the Xproject and alternate sprints of the Yproject (i.e., Sprint# 1, 3, and 5) and Zproject (i.e., Sprint# 2, 4, and 5).

Treatment-5		
Project Name	Xproject	
Sprint Numbers	All Sprints (i.e., from Sprint# 27 – 32)	**Conversions of ratings into values as given per Table 4.1**
Team Members' Expertise	3 Members of "High expertise" with "Very High" rating	3 * 0.3 = 0.9
	5 Members of "Medium expertise" with "High" rating	5 * 1.44= 7.2
	2 Members of "Adequate expertise" with "Moderate" rating	2 * 0.82 = 1.64
Team Expertise - calculated as per method given in Chapter-4 subsection 4.6.1.1		9.74/10 = **0.97**
Team Members' Experience	3 Members of "Proficient level" with "Very High" rating	3 * 0.2 = 0.6
	5 Members of "Experienced level" with "High" rating	5 * 0.85 = 4.25
	2 Members of "Intermediate level" with "Moderate" rating	2 * 1.7 = 3.4
Team Experience - calculated as per method given in Chapter-4 subsection 4.6.1.2		8.25/10 = **0.83**
Stakeholder Coordination	8 Stakeholders are "Fully" cooperative with "Very High" scaling / rating	8 * 0.25 = 2
	2 Stakeholders are "Partially" cooperative with "Very High" scaling / rating	2 * 0.6 = 1.2
Stakeholder Coordination - calculated as per method given in Chapter-4 subsection 4.6.1.3		3.2/10 = **0.32**
Team Relationship & Spirit	Team's level of Relationship is "Very Good" with "High" rating/scaling	1.21
System Complexity	Sys-2 (System's type is Enterprise with "Very High" complexity rating/scaling)	0.8
Organization's Type	Organization type is "Joint Sector" with "Very High" rating	0.6
Organization's Culture	Organization culture is "Strong" with "Very High" grading	0.15
Organization's Work-Environment	Work environment is "Best" with "High" rating	0.3
Project's Risks	Known Planned risks with "Moderate" grading	2.5
	Known Unplanned risks with "High" grading	-0.4
	Unknown risks with "High" grading	-1.3

Transitional Factors - calculated as per method given in Chapter-4 subsection 4.6.1.6	HI	Tcap	OEI	Tprd
	3.33	0.85	1.05	0.48

Sprint(s)-wise Story Points (SPs) – TF, EnvF, EF, LF, CEAF calculated as per methods given in Chapter-4 subsection 4.6.1.6

	Sprint-27	Sprint-28	Sprint-29	Sprint-30	Sprint-31	Sprint-32
Sprint(s) Complexity (Very High for all sprints)	0.66	0.67	0.68	0.70	0.71	0.72
Learning Factor	0.16	0.16	0.17	0.17	0.18	0.18
COEEMO Effort Adjustment Factor (CEAF)	0.90	0.90	0.91	0.91	0.92	0.92

Technical Factor (TF)	**Environmental Factor (EnvF)**	**Effort Factor (EF)**
0.73	1.01	0.74

	Sprint-27	Sprint-28	Sprint-29	Sprint-30	Sprint-31	Sprint-32
Sprint's SPs Estimated by COEEMO	33.3	40.7	44.5	71.3	70.8	57.3

Table 7.7 Treatment-5's Important DPs of Xproject

The important DPs that the agile coach was required to record on the provided Excel sheets were the same as those given in Treatment-1 and Treatment-2 along with the 10 significant factors of the proposed model as given in Chapter 4. The DPs values of the factors along with the ratings provided by the agile coach for all three projects are given in Table 7.7 and 7.8, respectively.

Treatment-5						
Project Name	Yproject & Zproject					
Sprint Numbers	Sprint# 1, 3, and 5 of Yproject Sprint# 2, 4, and 5 of Zproject					**Conversions of ratings into values as given per Table 4.1**
Team Members' Expertise	1 Member of "High expertise" with "High" rating					1 * 0.34 = 0.34
	2 Members of "Medium expertise" with "High" rating					2 * 1.48= 2.96
	2 Members of "Adequate expertise" with "Moderate" rating					2 x 0.74 = 1.48
Team Expertise - calculated as per method given in Chapter-4 subsection 4.6.1.1						4.7/5 = **0.97**
Team Members' Experience	3 Members of "Experienced level" with "High" rating					3 * 0.85 = 2.55
	2 Members of "Intermediate level" with "Moderate" rating					2 * 1.7 = 3.4
Team Experience - calculated as per method given in Chapter-4 subsection 4.6.1.2						5.95/5 = 1.19
Stakeholder Coordination	2 stakeholders are "Fully" cooperative with "Very High" scaling / rating for Yproject					2 * 0.25 = 5
	4 stakeholders are "Fully" cooperative with "Very High" scaling / rating for Zproject					4 * 0.25 = 1
Stakeholder Coordination - calculated as per method given in Chapter-4 subsection 4.6.1.3						**0.25**
Team Relationship & Spirit	Team's level of Relationship is "Very Good" with "High" rating/scaling					1.21
System Complexity	System's type Sys-2 with "Very High" complexity rating/scaling					2.15
Organization's Type	Organization type is "Joint Sector" with "Very High" rating					0.6
Organization's Culture	Organization culture is "Strong" with "Very High" grading					0.15
Organization's Work-Environment	Work environment is "Best" with "High" rating					0.3
Project's Risks	Known Planned risks with "Moderate" grading					2.5
	Known Unplanned risks with "High" grading					-0.4
	Unknown risks with "High" grading					-1.3
Transitional Factors - calculated as per method given in Chapter-4 subsection 4.6.1.6			HI 3.62	Tcap 1.07	OEI 1.05	Tprd 0.53
Sprint(s)-wise Story Points (SPs) – TF, EnvF, EF, LF, CEAF calculated as per methods given in Chapter-4 subsection 4.6.1.6						
Yproject & Zproject	Sprint-1	Sprint-3	Sprint-5	Sprint-2	Sprint-4	Sprint-5
Sprint(s) Complexity (All Sprints High)	0.12	0.63	0.78	0.40	0.70	0.78
Learning Factor	0.02	0.13	0.18	0.07	0.15	0.18
COEEMO Effort Adjustment Factor (CEAF)	0.77	0.88	0.93	0.82	0.90	0.93
Technical Factor (TF)		**Environmental Factor (EnvF)**			**Effort Factor (EF)**	
0.76		0.99			0.75	
Sprint's SPs Estimated by COEEMO (Yproject/Zproject)	29.9	41.1	36.4	34.5	47	34.5
Note: If any value /rating is same in case of Yproject and Zproject then they are given only once in the table.						

Table 7.8 Treatment-5's Important DPs of Yproject & Zproject

7.2.6 Treatment-6 and Important Data Points (DPs)

This treatment – the proposed model of estimation (COEEMO) along with the combination of other techniques: planning poker, and Expert Judgement in a strictly Bottom-Up approach – was applied on alternate sprints of the Yproject (i.e., Sprint# 2, 4, and 6) and Zproject (i.e., Sprint# 1, 3, and 6) by the development teams of AUSIT. Table 7.9 exhibits the important DPs that the agile coach was required to record on the provided Excel sheets, the same as given in Treatment-1.

<table>
<tr><th colspan="7">Treatment-6</th></tr>
<tr><td>Project Name</td><td colspan="6">Yproject & Zproject</td></tr>
<tr><td>Sprint Numbers</td><td colspan="3">Sprint# 2, 4, and 6 of Yproject
Sprint# 1, 3, and 6 of Zproject</td><td colspan="3">Conversions of ratings into values as given per Table 4.1</td></tr>
<tr><td rowspan="3">Team Members' Expertise</td><td colspan="3">1 Member of "High expertise" with "High" rating</td><td colspan="3">1 * 0.34 = 0.34</td></tr>
<tr><td colspan="3">2 Members of "Medium expertise" with "High" rating</td><td colspan="3">2 * 1.48= 2.96</td></tr>
<tr><td colspan="3">2 Members of "Adequate expertise" with "Moderate" rating</td><td colspan="3">2 x 0.74 = 1.48</td></tr>
<tr><td colspan="4">Team Expertise - calculated as per method given in Chapter-4 subsection 4.6.1.1</td><td colspan="3">4.7/5 = 0.97</td></tr>
<tr><td rowspan="2">Team Members' Experience</td><td colspan="3">3 Members of "Experienced level" with "High" rating</td><td colspan="3">3 * 0.85 = 2.55</td></tr>
<tr><td colspan="3">2 Members of "Intermediate level" with "Moderate" rating</td><td colspan="3">2 * 1.7 = 3.4</td></tr>
<tr><td colspan="4">Team Experience - calculated as per method given in Chapter-4 subsection 4.6.1.2</td><td colspan="3">5.95/5 = 1.19</td></tr>
<tr><td rowspan="2">Stakeholder Coordination</td><td colspan="3">2 stakeholders are "Fully" cooperative with "Very High" scaling / rating for Yproject</td><td colspan="3">2 * 0.25 = 5</td></tr>
<tr><td colspan="3">4 stakeholders are "Fully" cooperative with "Very High" scaling / rating for Zproject</td><td colspan="3">4 * 0.25 = 1</td></tr>
<tr><td colspan="4">Stakeholder Coordination - calculated as per method given in Chapter-4 subsection 4.6.1.3</td><td colspan="3">0.25</td></tr>
<tr><td>Team Relationship & Spirit</td><td colspan="3">Team's level of Relationship is "Very Good" with "High" rating/scaling</td><td colspan="3">1.21</td></tr>
<tr><td>System Complexity</td><td colspan="3">System's type Sys-2 with "Very High" complexity rating/scaling</td><td colspan="3">2.15</td></tr>
<tr><td>Organization's Type</td><td colspan="3">Organization type is "Joint Sector" with "Very High" rating</td><td colspan="3">0.6</td></tr>
<tr><td>Organization's Culture</td><td colspan="3">Organization culture is "Strong" with "Very High" grading</td><td colspan="3">0.15</td></tr>
<tr><td>Organization's Work-Environment</td><td colspan="3">Work environment is "Best" with "High" rating</td><td colspan="3">0.3</td></tr>
<tr><td rowspan="3">Project's Risks</td><td colspan="3">Known Planned risks with "Moderate" grading</td><td colspan="3">2.5</td></tr>
<tr><td colspan="3">Known Unplanned risks with "High" grading</td><td colspan="3">-0.4</td></tr>
<tr><td colspan="3">Unknown risks with "High" grading</td><td colspan="3">-1.3</td></tr>
<tr><td colspan="3">Transitional Factors - calculated as per method given in Chapter-4 subsection 4.6.1.6</td><td>HI
3.62</td><td>Tcap
1.07</td><td>OEI
1.05</td><td>Tprd
0.53</td></tr>
<tr><td colspan="7">Sprint(s)-wise Story Points (SPs) – TF, EnvF, EF, LF, CEAF calculated as per methods given in Chapter-4 subsection 4.6.1.6</td></tr>
<tr><td>Yproject & Zproject</td><td>Sprint-2</td><td>Sprint-4</td><td>Sprint-6</td><td>Sprint-1</td><td>Sprint-3</td><td>Sprint-6</td></tr>
<tr><td>Sprint(s) Complexity
(All Sprints High)</td><td>0.40</td><td>0.70</td><td>0.85</td><td>0.12</td><td>0.63</td><td>0.85</td></tr>
<tr><td>Learning Factor</td><td>0.07</td><td>0.15</td><td>0.21</td><td>0.02</td><td>0.13</td><td>0.21</td></tr>
<tr><td>COEEMO Effort Adjustment Factor (CEAF)</td><td>0.82</td><td>0.90</td><td>0.96</td><td>0.77</td><td>0.88</td><td>0.96</td></tr>
<tr><td colspan="2">Technical Factor (TF)</td><td colspan="2">Environmental Factor (EnvF)</td><td colspan="3">Effort Factor (EF)</td></tr>
<tr><td colspan="2">0.76</td><td colspan="2">0.99</td><td colspan="3">0.75</td></tr>
<tr><td>Sprint's SPs Estimated by COEEMO (Yproject/Zproject)</td><td>27.9</td><td>38</td><td>30.8</td><td>26.1</td><td>39.4</td><td>30.8</td></tr>
<tr><td colspan="7">Note: If any value /rating is same in case of Yproject and Zproject then they are given only once in the table.</td></tr>
</table>

Table 7.9 Treatment-6's Important DPs of Yproject & Zproject

7.2.7 Treatments Vs Samples Matrix

There were thus a total of 18 samples collected from the experiments. The specific treatments applied (a number of times) to the specific samples is shown in Table 7.10. As illustrated, Treatments-1 and 2 were employed on 12 samples. Further, six samples were subjected to Treatments-3, 4 and 6.

Sample#	T1	T2	T3	T4	T5	T6	Treatments/ Sample
1	X	X	-	-	X	-	3
2	X	X	-	-	X	-	3
3	X	X	-	-	X	-	3
4	X	X	-	-	X	-	3
5	X	X	-	-	X	-	3
6	X	X	-	-	X	-	3
7	X	X	-	-	X	-	3
8	-	-	X	X	-	X	3
9	X	X	-	-	X	-	3
10	-	-	X	X	-	X	3
11	X	X	-	-	X	-	3
12	-	-	X	X	-	X	3
13	-	-	X	X	-	X	3
14	X	X	-	-	X	-	3
15	-	-	X	X	-	X	3
16	X	X	-	-	X	-	3
17	X	X	-	-	X	-	3
18	-	-	X	X	-	X	3
Samples / Treatments	12 T1	12 T2	06 T3	06 T4	12 T5	06 T6	

Table 7.10 Treatments Vs Samples Matrix

7.2.8 Implementation of the TESPs as recommended by Treatments 2, 4 & 5

There were some implementation issues to apply to total estimated SPs (TESPs) per sprint as suggested by treatments 2, 4, and 5 in an actual scenario as suggested. These are discussed in the subsequent subsections along with their possible solutions.

7.2.8.1 Rounding Issue

In the first step, all TESPs as suggested by treatments 2, 4, and 5 must be rounded by employing the simple mathematics rounding rule that says if the first digit after the decimal is 5 or greater than the TESPs value is rounded up to the nearest whole number, otherwise it is rounded down to the nearest whole number. For example, if the TESPs value is 35.61 then this value requires to be rounded up because the first digit after decimal is 6, which is greater than 5, and so the suggested nearest whole number is 36. However, if the TESPs value is 35.45 then this value needs to be rounded down and the recommended nearest whole value is 35. This rule is applied on all the TESPs values as proposed by treatments 2, 4, and 5. The rounded data are exhibited in Table 7.11.

Experimental Data: (Proposed subjects are Team-A and B, Agile coach / Capability Lead, and the Researcher while objects are Xproject, Yproject & Zproject)											
S#	**SPT #**	**US**	**T1 TESPs**	**T2a (PERT-1) TESPs**	**T2b (PERT-2) TESPs**	**T3 TESPs**	**T4a (PERT-1) TESPs**	**T4b (PERT-2) TESPs**	**T5 TESPs**	**T6 TESPs**	**AFSPs**
Xproject Data: - Subjects/Treatments: Team-A, Agile coach, Researcher / T1, T2, T5											
1	27	10	37	35	34	-	-	-	33	-	32
2	28	8	45	46	45	-	-	-	41	-	42
3	29	8	49	42	41	-	-	-	45	-	44
4	30	12	78	71	65	-	-	-	71	-	57
5	31	8	77	67	58	-	-	-	71	-	77
6	32	12	62	55	48	-	-	-	57	-	49
Yproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / TMT-1, TMT-2, TMT-3, TMT-4, TMT-5, TMT-6											
7	1	5	39	29	23	-	-	-	30	-	31
8	2	5	-	-	-	34	38	33	-	28	31
9	3	5	47	35	31	-	-	-	41	-	39
10	4	5	-	-	-	42	45	40	-	38	37
11	5	5	39	31	26	-	-	-	36	-	39
12	6	5	-	-	-	32	31	27	-	31	32
Zproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / TMT-1, TMT-2, TMT-3, TMT-4, TMT-5, TMT-6											
13	1	5	-	-		34	30	27	-	26	31
14	2	5	42	38	33	-	-	-	35	-	37
15	3	5	-	-	-	45	37	31	-	39	40
16	4	5	52	45	40	-	-	-	47	-	44
17	5	5	37	30	27	-	-	-	35	-	34
18	6	5	-	-	-	32	31	27	-	31	32

S#: Sample number **SPT#:** Sprint number **US:** User stories **T1, T2, T3, T4, T5, T6:** Treatments 1, 2, 3, 4, 5, 6 **TESPs:** Total Estimated Story-Points per sprint **AFSPs:** Actual Finished SPs **PERT-1 & 2:** Variant-1 & 2 of PERT

Table 7.11 Rounded TESPs of the Experimental Data of AUSIT's Xproject, Yproject & Zproject

7.2.8.2 Sprint's backlog Issues

How should the team implement the Treatments' suggested TESPs in their actual work environment for the forthcoming sprint backlog? To answer this question, a range of scenarios are now discussed along with the possible solutions and the same steps/solutions were applied on all the TESPs as recommended by Treatments 2 to 6. The new adjusted data due to the changes in SPs for Xproject (i.e., sprint# 28, and 31), Yproject (i.e., sprint# 1, 2, 3, 4, 5 and 6) and Zproject(i.e., sprint# 2, 3 and 6) shown in Table 7.12, the values of which were used in this study for the purpose of analysis and other evaluations.

Scenario-1: In the Xproject, the TESPs were 45 and 41 as proposed by the team and Treatment-5, respectively, for sprint-28. The team required to compare and adjust both these TESPs in the sprint backlog by following the given steps:

a. Compare both TESPs of sprint-28 as suggested by the team and Treatment-5 and find the difference - in this case the answer is 4, after subtracting TESPs 41 from 45.

b. Find the user story (US) that is equivalent or close to the size of 4 SP from the list of the agreed-upon sizes which the team has estimated against the individual USs in the sprint, which are 5, 3, 8, 5, 5, 8, 8, 3 against the user stories 1-8 correspondingly. From this given list, the team can choose US 2 or 8 to remove it from their sprint backlog because both these USs are close to the story size that in this case is 4. To remove among same-sized multiple USs, the team may also consider other factors such as their business value or the technical requirements of the user stories.

Scenario-2: In the Yproject, the TESPs were 42 and 38 as recommended by the team and Treatment-3, respectively, for sprint-4. The team required to compare and adjust both these TESPs in the sprint backlog by following the given steps:

a. Compare both TESPs of sprint-4 as indicated by the team and Treatment-3. Find the difference - in this case the answer is 4.
b. Locate the US that is equivalent or close to the size of 4 SPs from the list of the agreed-upon sizes which the team has estimated against the individual USs in the sprint, which are 8, 13, 8, 5, 8 against US 1-5 correspondingly. From this given list, the team can choose US 4 to remove it from their sprint backlog because this is the only US that is close to the size of the 4 SPs, as compared to all other user stories which are relatively larger in size. If the team considers that the candidate US for removal, i.e., US-4, is more important and has higher business value than others, then they can remove any other US from the sprint backlog which may be higher or lower in size but is less important and/or has less business value as compared to the candidate US for exclusion.

Experimental Data: (Proposed subjects are Team-A and B, Agile coach / Capability Lead, and the Researcher while objects are Xproject, Yproject & Zproject)											
S#	**SPT #**	**US**	**T1 TESPs**	**T2a (PERT-1) TESPs**	**T2b (PERT-2) TESPs**	**T3 TESPs**	**T4a (PERT-1) TESPs**	**T4b (PERT-2) TESPs**	**T5 TESPs**	**T6 TESPs**	**AFSPs**
Xproject Data: - Subjects/Treatments: Team-A, Agile coach, Researcher / T1, T2, T5											
1	27	10	37	35	34	-	-	-	33	-	32
2	28	8	45	46	45	-	-	-	42	-	42
3	29	8	49	42	41	-	-	-	45	-	44
4	30	12	78	71	65	-	-	-	71	-	57
5	31	8	77	67	58	-	-	-	70	-	77
6	32	12	62	55	48	-	-	-	57	-	49
Yproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / TMT-1, TMT-2, TMT-3, TMT-4, TMT-5, TMT-6											
7	1	5	39	29	23	-	-	-	31	-	31
8	2	5	-	-	-	34	38	33	-	29	31
9	3	5	47	35	31	-	-	-	42	-	39
10	4	5	-	-	-	42	45	40	-	37	37
11	5	5	39	31	26	-	-	-	34	-	39
12	6	5	-	-	-	32	31	27	-	29	32
Zproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / TMT-1, TMT-2, TMT-3, TMT-4, TMT-5, TMT-6											
13	1	5	-	-		34	30	27	-	26	31
14	2	5	42	38	33	-	-	-	37	-	37
15	3	5	-	-	-	45	37	31	-	42	40
16	4	5	52	45	40	-	-	-	47	-	44
17	5	5	37	30	27	-	-	-	34	-	34
18	6	5	-	-	-	32	31	27	-	29	32
S#: Sample number **SPT#:** Sprint number **US:** User stories **T1, T2, T3, T4, T5, T6:** Treatments 1, 2, 3, 4, 5, 6 **TESPs:** Total Estimated Story-Points per sprint **AFSPs:** Actual Finished SPs **PERT-1 & 2:** Variant-1 & 2 of PERT **Note:** Sprints' backlog adjustments in SPs are shown in yellow colour for the sample# 2, 5, 7, 8, 9, 10, 11, 12, and 14 of the Treatments.											

Table 7.12 Adjusted TESPs of the Experimental Data of AUSIT's Xproject, Yproject & Zproject

7.3 Chapter Review

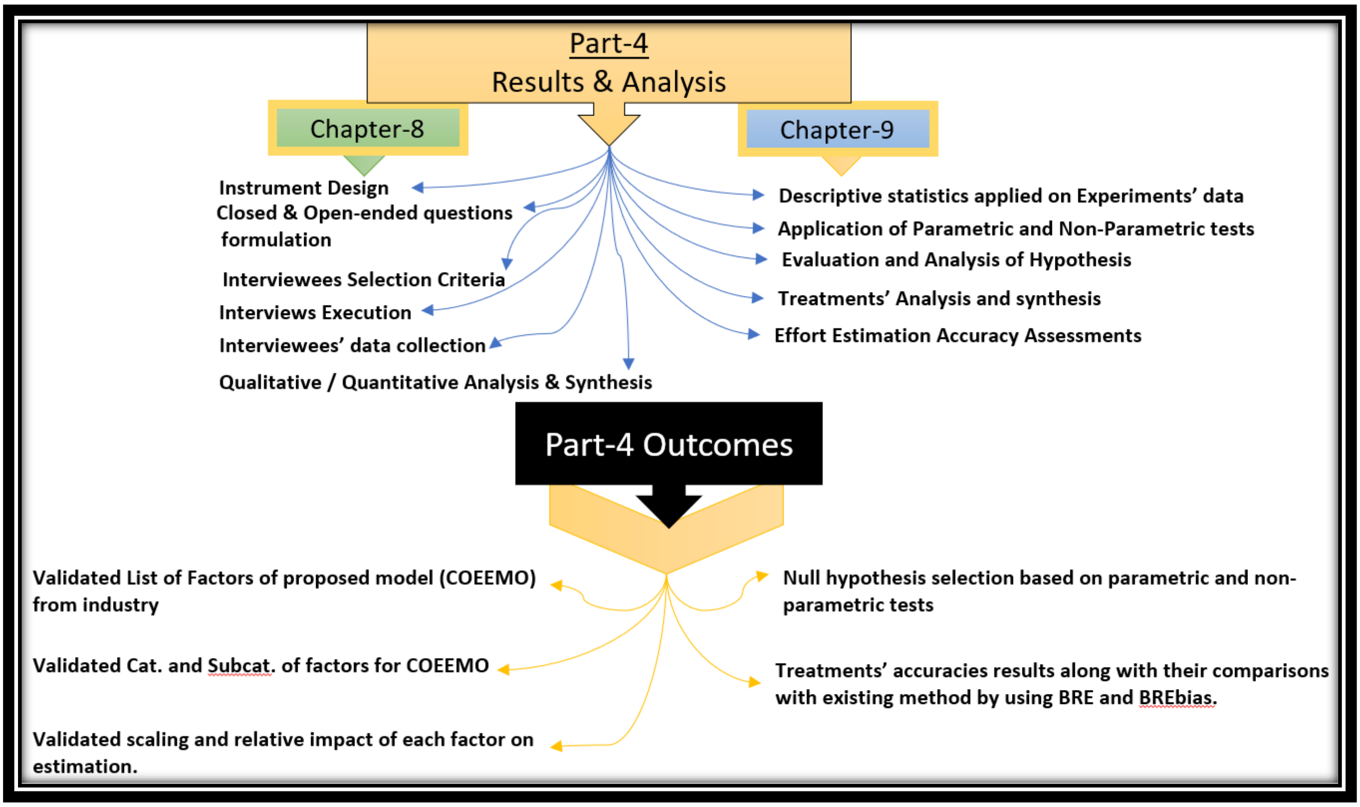
In this chapter, the settings for the real industry projects within which the proposed experiments were conducted, the methods through which the planned treatments were applied, and various aspects of the experiments including the details of subjects, objects, and important data points for each treatment, were presented.

In the next chapter, the design and execution of the instrument used for the interviews of agile developers, and their responses against the three intended topics (i.e., position, product development and estimation), are reported. Moreover, interviewees' responses with supporting graphic/tabular depictions and demographic details are also presented.

PART-4 RESULTS AND ANALYSIS

This part consists of two chapters which reports the conduct and results of the interviews conducted with 21 IT practitioners. This covers details of the instruments used for interviews, and the methods used in the execution of the interviews, including collection and categorization of the data, their interpretation, analysis, and synthesis. Moreover, this part exhibits descriptive statistics of the experiments conducted in real-industry settings along with evaluations of the hypotheses and accuracy comparisons of different treatments.

Detailed views of the research activities with respect to the chapters within this part are presented in the figure as given below. On the top of the diagram, the chapters covered in this part are shown along with their different research endeavours as accomplished within the individual chapters, while the outcomes of Part-4 based on its individual chapters are exhibited at the bottom of the figure.



Chapter-8 Interviews and Analysis

***Content Summary**: This chapter covers the design and execution of the instrument used for the interviews of agile developers. Responses of the interviewees against three intended topics are covered, with supporting graphic and tabular depictions. Demographic details are also reported in the last section of this chapter.*

As per the design of the intended study, interviews with IT practitioners were conducted to validate the factors considered to be significant for the estimation of effort in agile development. The factors themselves have been described fully in Chapter 2 and 4. Interviews were conducted individually with practicing industry professionals to collect data relating to their perceptions of the relevance and importance of the various factors in relation to effort estimation. Audio recordings along with note-taking techniques were used to collect interview data. Daily Interpretive Analysis (DIA) was utilized to assemble and interpret the information as collected from the interviewees at the end of every day of interviewing. Further, thematic analysis by following deductive approach was utilized to identify themes from the interview data (Braun &Clarke, 2006).

Suitably qualified and experienced participants were recruited and given pre-interview guidelines one week prior to the planned meeting. While a minimum of around ten interviews would be sufficient to achieve lend a level of confidence in the insights found we ultimately conducted 21 interviews. We did not necessarily expect consensus regarding all factors; rather, it was assumed that all relevant themes of the said research should be held by a minimum of 50% of the respondents (Galvin, 2015).

8.1 Instrument Used for the Interviews

A semi-structured questionnaire was used to collect data from the interviewees. Open-ended and closed questions were included in this questionnaire. The open-ended questions were included to gain an understanding of the participants' roles and responsibilities, and their experiences and expertise in product/project development and/or estimation in their existing and/or previous organizations. The closed questions were included to capture their perceptions regarding the list of factors that are each thought to have some level of influence on software estimation. The intent was to analyse the relative impact of each factor with respect to estimates of size, effort, and cost of projects undertaken in agile development contexts. Note that the factors included in the questionnaire were not necessarily the only factors that impact on estimation; therefore, suggestions from the interviewees for additional factor(s) were also welcomed.

8.1.1 Instrument Design

Three main topics were covered in the questionnaire, i.e., Position, Product/Project development, and Software estimation/scheduling experience. Their details found in the attached Appendix-K while the high-level structure is given in Figure 8.1.

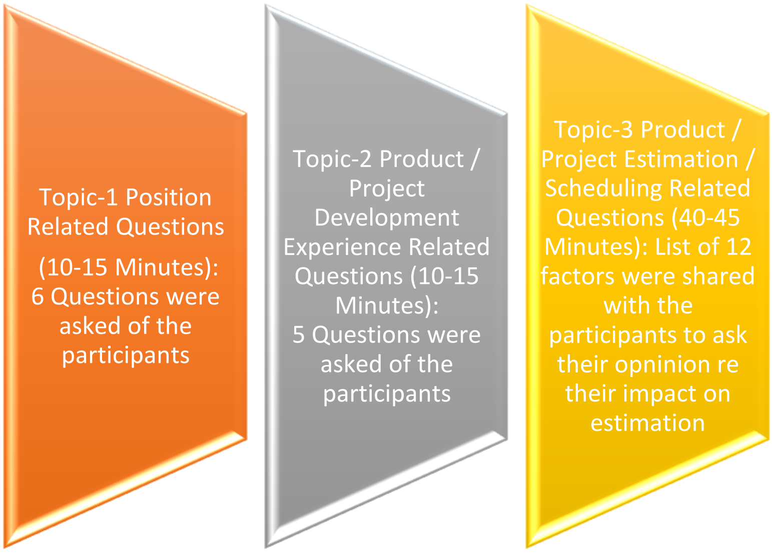


Figure 8.1 Structure of the Instrument for the interview

8.2 Selection Criteria for the Interviewees

We sought personnel actively involved in the development activities of a software project/product that included but was not limited to common practices such as requirements engineering, design, architecture, code development, testing, quality assurance, maintenance and management. They preferably had recent experience in an agile environment, in any one or multiple roles of Product owner, Scrum master, or agile team members, and had not less than 3 years overall experience as an IT practitioner. Details of the procedure used for the recruitment of participants is given in section C.3.5 of the attached Ethics approval document at Annex-L. Moreover, efforts were made to select interviewees from more than one country irrespective of their number of participants (i.e., no upper or lower limit of participants was pre-set for any specific country to ease the process of data collection, and in recognition that software engineering is an international discipline), so that feedback might be gathered from people working in different organisational and national cultures.

Twenty-one IT practitioners were recruited (some voluntarily and a few of them through the researchers' personal references and contacts) from seven different countries as shown in the graph in Figure 8.2. Unsurprisingly, given the researcher's location, the largest number of participants (i.e., 11) were drawn from New Zealand.

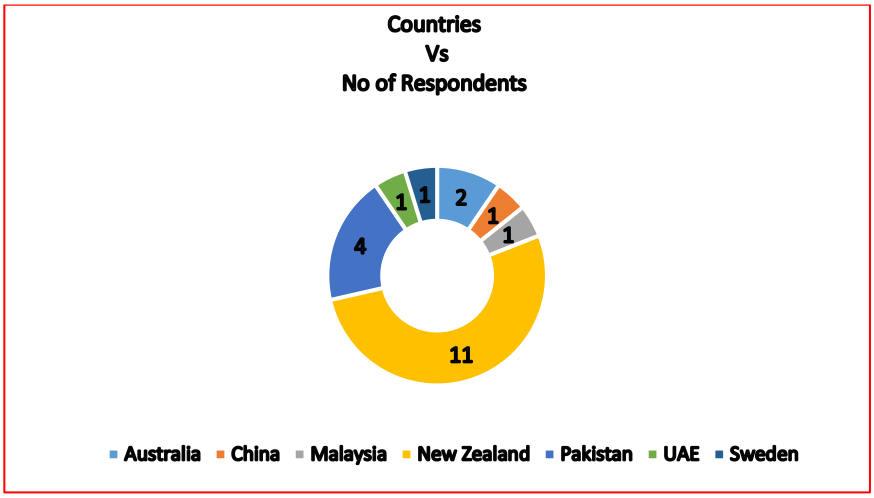


Figure 8.2 Pie Chart Showing Respondents Vs Countries

8.3 Responses of the participants against Topic-1 Position

Under this topic, six questions were framed to collect responses from the interviewees through which they could describe their roles/responsibilities, the extent of their domain experience, and their topmost three areas of expertise. In addition, questions probed their involvement in the development or management of software projects/products. Details of the questions and interviewees' responses are given in the subsequent sections.

Q.1: What is the title of your current position?

The core objective of this question was to provide a sense of the suitability of the respondents in terms of their providing feedback about effort estimations, and the degree to which they were working in a project/product in various roles. Table 8.1 shows that a range of key roles in an agile project/product development were included in the interviews. Moreover, the role frequencies were proportional to an actual working organization for a small agile team (normally from 5-8 members) who were deployed to undertake a real-world project/product in the industry; i.e., to undertake a real industry agile project a team of 5-8 members with one Product owner and one Scrum Master/facilitator is recommended.

S/No	Role	Respondents #
1	Product Owner	1
2	SCRUM Master	3
3	Lead (Quality, Agile Transformation)	3
4	Configuration / Business Analyst	2
5	Developers	6
6	System Architect	1
7	HoD Process Engineering	1
8	CTO / Data Officer	2
9	IT Consultant / Ecommerce Specialist	2

Table 8.1 Interviewees' Role

Q.2: How long (years, months) have you been involved in this occupation?

This question was asked to capture interviewees' length of experience in their professional domains to achieve two key goals. One was to verify their suitability as respondents in this intended study and the second was to better understand their feedback about effort estimation based on the depth of their professional experiences. As the graph given at figure 8.3 shows, the highest number of interviewees (8) had 10-14 years experiences, and no-one had experience of less than 5 years. Moreover, 3 respondents had more than 20 years of experience.

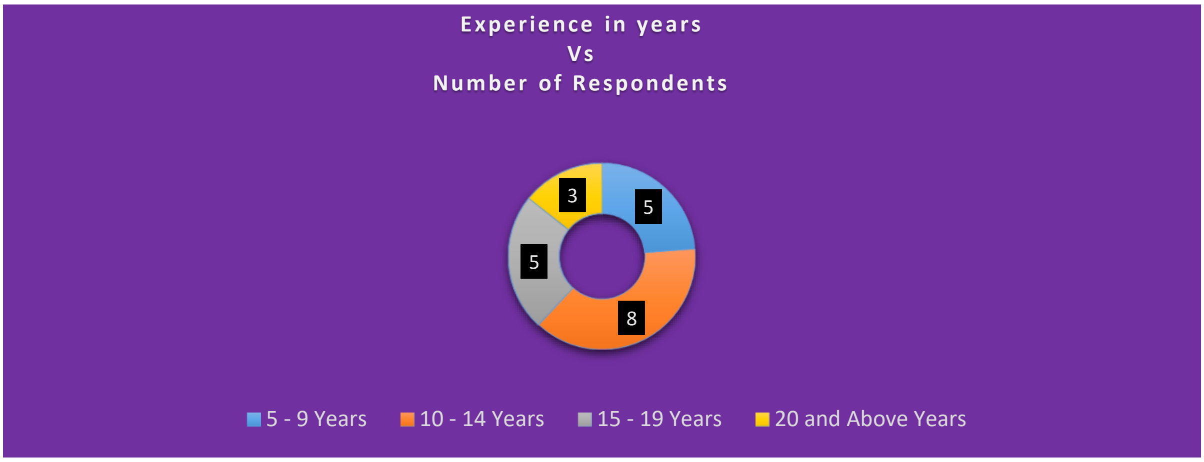


Figure 8.3 Respondents and their experiences in number of years

Q.3. Please list your top three areas of professional expertise.

This question was designed to categorize interviewees' professional expertise based on their own choice of topic and level of expertise. The core objective of asking this question was to provide an understanding of the diversity or similarity in interviewees' expertise and its relationship to their responses regarding estimation. The expertise categories Vs the number of respondents are shown by the bar graph in Figure 8.4 while the types of expertise are listed in Table 8.2.

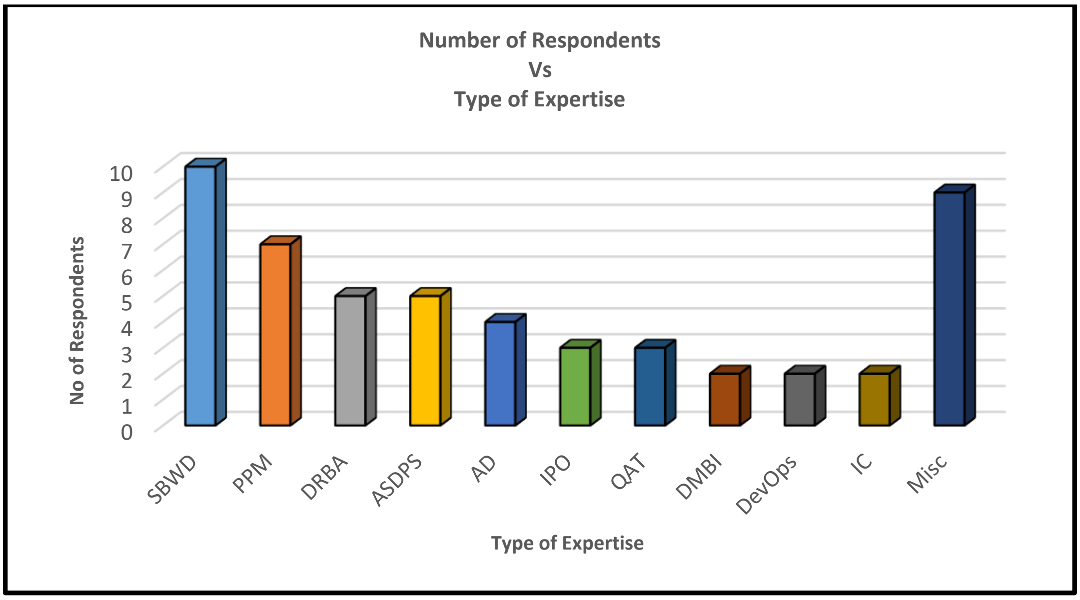


Figure 8.4 Type of expertise Vs number of respondents

The graph shows that 10 respondents had Software/Business/Web Development (SBWD) expertise either alone or in combination with other expertise.

Expertise's Category	Expertise Full Name
SBWD	Software/Business/Web Development
PPM	Project/Product Management
DRBA	Data/Requirements/Business Analysis
ASDPS	Agile / SD process / SCRUM
AD	Architecture / Designing
IPO	Integration/performance Optimization
QAT	Quality Assurance / Testing
DMBI	Data Mining / Business Intelligence
DevOps	DevOps
IC	IT Consultancy
Miscellaneous	GIS, Change/Enterprise management, Java tech, SAP, Info Security, Block chain, IT Governance

Table 8.2 Expertise's categories

Q.5: Are you involved in software development?

This question was asked to find out about the respondents; whether they were fully, partially or not involved in software development in their current roles. The objective in asking this question was to surface and collect feedback on the varieties of challenges faced in estimating effort – a person who was fully involved in development might had encounter different challenges or issues in estimation than the person who was partially involved in development but was at the same time responsible for other responsibilities in the project or organization. Moreover, individuals who were not involved in development in their current roles but still had experience in development and estimation could give a different perspective regarding the challenges that could arise in estimation. As the graph given at figure 8.5 shows 3 and 10 respondents who were involved in the software development (SD) in the partial and full roles respectively. Further, 8 respondents were not involved in SD in their current roles.

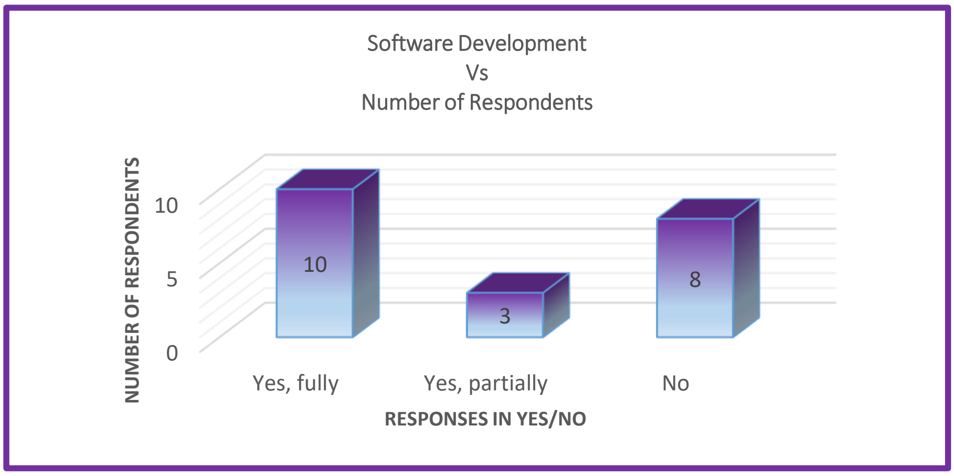


Figure 8.5 Software Develoment Vs Number of Respondents

8.4 Responses of the participants against Topic-2 Product/Project Development experience

Under this topic five questions were asked, to collect responses from the interviewees through which they could describe various aspects of their experience in the context of product/project development: the type of projects/products on which they were working and/or had worked, the development environment, i.e., global and/or local, common practices in their development activities, their experiences in estimating effort, size and/or scheduling, and common methodologies and challenges encountered in estimation. Details of each question along with the interviewees' responses are given in the subsequent sections.

Q.1: What kinds of projects/products does your company mainly develop e.g., Web, Desktop, mobile, built-in/embedded, games etc.?

The core objective in asking this question was to gain insights into the types of systems on which the interviewees were working or had previously worked. Further, it was assumed that exposure of the respondents to more varieties or types of systems might help them in completing one of the aspects considered in Topic-3 of the questionnaire where they were required to provide comparative ratings on the complexities of various system types with respect to their implementation. Respondents indeed revealed different types of systems/products on which they were working. All but two respondents communicated that their companies/organizations were dealing with the development of more than one project/product. Respondents' sharing types of systems/products were grouped into 11 categories, as shown in the graph given at Figure 8.6. This shows that the majority of the organizations (13 responses) were working on Web based systems while Desktop

(10 responses) and Mobile applications' (6 responses) development were placed at rank 2 and 3, respectively.

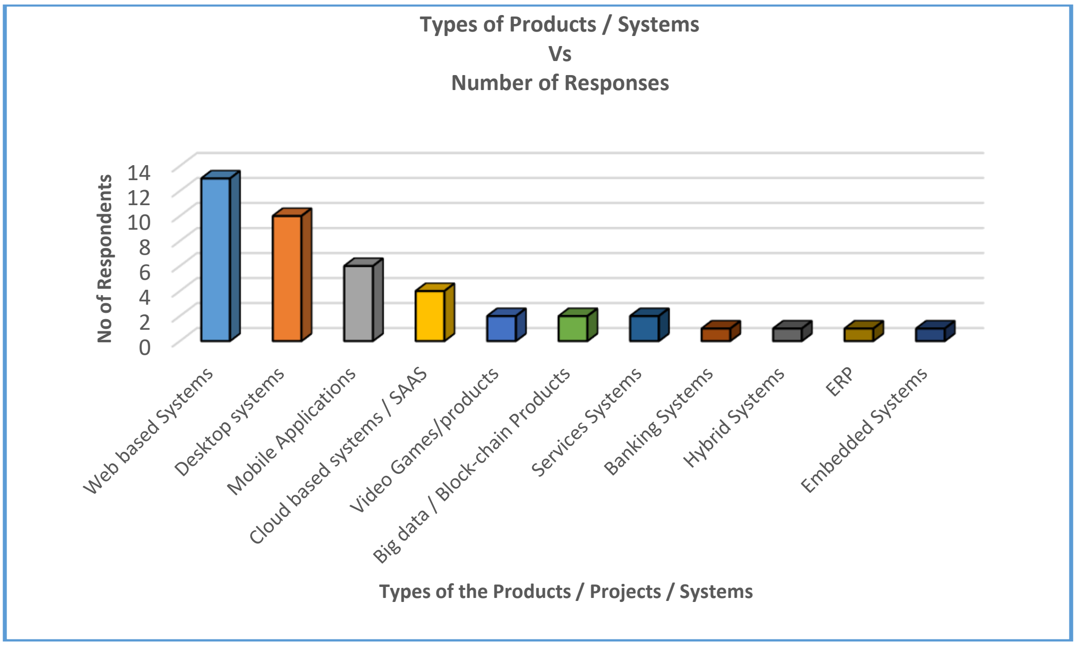
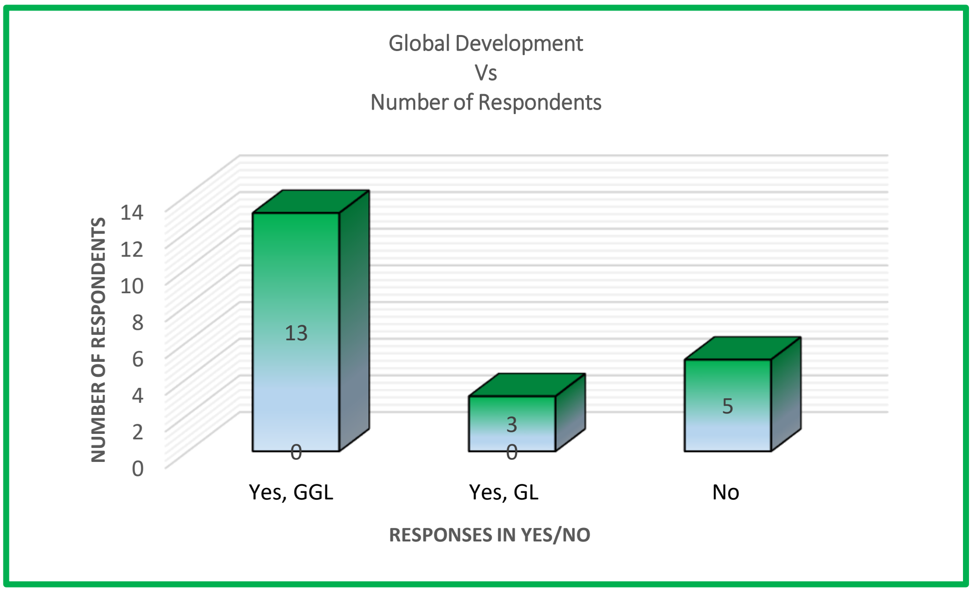


Figure 8.6 Type of Systems / Products Vs Number of Responses

Q.2: Does your project/product involve global development?

Offshore/global software development has become common due to various reasons, e.g., cheap offshore labour market, non-availability of trained or experienced human resources, and/or team placed at various geo-locations. This question was posed to find out how many interviewees were working in global development environments and based on their answers, in further intended questions their feedback collected on the challenges they observed or faced in estimating the effort of their projects/products because of the global development context and/or multi geo-located teams. The responses are shown by the bar graph as given at Figure 8.7, which demonstrates that 13 respondents were working and/or worked in global development contexts where their teams were geo-located in different countries, while 3 respondents' teams were geo-located in the same country.



GGL: Golbal Geo-located **GL**: Geo-loacted in the same country

Figure 8.7 Global Development Vs Number of Responses

Q.3: What practices do you regularly or often employ in software development?

In this question, the respondents were asked about their usage of business processes/practices in their project/product development. A list of the current practices and the responses against them are given in Table 8.3. The majority of the respondents (13) revealed the use of SCRUM as a single practice for the development of their projects/products. Eight respondents disclosed SCRUM was used along with other methods.

Current Practices in SD	No. of Respondents
Agile (SCRUM)	13
Agile (SCRUM) and other methods (Kanban, Xtreme Programming, SAFe, Waterfall, RAD, TDD, CDD, V-Model, Lean, and Prototyping)	8
SAFe: Scaled Agile Framework TDD: Test Driven Development	RAD: Rapid Application Development CDD: Component Driven Development

Table 8. 3 Current Software Development Practices Vs number of Responses

Q.4: Do you have any experience estimating the size, effort, cost or schedule of a software development project? If yes, then which methodology or methodologies have you used for the purposes of estimation?

This question was designed to draw out respondents' experience in estimating effort, cost or the scheduling of a development project/product, and which methodologies or practices they had used for the purpose of estimation. This was one of the core questions in the context of the intended study because it was assumed that if more respondents had experience in estimation, and further, the usage of its various methodologies, then they could be effective in explaining the actual challenges as faced by the IT industry in estimation generally and in the employment of specific techniques.

All respondents revealed that they had experience in estimating the effort, size, and/or the cost of software development projects/products. The techniques that they disclosed were being used for their project/product estimations are given in Table 8.4. These techniques were grouped into three categories based on the number of responses. Category-A techniques centred on planning poker were in use by the majority of the respondents (12 in number), while 7 respondents were employing various Category-B practices for the estimation of their development projects. Further, 3 respondents were using Category-C techniques, i.e., their own customized methods and/or no formal techniques for estimating their projects.

Category#	Methods / Techniques Used in Estimation	No. of Respondents
A	Planning Poker (Story Points based estimation) through expert judgement / Top-Down / Analogies approaches	12
B	Mixed methods of estimation (Parametric, T-shirt sizing, Kanban, EVM, CPI & SPI, Lean, SCRUM, Waterfall/SDLC)	7
C	Own customized or no formal estimation method through expert judgement / Top-Down / Analogies approaches	3
EVM: Earned Value Management CPI: Cost performance index SPI: Schedule performance index SDLC: Software development life cycle		

Table 8.4 Estimation Methods / Techniques Vs number of Responses

Q.5: Please describe any problems / challenges you might have experienced in estimating the size, effort, cost, or schedule of a software development project in which you were involved?

This was a supplementary question of previously asked question (Q.4) and designed to know the challenges or issues which respondents had experienced while they worked on their products. This was also one of the core questions in the context of the intended study because it was assumed that if more respondents had experience in estimation, then they could be effective in explaining the actual challenges / issues as faced by the IT industry in estimation generally and in the employment of specific techniques.

All respondents revealed that they had experienced numerous issues / challenges / problems while in estimating the effort, size, and/or the cost of software development projects/products. These problems / issues were grouped into 10 types based on the number of responses as presented in Table 8.5. Majority of the respondents were in the views that one of the core challenges of the estimation was the presence of new developer(s) in the team which impacted estimations' accuracy (due to many reasons e.g., lack of understanding for estimation activity, more time required to understand project, team, culture, environment and others), while 5 respondents indicated unclear requirements as one of the problems. Further, 3 respondents in each case were presented four challenges as indicated at S# 3, 4, 5, and 6 of Table 8.5 while estimating size of the project. Furthermore, 2 respondents unveiled that estimation could be impacted by various projects domains, lack of technological tools' expertise or experience and also the understanding issues for a story complexity. However, single respondent in each case had indicated 6 problems / issues during estimation. When the respondents were asked about the effort estimates' deviations as they observed for their completed and/or on-going projects, then 6 of them revealed 21-30%, 5 indicated 10-20% while 3 respondents conversed that the accuracy deviations were above 30%. Though to overcome the estimation inaccuracies, 4 respondents exhibited that they included 15-25% buffer or cushion time as over and above to their estimates for non-agile projects while for ASD projects the equivalent percentage of deviations were created deliberately in estimated story-points.

S#	Problems / Issues / Challenges	No. of Respondents
1	New Team Member / Developer	11
2	Requirements are unclear / not enough	5
3	Relationship of time with Story-points while estimating	3
4	HR availability for development tasks	3
5	Vertical Slicing of Stories especially for new Project	3
6	Different project's types / domains	3
7	Technological tools' expertise /experience	2
8	Understanding in story complexity	2
9	Integration Issues	2
10	Miscellaneous Issues: Global Development, Type of client-data especially for Business Intelligence Project, Estimator experience, Ambiguous done criteria, Priority of project, Involvement of non-developers for estimation	1

Table 8.5 Problems/Issues/Challnegs Vs number of Responses

8.5 Responses of the participants against Topic-3 Estimation / scheduling

The core intent of Topic-3 was to test and potentially validate the identified list of factors, their proposed descriptions, categories/subcategories, and scaling, based on the views of IT practitioners estimating size, effort, and cost of development projects and products in agile development contexts.

Q.1: Do you agree that the list of factors is adequate and significant in the context of estimating effort, size, and/or cost of projects/products in agile development contexts? If not, then state which factors(s) are missing and/or not significant, with some justifications.

A list of 10 factors was shared with the respondents to ask their perceptions about their significance in the context of effort estimation and the possibility that they might impact accuracy with respect to the actual effort. All 21 respondents agreed upon their significance in relation to effort estimation at a general level and their potential to impact on accuracy. No other additional factors were suggested for this purpose.

Q.2: Do you agree with the proposed description, categories / sub-categories and
scaling of the factors?

Almost all respondents agreed with the descriptions, categories and scaling. A few respondents suggested including more subcategories in the existing subclasses of the system types, working environment, and culture factors. They were of the opinion that the given subcategories (e.g., best & worst working environment and open & closed culture, respectively) could only cater for extreme scenarios of the given factors, whereas more subcategories were required to address in-between situations. Further, two respondents were proposed to include IoT, Data analytics, and Machine Learning as additional subcategories of the system's types.

Q.3: Provide ratings against each of the given or proposed factors?

To secure responses from the participants, detailed views of all factors were provided in the interview guide, attached at Annex-K. In the guide, each factor was elaborated based on the following perspectives:

a. Description of the factor: A detailed concept description of each factor was provided, showing its significance in the context of estimating effort, and its perceived impact on accuracy in relation to the actual effort, size, and cost of the project / product.

b. Subcategories of the factor along with their descriptions: Each factor was divided into further subclasses to measure their level of significance in terms of estimating effort and its impact on accuracy with respect to the actual effort, size, and cost of the project/product; e.g., in the expertise factor, the level of expertise was divided into four further subclasses of High, Medium, Adequate, and Inadequate expertise. Each subcategory had a full description to show its relevance. The proposed subclasses of all the factors along with their level of significance are given in the table.

c. Fuzzy scales of five values from Very High to Very Low (i.e., Very High, High, Moderate, Low, Very Low): Scaling / rating options were provided against the subcategories of a factor so that respondents could choose the scaling and ratings of their choice. One additional "Don't know", option was also provided so that respondents could choose this option if they were unaware of the significance or impact of a specific factor(s) or its subclasses.

   The meaning of the scales for each factor was different, e.g., in the expertise factor, if the respondent marked the rating "Very High" against the subclass "High Expertise", it meant a team member with "High Expertise" could estimate effort very close to the actual estimations of effort, while rating "High" meant estimations close to the actual estimates but less than “Very High” and so on. For the factor of team relationship and spirit, subclass “Excellent” of the said factor demonstrated very high effectiveness of the team, if a respondent selected rating “Very High” whereas rating “High” meant team effectiveness was still high but less than “Very High” and so on.

8.6 Ratings of the respondents

In this section, the ratings of the respondents for all of the individual factor(s) as obtained from the interviews are reported using graphs and tables. Any additional comments of the respondents are included in the table following the graph. Moreover, the most frequently used terminologies in this section are explained here for more clarity:

i. Actual Effort: The effort required to implement a software development task by the team.
ii. Estimated Effort: The projected/estimated effort required to implement a software development task by the team.
iii. Impact on estimation: This is an effect which may influence estimation negatively or positively due to which the projected estimations may be accurate, close or deviated from the actual effort.

8.6.1 Respondents’ Rating for all factors

Respondents’ ratings against all factors, i.e., Team members’ expertise, experience, and relationships, Stakeholder coordination, Organization type, culture, and work environment, System context, Sprint and Risk contexts, are shown by the bar graph given in Figure 8.8, demonstrating the perceived impacts of the identified factors on projected/estimated effort.

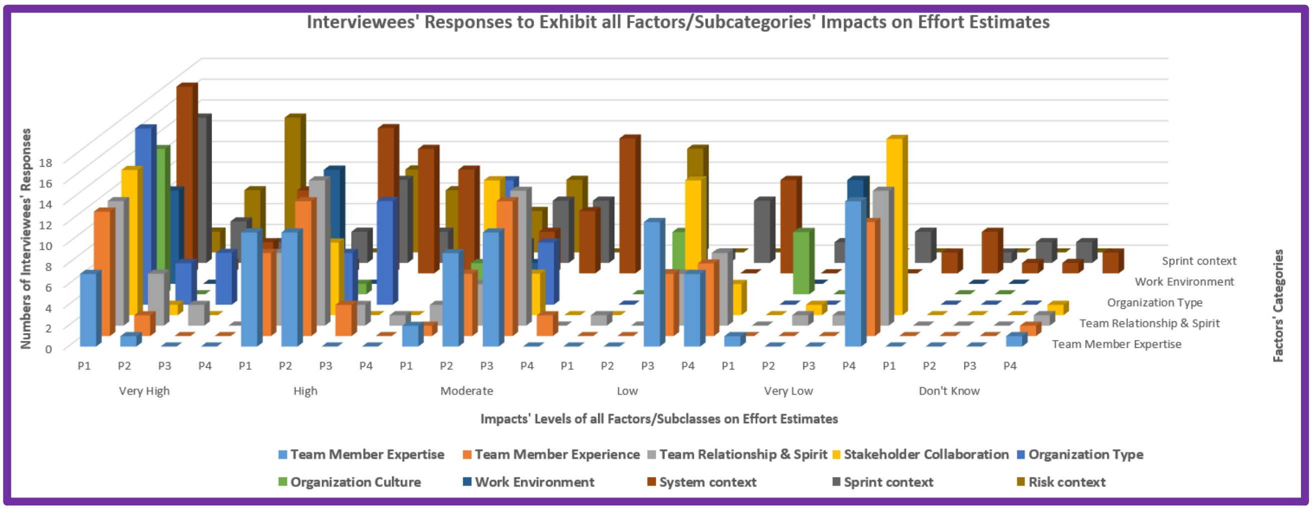


Figure 8.8 Interviewees' Responses to Exhibit all Factors/Subclasses's Impacts on effort Effort Estimates

The y-axis shows the number of responses while the x-axis denotes ratings (i.e., Very High, High, Moderate, Low, Very Low) in each subclass of the factors (i.e., P1, P2, P3, and P4) where P1 represents the highest and P4 the lowest subcategory of the specific factor. (Note that System context has five subcategories but the last two subcategories, i.e., P4, and P5, are combined just to create symmetry with other factors, meaning that all factors along with their subcategories can be exhibited in a single graph as presented in Figure 8.8.) The respondents' ratings for the individual factors are given in the subsequent sections. Moreover, to describe the responses of the interviewees, the acronyms VH, H, M, L, VL are used to represent different grades' levels as "Very High", "High", "Moderate", "Low" and "Very Low" respectively.

8.6.2 Respondents' Ratings for individual factors

a. Team members' expertise

Respondents were required to rate this factor based on the level of expertise in the team members and its impact on the anticipated effort accuracy. High, Medium, Adequate, and Inadequate are the proposed expertise levels, presented in the graph as P1, P2, P3, and P4 respectively. Their complete descriptions are provided in Chapter 4. The responses are shown in the graph as given in Figure 8.9. The different coloured bars represent the different levels of expertise within one scale, e.g., P1 in scales of "Very High" and "High" are in different colours. On the y-axis of the graph the numbers of responses are given while ratings / scales are shown on the x-axis. Response-wise ratings and their explanations are given below:

- Respondents rated P1 as per the given grade-response matrix: VH-7, H-11, M-2, L-0, and VL-1. The majority of respondents (19/21) were of the view that if the team had members with "High expertise" then their estimated effort's accuracy could be from "VH to H" and likely to be very close to the actual effort.

- Respondents rated P2 as per the given grade-response matrix: VH-1, H-11, M-9, L-0, and VL-0. The majority of respondents (20/21) were in the opinion that if the team had members with “Adequate expertise” then their estimated effort’s accuracy could be from “H to M” and likely to be close to the actual effort.
- 19 respondents rated P3 as per the given grade-response matrix: VH-0, H-0, M-9, L-10, and VL-0 while two respondents graded P3 from “Moderate to Low”. This reveals that almost 12 respondents were of the view that if the team members had the “Medium expertise” then their estimated effort’s accuracy could be low while 11 respondents said moderate as compared to the actual effort.
- 19 respondents rated P4 as per the given grade-response matrix: VH-0, H-0, M-0, L-6, and VL-13 while one respondent graded P4 from “Low to Very Low”. This discloses that almost 14 respondents were of the opinion that if the team members had the “Inadequate expertise” then their estimated effort’s accuracy might be “VL” than the actual effort.
- 1 respondent rated P4 “Don’t Know” which means that if the team had members with “Inadequate expertise” then their estimated effort’s accuracy could not be known by the respondent.

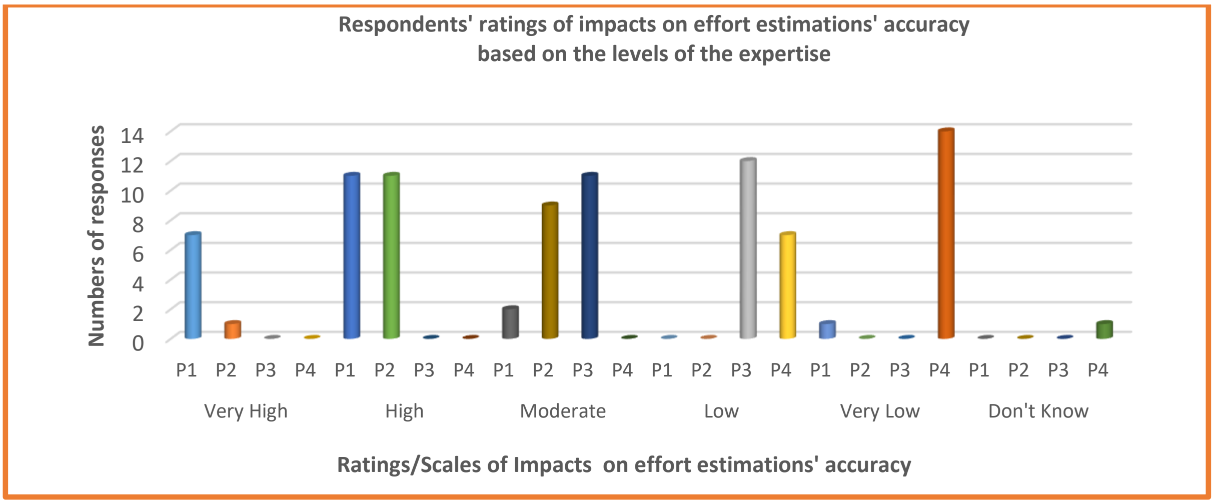


Figure 8.9 Respondents’ ratings of impacts on effort estimates based on the expertise levels

b. Team member’s Experience

Respondents were required to rate this factor based on the level of experience in the team members and its impact on the anticipated effort’s accuracy. Proficient, Experienced, Intermediate, and Beginner are the proposed experience levels and are presented in the graph as P1, P2, P3, and P4 respectively. Their complete descriptions are provided in Chapter 2. The responses are shown in the graph as given in Figure 8.10. Response-wise ratings and their explanations are given below:

- Respondents rated P1 as per the given grade-response matrix: VH-12, H-8, M-1, L-0, and VL-0. The majority of respondents (20/21) were of the view that if the team had members with “Proficient” experience then their estimated effort’s accuracy could be from “VH to H” and likely to be very close to the actual effort.

- Respondents rated P2 as per the given grade-response matrix: VH-2, H-13, M-6, L-0, and VL-0. The majority of respondents (19/21) had the opinion that if the team members were in the “Experienced” category then their estimated effort’s accuracy could be from “H to M” and likely to be close to the actual effort.
- 20 respondents rated P3 as per the given grade-response matrix: VH-0, H-3, M-12, L-5, and VL-0 while one respondent graded P3 from “Moderate to Low”. This reveals that almost 13 respondents were of the view that if the team members had the “Intermediate” experience then their estimated effort’s accuracy could be moderate while almost 6 respondents said low as compared to the actual effort.
- 20 respondents rated P4 as per the given grade-response matrix: VH-0, H-0, M-2, L-7, and VL-11. This reveals that 11 respondents had the opinion that if the team members were “Beginner” in their experiences then the estimated effort’s accuracy could be very low while 7 respondents said low as compared to the actual effort.
- 1 respondent rated P4 “Don’t Know” which means that if the team had members with “Beginner experiences” then their estimated effort’s accuracy could not be known by the respondent.

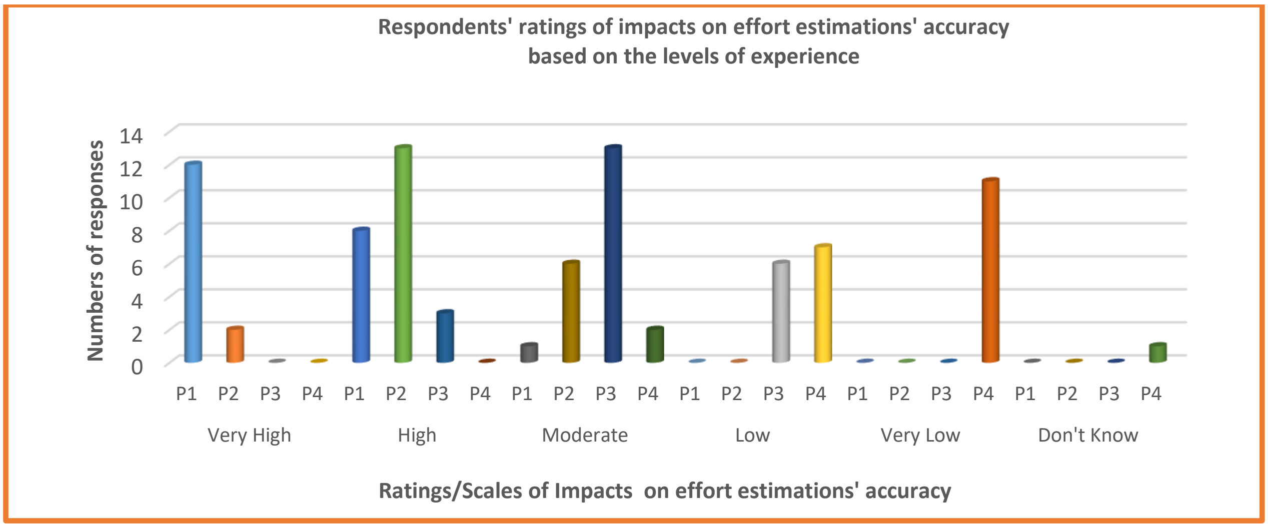


Figure 8.10 Respondents’ ratings of impacts on effort estimates based on the experience levels

c. Team relationship and spirit

Respondents were required to rate this factor based on the level of coherence and effectiveness in the team and its impact on the anticipated effort’s accuracy. Excellent, Very Good, Good, and Poor or conflicting are the proposed levels for team effectiveness and are presented in the graph given at Figure 8.11 as P1, P2, P3, and P4, respectively. Response-wise ratings and their explanations are given below:

- 20 respondents rated P1 as per the given grade-response matrix: VH-12, H-7, M-1, L-0, and VL-0 while one respondent graded P1 from "Moderate to Low". This reveals that if the team effectiveness is "Excellent" then their estimated effort's accuracy could be from "VH to H" and likely to be very close to the actual effort. This is because of the very high synergy level within the team which assists the team to accomplish the projected tasks at their earliest.
- 19 respondents rated P2 as per the given grade-response matrix: VH-5, H-11, M-2, L-0, and VL-1 while two respondents graded P2 from "High to Moderate". This discloses that if the team effectiveness is "Very Good" then the estimated effort accuracy could be high and likely to be close to the actual effort.
- 19 respondents rated P3 as per the given grade-response matrix: VH-2, H-2, M-11, L-3, and VL-1 while two respondents graded P3 from "Moderate to Low". This discloses that if the team effectiveness is "Good" then its estimated effort's accuracy could be "Moderate" and likely to be moderately close to the actual effort.
- 19 respondents rated P4 as per the given grade-response matrix: VH-0, H-1, M-0, L-6, and VL-12 while one respondent graded P4 from "Low to Very Low". This reveals that if the team effectiveness is "Poor or conflicting" then its estimated effort's accuracy could be "Low to Very Low" as compared to the actual effort. This is because of the low synergy level of the team.
- Only 1 respondent rated P4 "Don't Know" which means that if the team effectiveness is "Poor or conflicting" then its estimated effort's accuracy is not known by the respondent.

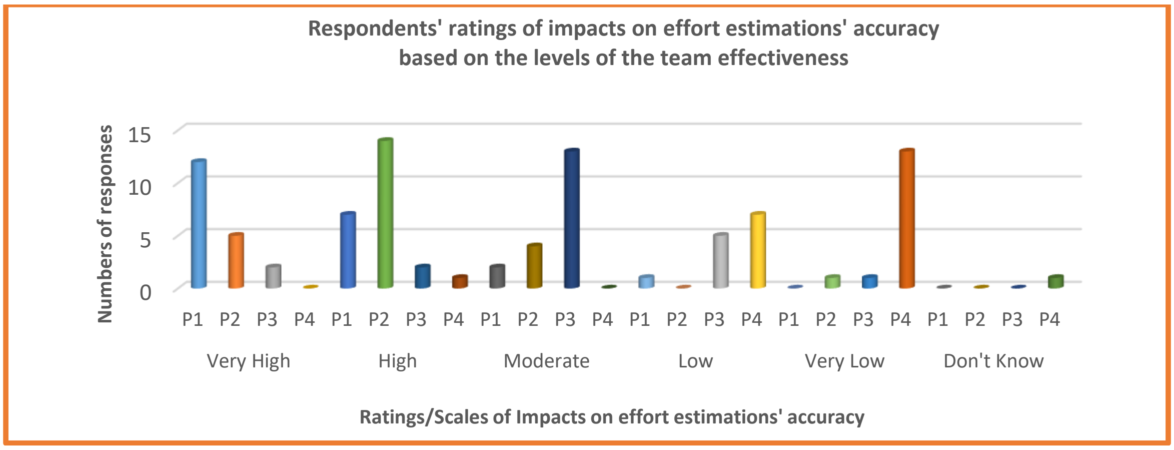


Figure 8.11 Respondents' ratings of impacts on effort estimates based on the team effectiveness levels

d. User / Stakeholder Collaboration

Respondents were required to rate this factor based on the level of user/stakeholder collaboration and its impact on the anticipated effort's accuracy. Fully, Partially, Casually, and Poorly are the proposed levels for user/stakeholder collaboration, presented in the graph given at Figure 8.12 as P1, P2, P3, and P4, respectively. Response-wise ratings and their explanations are given below:

- Respondents rated P1 as per the given grade-response matrix: VH-14, H-7, M-0, L-0, and VL-0. This demonstrates that if the users/stakeholders are “Fully” collaborative then the estimated effort accuracy of the team could be very high, due to the timely feedback and active participation of the user/stakeholder in the various development activities and meetings of the product. Therefore, it is more likely that the tasks of the team are completed within the anticipated effort and hence its accuracy could be very close to the actual effort.
- 20 respondents rated P2 as per the given grade-response matrix: VH-1, H-6, M-12, L-0, and VL-1 while one respondent graded P2 from “High to Moderate”. This reveals that if the users/stakeholders are “Partially” collaborative then the accuracy of the team’s estimated effort could be “Higher to Moderate” as compared to the actual effort due to partially feedback/participation of the stakeholders/users in the product’s endeavours.
- 20 respondents rated P3 as per the given grade-response matrix: VH-0, H-0, M-3, L-12, and VL-5 while one respondent graded P3 from “Moderate to Low”. This demonstrates that if the users/stakeholders are “Occasionally” collaborative then the estimated effort’s accuracy of the team could be “Low to very Low” than the actual effort due to the users’ occasional feedback and participation in the product’s endeavours.
- 20 respondents rated P4 as per the given grade-response matrix: VH-0, H-0, M-0, L-3, and VL-17. This reveals that if the user/stakeholder “Poorly” collaborates then the estimated effort’s accuracy of the team could be very low as compared to actual effort, due to the poor participation in different activities of the product.
- Only 1 respondent rated P4 “Don’t Know”, meaning that if the user/stakeholder collaborates “Poorly” then the estimated effort’s accuracy of the team is not known by the respondent.

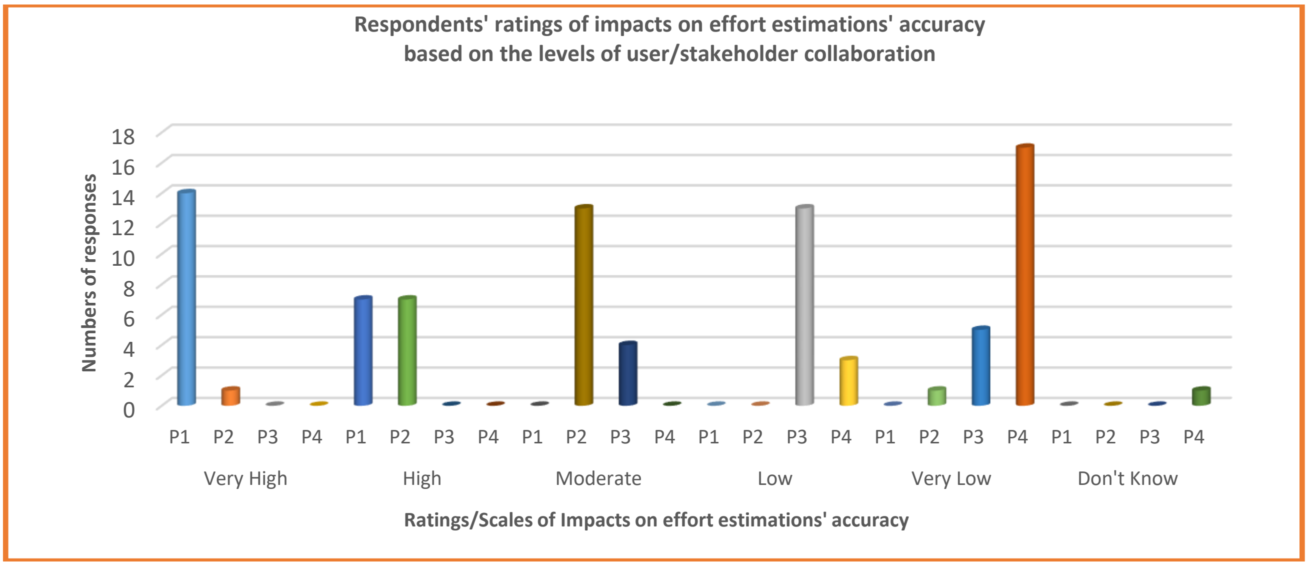


Figure 8.12 Respondents' ratings of impacts on effort estimates based on the coord levels of stakeholders

e. Organization Type

Respondents were required to rate this factor based on organizations types and their impact on anticipated effort accuracy. Govt/Public, Private, and Joint (i.e., a mix of Public and Private) are the proposed types of organization, presented in the graph given at Figure 8.13 as P1, P2, and P3, respectively. As with all factors considered in these interviews their complete descriptions are provided in Chapter 2. Response-wise ratings and their explanations are given below:

- 20 respondents rated P1 as per the given grade-response matrix: VH-16, H-4, M-0, L-0, and VL-0 while one respondent graded P1 from "Very High to High". This reveals that Govt/Public sector type of organization has very high impact (normally negative) on the accuracy of the team's estimated effort due to lengthy bureaucratic channels, project size, political interference, and others in government/ public sector organization. Therefore, "Very High to High" level of deviations are expected in the estimated effort of the team as compared to the actual effort.
- 20 respondents rated P2 as per the given grade-response matrix: VH-4, H-3, M-10, L-3, and VL-0 while two respondents graded P2 from "Moderate to High". This demonstrates that the Joint sector type of organization has "Moderate to High" impact (normally positive) on the estimated effort accuracy due to joint nature of the organization (i.e., a mix of Govt/Public and Private). Therefore, this organization type has less influence on the anticipated effort of the team for the completion of their projects/products. Because of this, some deviations are expected in the estimated effort of the team as compared to the actual effort.
- 19 respondents rated P3 as per the given grade-response matrix: VH-4, H-9, M-5, L-1, and VL-0 while one respondent graded P3 from "Very High to Moderate" and other from "High to Moderate". This reveals that the Private type of organization has "High to

Moderate" impact (normally optimistic) on the estimated effort accuracy due to the private nature of the organization (where no lengthy bureaucratic channels, small project size, no political interference, and others are expected). Because of this, little deviations are likely in the estimated effort of the team as compared to the actual effort.

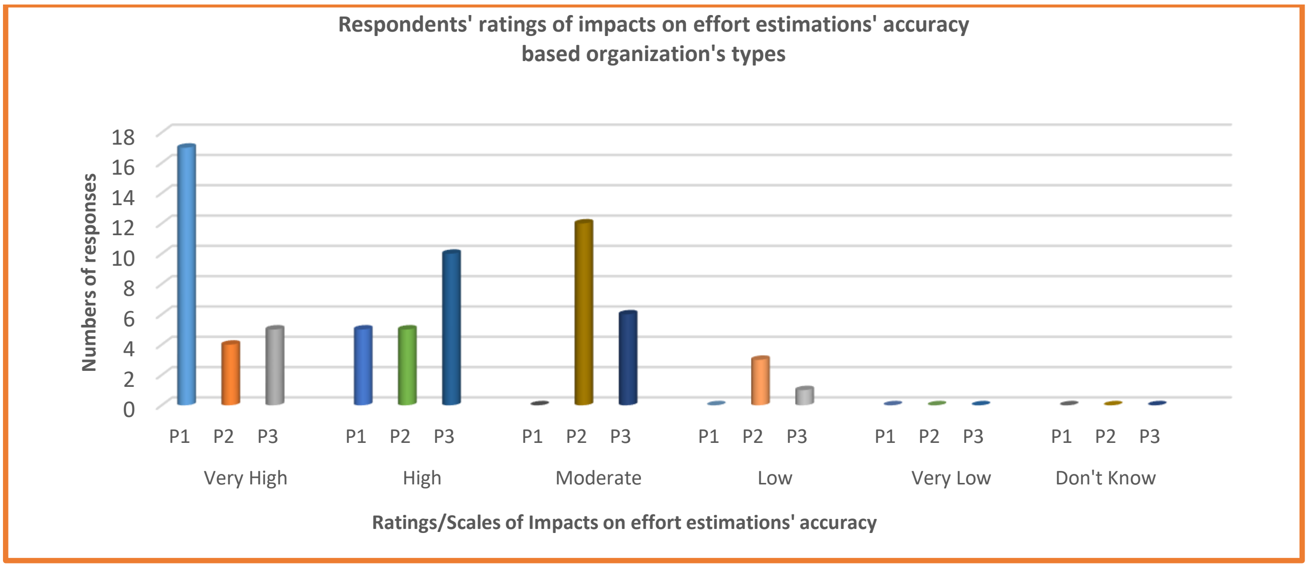


Figure 8.13 Respondents' ratings of impacts on effort estimates based on the organizations' types

f. Organization's Culture

Respondents were required to rate this factor based on the impact of the culture of organizations on estimated effort accuracy. Open and Robust, Fragile and Closed are the proposed cultures, presented in the graph given at Figure 8.14 as P1 and P2, respectively. Response-wise ratings and their explanations are given below:

- 19 respondents rated P1 as per the given grade-response matrix: VH-12, H-4, M-3, L-0, and VL-0 while two respondents graded P1 from "Very High to High". The majority of the respondents (18/21) were of the opinion that the Open & Robust culture had a positive impact on the anticipated effort estimations. Therefore, it was more likely that the team might finish their intended tasks within the anticipated effort and consequently could achieve "High to Very High" level of estimation's accuracy as compared to the actual effort.
- 20 respondents rated P1 as per the given grade-response matrix: VH-0, H-1, M-0, L-5, and VL-14 while one respondent graded P2 from "Low to Very Low". Respondents (20/21) were of the opinion that the Fragile & Closed culture had an adverse impact on the accuracy of effort estimation. Therefore, it was more likely that the team could not be able to finish their intended tasks within the anticipated effort and hence might achieve "Low to Very Low" level of estimation's accuracy.

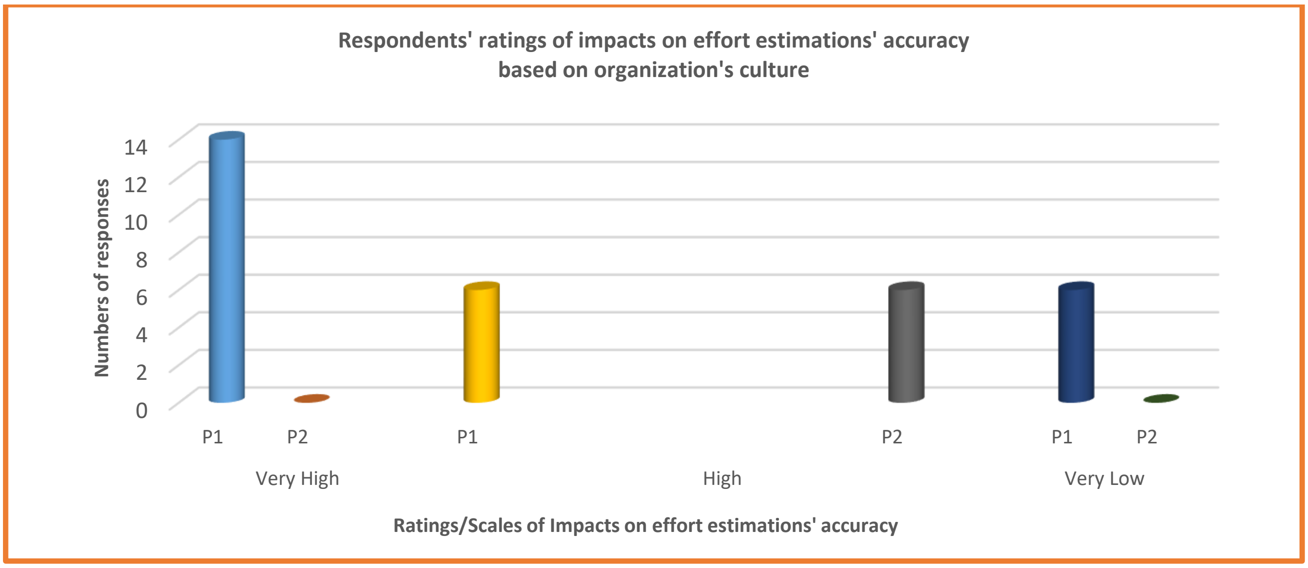


Figure 8.14 Respondents' ratings of impacts on effort estimates based on the organizations' culture

g. Organization's Work Environment

Respondents were required to rate this factor based on the working environments of organizations and their impacts on anticipated effort accuracy. Quite simply, Best and Worst are the proposed work environments, presented in the graph given at Figure 8.15 as P1 and P2, respectively. Response-wise ratings and their explanations are given below:

- 20 respondents rated P1 as per the given grade-response matrix: VH-9, H-10, M-1, L-0, and VL-0 while one respondent graded P1 from "High to Moderate". Almost all of the respondents (20/21) were of the opinion that the best working environment had an inspiring impact on the anticipated effort's estimations. Therefore, "High to Very High" estimation's accuracy could be expected when compared to the actual effort.
- 20 respondents rated P1 as per the given grade-response matrix: VH-0, H-0, M-2, L-9, and VL-9 while one respondent graded P2 from "Low to Very Low". The majority of the respondents (19/21) were of the opinion that the Worst working environment had an undesirable impact on the effort estimation accuracy. Therefore, "Low to Very Low" estimation's accuracy could be expected as compared to the actual effort.

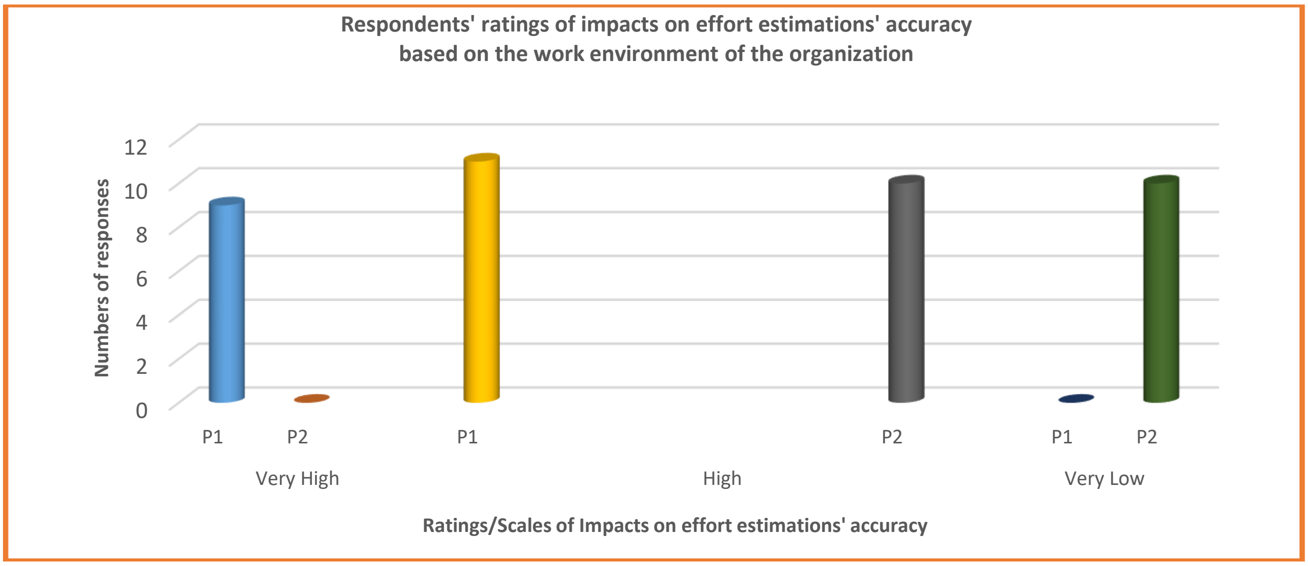


Figure 8.15 Respondents' ratings of impacts on effort estimates based on the organizations' work Environment

h. Sprint context

Respondents were required to rate this factor based on the complexity of the Sprints/Cycles and their impacts on the anticipated effort. Sprint-1, -2 and Sprint-n are the proposed sprints/cycles of a product/project, presented in the graph given at Figure 8.16 as P1, P2, and P3, respectively. Response wise-ratings and their explanations are given below:

- 18 respondents rated P1 as per the given grade-response matrix: VH-12, H-1, M-1, L-2, and VL-2 while two respondents graded P1 from "Very High to High". The majority of the respondents (14/21) were of the opinion that the first sprint/cycle of a product/project had a very high level of complexity and therefore required more effort as compared to the succeeding sprints, due to factors such as the level of uncertainty, longer time consumption by the team in initial learning and understanding of the user requirements, and/or also in developing the first blueprint of the basic architecture of the system. Therefore, more deviations were expected in the anticipated effort's accuracy of the team in this sprint as compared to in successive sprints.
- 18 respondents rated P2 as per the given grade-response matrix: VH-4, H-7, M-5, L-1, and VL-1 while one respondent graded P2 from "High to Moderate". A simple majority of the respondents (12/21) held the view that sprint-2 had a complexity level from "High to Moderate". Therefore, fewer deviations were expected in the anticipated effort accuracy of the team in sprint-2 as compared to in successive sprints.
- 17 respondents rated P3 as per the given grade-response matrix: VH-2, H-3, M-6, L-5, and VL-2 while one respondent graded P3 from "Low to Very Low". A bare majority of the respondents (12/21) were of the opinion that sprint-n had the complexity level of "Medium to Low". Therefore, fewer deviations were expected in the anticipated effort's accuracy of the team in sprint-n as compared to in successive sprints and it was likely

that the projected effort estimations of the team could be close to the actual estimation of the effort.

- 1, 2 and 3 respondents were unsure about the complexity of sprint 1, 2, and *n* respectively. Therefore, they selected the “Don’t Know” option while responding in relation to the sprint factor.

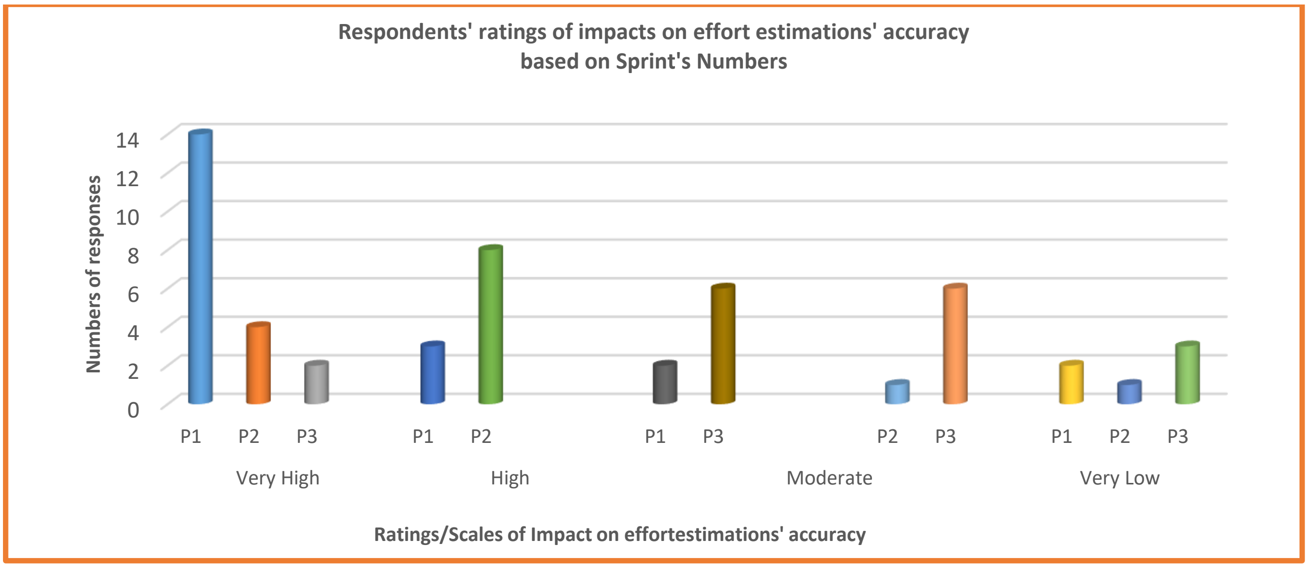


Figure 8.16 Respondents’ ratings of impacts on effort estimates based on the Sprint’s numbers

i. Risk context

Respondents were required to rate this factor based on the types of the risks that may influence the success of the project/product and hence impact the anticipated effort’s accuracy. Known mitigated, known non-mitigated, and Unknown are the proposed risks on a project/product, presented in the graph given at Figure 8.17 as P1, P2, and P3, respectively. Their complete descriptions are provided in Chapter 2. Response-wise ratings and their explanations are given below:

- Respondents rated P1 as per the given grade-response matrix: VH-2, H-0, M-4, L-10, VL-7. The vast majority of the respondents (17/21) were of the opinion that, in the case of known mitigated risk, there will be “Low” (10 responses) to “Very Low” (7 responses) chance of an adverse impact on the success of the project/product. Therefore, few deviations are expected in the anticipated effort accuracy, and it is likely that the projected effort estimations of the team will be very close to the actual effort expended.
- Respondents rated P2 as per the following grade-response matrix: VH-6, H-8, M-7, L-0, VL-0. The majority of respondents (14/21) are of the opinion that, in the case of known unmitigated risk, there will be a “High” (8 responses) to “Very High” (6 responses) chance of a detrimental impact on the success of the project/product. Moreover, 7 respondents

were of the view that this type of risk will have a moderate level of negative impact on the accomplishment of the project.

- Respondents rated P3 as per the given grade-response matrix: VH-13, H-6, M-0, L-0, VL-0. Nearly all of the respondents (19/21) were of the opinion that, in the case of unknown risk, there will be a "High" (6 responses) to "Very High" (13 responses) chance of a detrimental impact on the success of the project/product. Sizeable deviations are expected between the anticipated effort of the team and the actual effort incurred.

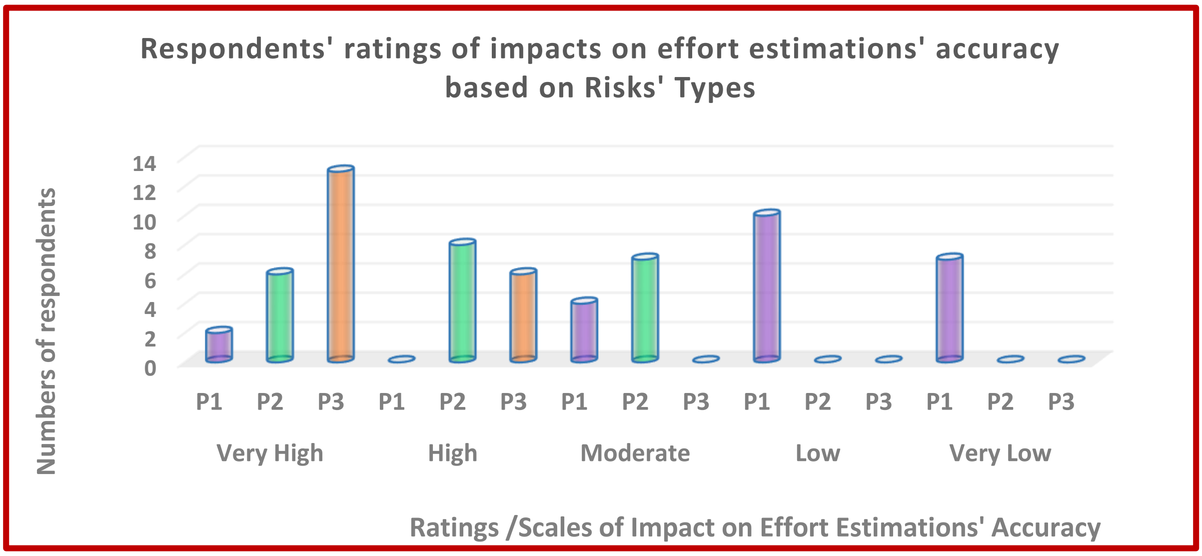


Figure 8.17 Respondents' ratings of impacts on effort estimates based on Risks types

j. System Complexity

Respondents were required to rate this factor based on the impact of the Systems' types and complexities (with respect to their implementation) on effort estimation accuracy. Defence/Military/Mission Critical, Embedded, DSS/EIS/Enterprise, Financial/Banking, and General Purpose (other than financial) are the proposed system types, presented in the graph given at Figure 8.18 as P1, P2, P3, P4, and P5, respectively. Response-wise ratings and their explanations are given below:

- Respondents rated P1 as per the given grade-response matrix: VH-18, H-1, M-2, L-0, VL-0. The majority of them (18/21) were of the opinion that Defence/Military/Mission Critical systems have very high levels of complexity and therefore, require very high effort to implement as compared to other types of systems. Consequently, the team may expect very high levels of deviation in its anticipated effort's accuracy as compared to the actual effort.
- Respondents rated P2 as per the given grade-response matrix: VH-1, H-14, M-4, L-0, VL-0. A majority (14 in number) were of the view that the Embedded system type is

relatively complex and requires a high amount of effort to implement as compared to other types of system. Therefore, a high level of deviation is expected in the team's anticipated effort accuracy.

- Respondents rated P3 as per the given grade-response matrix: VH-3, H-12, M-6, L-0, VL-0. Many of the respondents (18 in number) were of the opinion that the DSS/EIS/Enterprise system type has moderate (6 responses) to high (12 responses) complexity and, as a result, require moderate to high amounts of effort to implement as compared to other types of system. Hence, a team may expect moderate to high deviations in its anticipated effort accuracy when compared to the actual effort.
- Respondents rated P4 as per the given grade-response matrix: VH-8, H-9, M-3, L-1, VL-0. The respondents (20/21) had mixed opinions, indicating that Financial/Banking Systems are very high (8 responses), high (9 responses), and/or moderate (3 responses) in their complexities. Therefore, these types of system require very high to moderate amounts of effort to implement as compared to the other types of systems.
- Respondents rated P5 as per the given grade-response matrix: VH-0, H-1, M-10, L-8, VL-2. They (18 in number) were of the opinion that the General-Purpose system type are is of moderate (10 responses) to low (8 responses) complexity, and hence require a moderate to low amount of effort to implement as compared to the other types of systems. Accordingly, the team may expect deviations from moderate to low levels in its anticipated effort accuracy.
- 4 respondents were unsure about the implementation complexity level of the DSS/EIS/Enterprise system type while one was uncertain in each type of system, therefore, they selected the "Don't Know" option when responding regarding the impact of the system complexity factor.

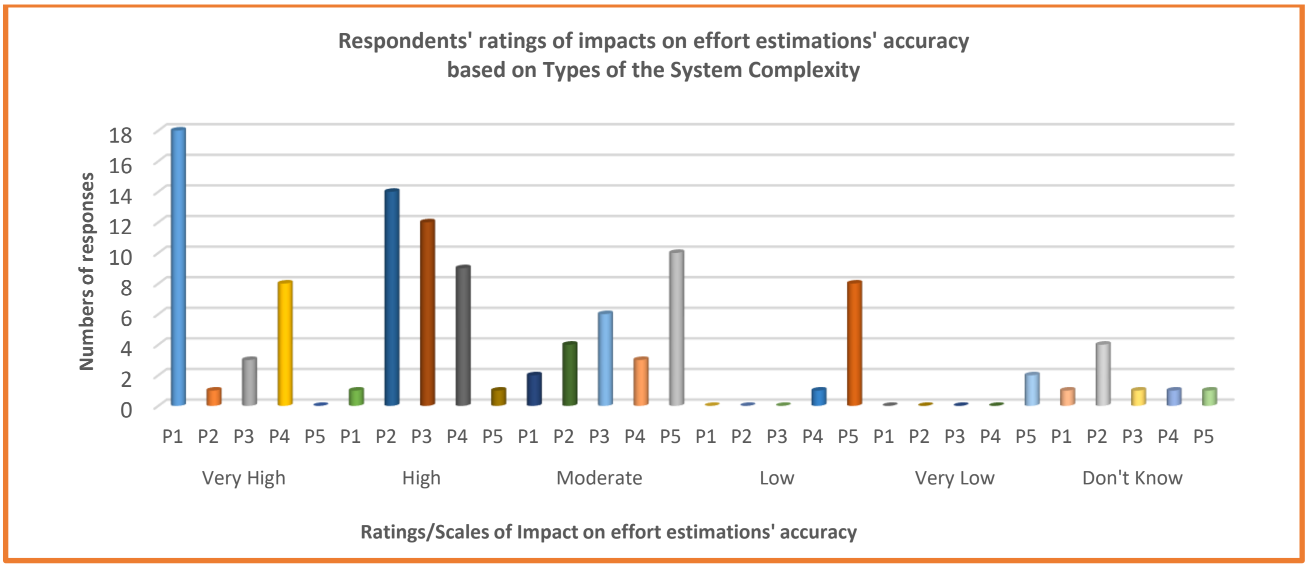


Figure 8.18 Respondents' ratings of impacts on effort estimates based on the System types' complexity

8.7 Chapter Review

In this chapter, design and execution of the instrument used for the interviews of agile developers, and their responses against the three intended topics (i.e., position, product development and estimation/scheduling) were discussed. Moreover, interviewees' responses with supporting graphic/tabular depictions and demographic details were also presented.

In next chapter, experimental data preparation, descriptive statistics, analysis, normality testing, and hypothesis evaluations are demonstrated. Further, various treatments, and effort estimation analysis are also provided.

Chapter-9 Data Preparation and Analysis

Contents' Summary: *This chapter explains the experiment data preparation, the selection of descriptive statistics and the inferential data analysis for the six treatments employed on three industrial projects. Consideration is given to the normality of data distributions, and testing of hypotheses using both parametric and non-parametric tests. At the end of this chapter various comparisons are presented among the treatments' estimated and actual finished story-points, effort estimations, and actual effort based on single story-points. Furthermore, Balanced Relative Error (BRE) and Balanced Relative Error Bias (BREbias) metrics are reported to analyse over- and under-estimation for the applied treatments.*

Descriptive statistics are an effective means through which to characterise various aspects of research data, which mainly incudes the central tendency (i.e., the Mean, Median, and Mode) and/or variability/dispersion (e.g., standard deviation, inter-quartile range – indicators of how much the individual recorded/observed values are similar or different) (Vetter, 2017). In this study, data was collected during two activities: interviewing the 21 IT experts and at the time of conducting the planned experiments. The interview data collection, preparation and analysis were explained in Chapter 8 while the experimentation details along with the data recordings were reported in Chapter 7. This chapter describes the experimental data quantitatively with the help of descriptive statistics supported by text descriptions and various graphs/tables. Further, the hypotheses (i.e., null and alternate) which were formulated in Chapter 6 are discussed and evaluated here.

9.1 Experiment Data Preparation

The experiments were conducted in the real-world environment of an IT company that was itself a subsidiary of an Australian-based international company dealing in building products and construction. As per AUT ethics approval, the actual name of this company could not be disclosed. Therefore, the fictitious name AUSIT was used to refer to the experiment host organization. AUSIT was involved in the development of software projects ranging from Mobile/Web applications to ERP solutions for their clients. AUSIT followed an agile/scrum framework for the development of their IT-related projects. The proposed treatments were applied on three real AUSIT projects (i.e., the objects of the experiment), which were named as Xproject, Yproject, and Zproject throughout this experimentation. These projects were developed by two different AUSIT development teams (i.e., the subjects of the experiment) who were referred to in this experiment as Team-A and Team-B. Further details of the teams, treatments, and projects are given in Chapter 7. The data collected through the experiments were shared on the Google/Excel sheets with the researcher by AUSIT. The data items for all of the treatments are shown in Table 9.1. This table exhibits samples, sprint numbers, number of stories per sprint, estimated story-points, person-hours, finished and unfinished story-points for all treatments.

S#	SPT #	PHs	T1 TESPs	T2a (PERT-1) TESPs	T2b (PERT-2) TESPs	T3 TESPs	T4a (PERT-1) TESPs	T4b (PERT-2) TESPs	T5 TESPs	T6 TESPs	AFSPs
Experimental Data: (Proposed subjects are Team-A and Team-B, Agile coach / Capability Lead, and the Researcher, while objects are Xproject, Yproject & Zproject)											
Xproject Data: - Subjects/Treatments: Team-A, Agile coach, Researcher / T1, T2, T5											
1	27		37	35	34	-	-	-	33	-	32
2	28		45	46	45	-	-	-	42	-	42
3	29		49	42	41	-	-	-	45	-	44
4	30	700	78	71	65	-	-	-	71	-	57
5	31		77	67	58	-	-	-	70	-	77
6	32		62	55	48	-	-	-	57	-	49
Yproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / T1, T2, T3, T4, T5, T6											
7	1		39	29	23	-	-	-	31	-	31
8	2		-	-	-	34	38	33	-	29	31
9	3		47	35	31	-	-	-	42	-	39
10	4	350	-	-	-	42	45	40	-	37	37
11	5		39	31	26	-	-	-	34	-	39
12	6		-	-	-	32	31	27	-	29	32
Zproject Data: - Subjects/Treatments: Team-B, Agile coach, Researcher / T1, T2, T3, T4, T5, T6											
13	1		-	-		34	30	27	-	26	31
14	2		42	38	33	-	-	-	37	-	37
15	3		-	-	-	45	37	31	-	42	40
16	4	350	52	45	40	-	-	-	47	-	44
17	5		37	30	27	-	-	-	34	-	34
18	6		-	-	-	32	31	27	-	29	32

S#: Sample number **SPT#:** Sprint number **US:** User stories **T1, T2, T3, T4, T5, T6:** Treatments 1, 2, 3, 4, 5, 6 **TESPs:** Total Estimated Story-Points per sprint
AFSPs: Actual Finished SPs **PERT-1 & 2:** Variant-1 & 2 of PERT **PHs:** Person-Hours

Table 9.1 PHs and TESPs of the Experimental Data of AUSIT's Xproject, Yproject & Zproject

To use these data items further in the analysis and hypothesis evaluations, further data points are added to account for estimated/actual effort per story-point in all the sprints against all projects/treatments, and the modified data are shown in the Table 9.2. The estimated/actual effort per story-point is calculated by dividing the ideal Person-hours by the total estimated/finished story-points in a sprint. The ideal Person-hours (PHs) are the product of the number of team members (who actually worked in a sprint), the duration of the sprint (i.e., 10 days in this case) and the ideal hours per day. AUSIT conveyed to the researcher that their average ideal hours were 7 hours a day in which the teams concentrated solely on their work with no disruptions.

Further, post adjustments to implement the suggested TESPs by the treatments in the actual environment were made. The details of these post modifications, along with indicative scenarios, can be viewed in the last section of Chapter 7.

Experimental Data: (Proposed subjects are Team-A and B, Agile coach / Capability Lead, while objects are Xproject, Y & Zproject)													
S#	**Spt No**	**Estimated Effort / SP as suggested by the Treatments**								**AE / SP**	**PHs**	**TMs**	**US**
		T1	**T2a**	**T2b**	**T3**	**T4a**	**T4b**	**T5**	**T6**				
Xproject Data: - Subjects/Treatments: Team-A, Agile coach / T1, T2, T5													
1	27	18.92	20.00	20.59	-	-	-	21.21	-	21.88			10
2	28	15.56	15.22	15.56	-	-	-	16.67	-	16.67			8
3	29	14.29	16.67	17.07	-	-	-	15.56	-	15.91			8
4	30	8.97	9.86	10.77	-	-	-	9.86	-	12.28	700	10	12
5	31	9.09	10.45	12.07	-	-	-	10.00	-	9.09			8
6	32	11.29	12.73	14.58	-		-	12.28	-	14.29			12
Yproject Data: - Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5													
7	1	8.97	12.07	15.22	-	-	-	11.29	-	11.29			
8	2	-	-	-	10.29	9.21	10.61	-	12.07	11.29			
9	3	7.45	10.00	11.29	-	-	-	8.33	-	8.97	350	5	5
10	4	-	-	-	8.33	7.78	8.75	-	9.46	9.46			
11	5	8.97	11.29	13.46	-	-	-	10.29	-	8.97			
12	6				10.94	11.29	12.96	-	12.07	10.94			
Zproject Data: - Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5													
13	1	-	-	-	10.29	11.67	12.96	-	13.46	11.29			
14	2	8.33	9.21	10.61	-	-	-	9.46	-	9.46			
15	3	-	-	-	7.78	9.46	11.29	-	8.33	8.75			
16	4	6.73	7.78	8.75	-	-	-	7.45	-	7.95	350	5	5
17	5	9.46	11.67	12.96	-	-	-	10.29	-	10.29			
18	6	-	-	-	10.94	11.29	12.96	-	12.07	10.94			
SP: Story Point **S#**: Sample Number **Spt No**: Sprint number **TM**: Team member **US**: User Stories **T1, T3, T5, T6**: Treatment-1, 3, 5 and 6 while **T2a**, **T4a** are the treatment-2, 4 with PERT variant-1 and **T2b**, **T4b** treatment-2, 4 with PERT variant-2													

Table 9.2 Effort per SP for the Experimental Data-Points of Treatments T1 to T6 of AUSIT's Projects

9.2 Descriptive statistics

Minitab 19, IBM SPSS, MS Excel 2019 and an online statistical calculator (Statstestcalc, n.d.) were utilized as statistical software tools to perform descriptive and other similar statistical tests/analyses in this study. The six treatments were applied variably across the three real industry projects (i.e., Xproject, Yproject, and Zproject of AUSIT) as described fully in Chapter 7. To reiterate, the treatments were as follows: T1 - Planning Poker based estimation (PPBE), this was an existing methodology that teams were using for their estimations by using expert judgement based on gut feelings, analogy, or Top-Down techniques but excluding Bottom-Up methods; T2a - treatment T1 extended with variant-1 of PERT; T2b - treatment T1 extended with PERT's variant-2; T3 - Planning Poker extended with expert judgement by following strictly the Bottom-Up approach based on the work breakdown structure; T4a - treatment T3 extended with PERT's variant-1; T4b - treatment T3 extended with PERT's variant-2; T5 - treatment T1 extended with the COEEMO, the proposed parametric model of this study; and T6 - treatment T3 extended with the COEEMO.

Measures of the central tendency and dispersion of the total estimated SPs (TESPs) and the actual finished story-points (i.e., story-points that meet the development team's definition of "Done" and are ready for delivery to the client) per sprint are reported using descriptive statistics. However, Individual Value Plots (IVPs) are used to compare TESPs, AFSPs, Estimated and actual effort per story-point of the treatments because IVPs effective in depicting low numbers of data points, especially if data-points are fewer than 50 (Frost, 2019).

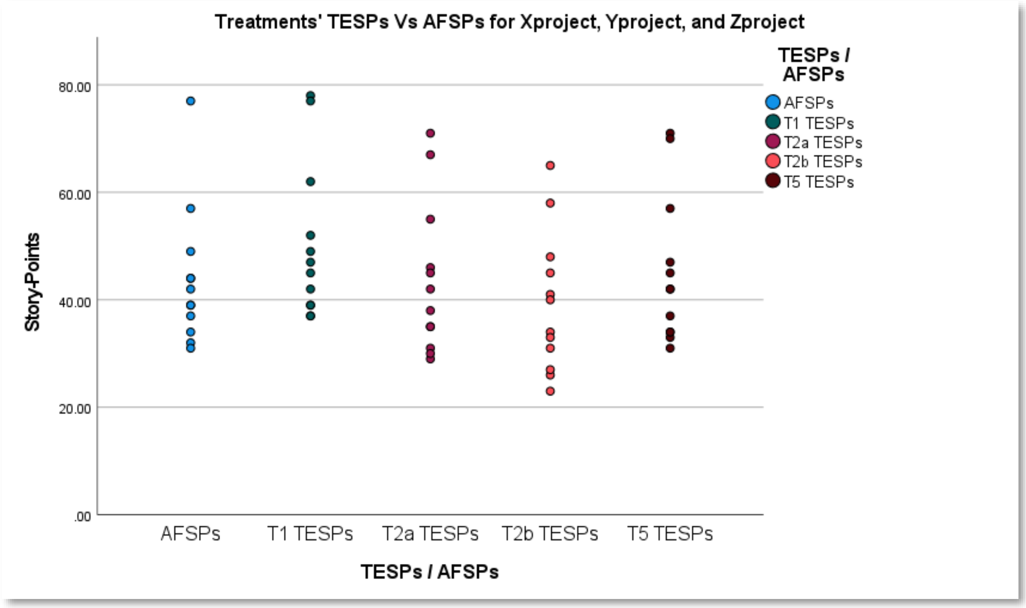
9.2.1 AFSPs and TESPs for Treatments ─ T1, T2a-b, and T5

Four treatments T1, T2a, T2b, and T5 were applied on all sprints of Xproject and alternate sprints of Yproject and Zproject (i.e., 12 sprints in total). Descriptive statistics and IVPs, as presented in Figure 9.1, show T1 has the highest central tendency of the four treatments (i.e., Maximum TESP is 78) while T2b has the lowest (i.e., Minimum TESP is 23).

Regarding variability, T2b has a broader range than the other treatments. Further, all treatments resulted in a different spread of values (i.e., all treatments have distinct Mean values that include 50.3, 43.7, 39.3, and 45.3) when they are compared to each other or with Actual finished story-points (AFSPs). The data distributions of the TESPs for all treatments, except AFSPs, seem to be more toward the normal because the values of skewness and kurtosis are in between 0 and ± 2.03 (for the perfect normal distribution the values of skewness and kurtosis must be equal to 0 and 3 respectively). However, due to small number of data-points (i.e., only 18 in this intended study) such deduction could be misleading.

Descriptive Statistics

	N	Range	Minimum	Maximum	Mean		Std. Deviation	Variance	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
T1 TESPs	12	41.00	37.00	78.00	50.3333	4.20918	14.58102	212.606	1.172	.637	.192	1.232
T2a TESPs	12	42.00	29.00	71.00	43.6667	4.05891	14.06048	197.697	.996	.637	-.041	1.232
T2b TESPs	12	42.00	23.00	65.00	39.2500	3.74596	12.97638	168.386	.755	.637	-.142	1.232
T5 TESPs	12	40.00	31.00	71.00	45.2500	4.00024	13.85723	192.023	1.052	.637	-.039	1.232
AFSPs	12	46.00	31.00	77.00	43.7500	3.69095	12.78582	163.477	1.779	.637	3.699	1.232
Valid N (listwise)	12											



T1, T5: Treatments-1 and 5 **T2a:** Treatment-2 with PERT variant-1 **T2b:** Treatment-2 with PERT variant-2
TESPs: Total Estimated Story points **AFSPs:** Actual Finished /done Story points by the team

Figure 9.1 Descriptive Statistics and IVPs graph for AFSPs and TESPs for Treatments ─ T1, T2a, T2b, and T5

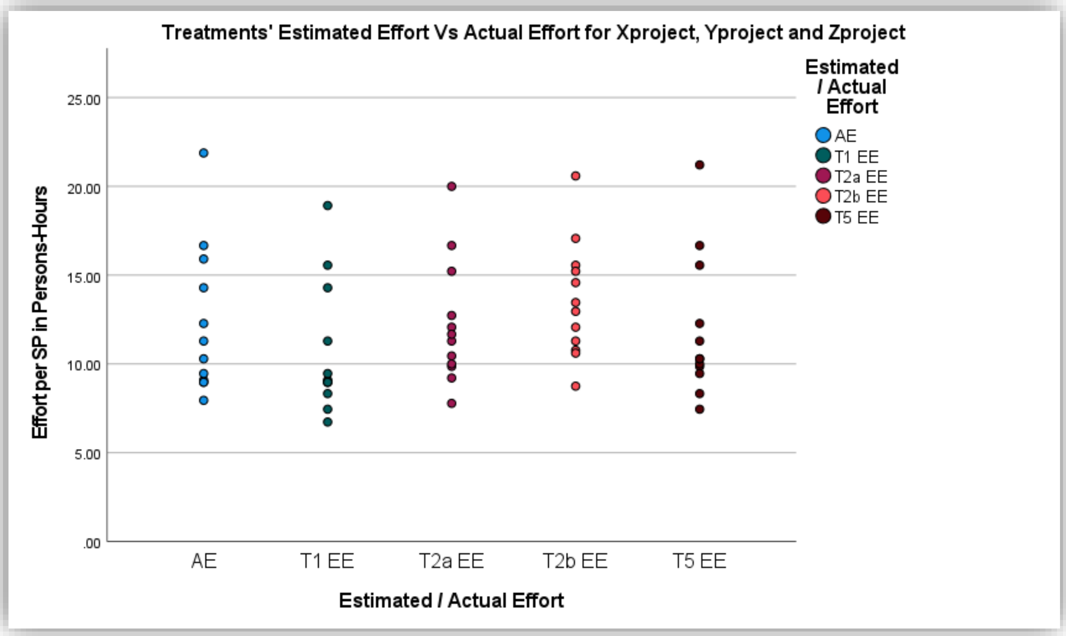
9.2.2 Actual and Estimated Effort for Treatments ─ T1, T2a, T2b, and T5

Descriptive statistics and Individual Value Plots (IVPs) for the estimated (EE/SP) and the actual (AE/SP) effort in Person-Hours per SP in the sprints for treatments T1, T2a-b and T5 are depicted in Figure 9.2. The statistical results for the EE/SP of these treatments indicate that the Mean of all treatments are indeed distinct (i.e., 10.7, 12.2, 13.6, and 11.9 respectively) which is as expected given the different treatments. Moreover, this aspect is also substantiated through the IVPs (as exhibited in Figure 9.2), which show the diverse spread of values when they are compared to each other or with Actual Effort (AE/SP).

The Mean of EE/SP for T1 is smaller in value among the treatments as well as than the Mean of the AE/SP (the latter value is 12.3) that foresees the higher deviations in the effort estimates for this treatment as compared to the others. This specific aspect is also revealed by the IVPs, which show that T5 has the highest central tendency (i.e., Effort per SP is 21.21 PHs) of the 4 treatments while T1 has the lowest (i.e., Effort per SP 6.73 PHs).

Descriptive Statistics

	N	Range	Minimum	Maximum	Mean		Std. Deviation	Variance	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
T1 EE/SP	12	12.19	6.73	18.92	10.6692	1.06409	3.68611	13.587	1.304	.637	.909	1.232
T2a EE/SP	12	12.22	7.78	20.00	12.2458	1.00445	3.47952	12.107	1.115	.637	.942	1.232
T2b EE/SP	12	11.84	8.75	20.59	13.5775	.94106	3.25994	10.627	.706	.637	.580	1.232
T5 EE/SP	12	13.76	7.45	21.21	11.8908	1.15075	3.98630	15.891	1.373	.637	1.504	1.232
AE/SP	12	13.93	7.95	21.88	12.2542	1.20721	4.18190	17.488	1.227	.637	1.075	1.232
Valid N (listwise)	12											



T1, T5: Treatments-1 and 5 **T2a:** Treatment-2 with PERT variant-1 **T2b:** Treatment-2 with PERT variant-2
EE: Estimated Effort **AE:** Actual Effort **PHs**: Persons-Hours

Figure 9.2 Descriptive Stats. and IVPs graph for Actual/Estimated Effort for Treatments ─ T1, T2a-b and T5

Regarding variability, T1 has a broader range than the other treatments. Moreover, the values of skewness and kurtosis are in between 0 and ±1.5 as shown in the descriptive statistics.

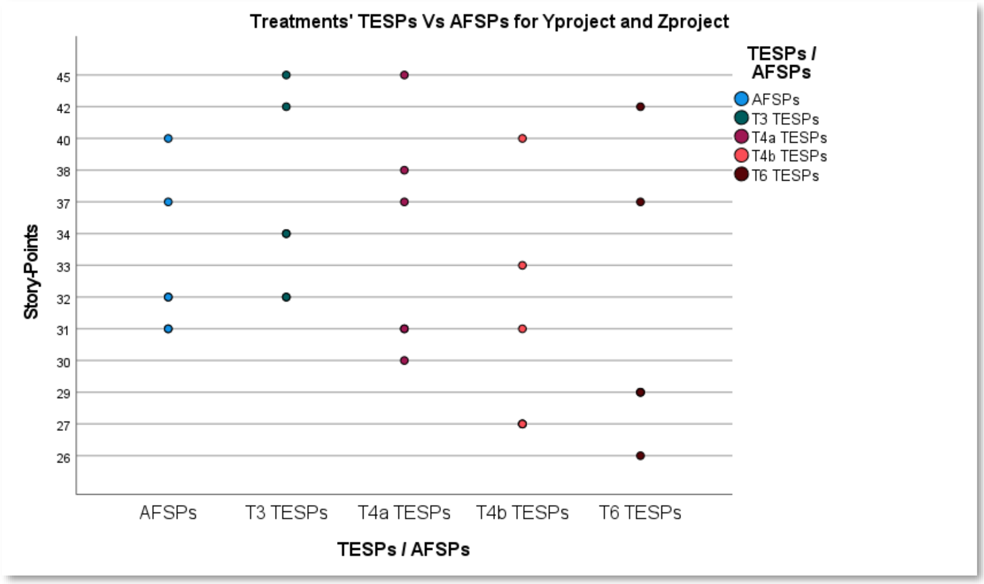
9.2.3 AFSPs and TESPs for Treatments ─ T3, T4a, T4b, and T6

Four different treatments were employed on the alternate sprints of the Yproject (i.e., T1, T2a-b, T5 on sprint number 1, 3 and 5 and T3, T4a-b, T6 on sprint number 2, 4 and 6) and Zproject (i.e., T1, T2a-b, T5 on sprint number 2, 4 and 6 and T3, T4a-b, T6 on sprint number 1, 3, and 6). Descriptive statistics and IVPs, as presented in Figure 9.3, show T3 and T4a to have the highest central tendencies of the four treatments (i.e., Maximum TESP is 45 in both treatments) while T6 has the lowest (i.e., Minimum TESP is 26).

Regarding variability, T6 has a broader range than the other treatments. Further, all treatments have shown different value spreads (i.e., all treatments have distinct Mean values that include 36.5, 35.3, 30.8, and 32) when they are compared to each other or with Actual finished story-points (AFSPs). The values of skewness and kurtosis are between 0 and ± 1.74.

Descriptive Statistics

	N	Range	Minimum	Maximum	Mean		Std. Deviation	Variance	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
T3 TESPs	6	13.00	32.00	45.00	36.5000	2.27669	5.57674	31.100	.981	.845	-1.160	1.741
T4a TESPs	6	15.00	30.00	45.00	35.3333	2.37580	5.81951	33.867	.932	.845	.039	1.741
T4b TESPs	6	13.00	27.00	40.00	30.8333	2.10423	5.15429	26.567	1.340	.845	1.417	1.741
T6 TESPs	6	16.00	26.00	42.00	32.0000	2.50333	6.13188	37.600	1.077	.845	-.227	1.741
AFSPs	6	9.00	31.00	40.00	33.8333	1.53659	3.76386	14.167	1.173	.845	-.327	1.741
Valid N (listwise)	6											



T3, T6: Treatments-1 and 6 **T4a:** Treatment-4 with PERT variant-1 **T4b:** Treatment-4 with PERT variant-2
TESPs: Total Estimated SPs **AFSPs:** Actual Finished SPs

Figure 9.3 Descriptive Statistics and IVPs graph for AFSPs and TESPs for Treatments ─ T3, T4a-b, T6

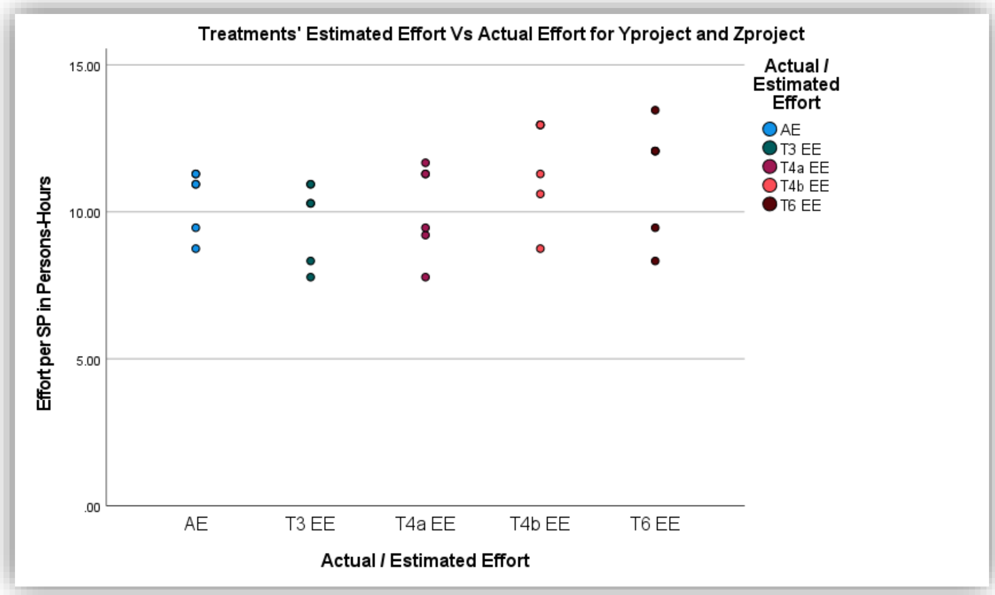
9.2.4 Actual and Estimated Effort for Treatments ─ T3, T4a, T4b, and T6

Descriptive statistics and Individual Value Plots (IVPs) for the estimated (EE/SP) and the actual (AE/SP) effort in Person-Hours per SP in the sprints for treatments T3, T4a, T4b and T6 are represented in Figure 9.4. The statistical results for the EE/SP of these treatments indicate that the Mean of all treatments are distinctive (i.e., 9.7, 10.1, 11.6 and 10.4 respectively) which is as expected given the different treatments. Moreover, this aspect is also validated through IVPs (as displayed in Figure 9.4) which indicates the varied spread of values when they are matched to each other or with Actual Effort (AE/SP).

The Mean of EE/SP for T3 (i.e., 9.8) is smaller in value among the treatments as well as than the Mean of the AE/SP (the latter value is 10.4) that foresees the higher deviations in the effort estimates for this treatment as compared to the others. This particular aspect is also revealed by the IVPs, which show that T6 has the highest central tendency of the 4 treatments (i.e., Effort per SP is 13.46 PHs) while T3 and T4a have the lowest (i.e., Effort per SP is 7.78 PHs in both treatments). Regarding variability, T4a has a greater range than the other treatments. Moreover, the values of skewness and kurtosis are in between -1.48 and -0.09 as shown in descriptive statistics.

Descriptive Statistics

	N	Range	Minimum	Maximum	Mean		Std. Deviation	Variance	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
T3 EE	6	3.16	7.78	10.94	9.7617	.55713	1.36469	1.862	-.844	.845	-1.480	1.741
T4a EE	6	3.89	7.78	11.67	10.1167	.62927	1.54140	2.376	-.558	.845	-1.254	1.741
T4b EE	6	4.21	8.75	12.96	11.5883	.70111	1.71736	2.949	-.953	.845	-.098	1.741
T6 EE	6	5.13	8.33	13.46	11.2433	.78807	1.93036	3.726	-.743	.845	-.821	1.741
AE	6	2.54	8.75	11.29	10.4450	.43823	1.07344	1.152	-1.061	.845	-.801	1.741
Valid N (listwise)	6											



T3, T6: Treatments-1 and 5 **T4a:** Treatment-4 with PERT variant-1 **T4b:** Treatment-4 with PERT variant-2
EE: Estimated Effort **AE:** Actual Effort **PHs**: Persons-Hours

Figure 9.4 Descriptive Stats. and IVPs graph for Actual/Estimated Effort for Treatments – T3, T4a-b, and T6

9.3 Normality Test

Numerous statistical methods may be utilized to check the distribution of data and whether or not it is normally distributed, and further, based on the normality of the data, different parametric or non-parametric tests are suggested to avoid deriving the wrong interpretations from the data. For example, if the data-points are not normally distributed in the continuous data type, then the representative mean (i.e., applying a parametric method) instead of the median (i.e., using a non-parametric method) to compare between groups may mislead the final outcome from the data (Mishra et al., 2019).

The statistical analysis reported in Section 9.2 and further, the values of the skewness and kurtosis of the data sets, reveal that the spread of the data is normally distributed. However, due to small sample size in the study, i.e., only 18 data-points from all of the six treatments, the deduction of normality in the distributions could be deceptive. Therefore, both the parametric one-way ANOVA test for normally distributed data and the non-parametric Kruskal-Wallis test for non-normally distributed data are applied to test the Null hypothesis in section 9.4 of this chapter. Moreover, the normality testing is done by using the Ryan-Joiner test (that is similar to but simpler than the Shapiro-Wilk test) to determine the normality of the data. The results of these normality tests are exhibited in Figures 9.5 and 9.6.

The outputs shown in Figures 9.5 (A, B, C, D) and 9.6 (E, F, G, H) reflect the normality tests for the TESPs as calculated by Treatments 1-6. The results suggest that the data follow the normal distribution in all cases except for one treatment, i.e., T1, which shows a p value equal to 0.05. That can be considered as a border line value because the value of p should be >0.05 for a normal distribution, otherwise it should be equal to or smaller than 0.05 when the data is non-normally distributed (ADT, 2018). The probability graphs as depicted in Figures 9.5 and 9.6 show most of the data-points on or close to the straight lines, in contrast to the non-normal spread of data where most of the points are not close to or placed on the straight lines.

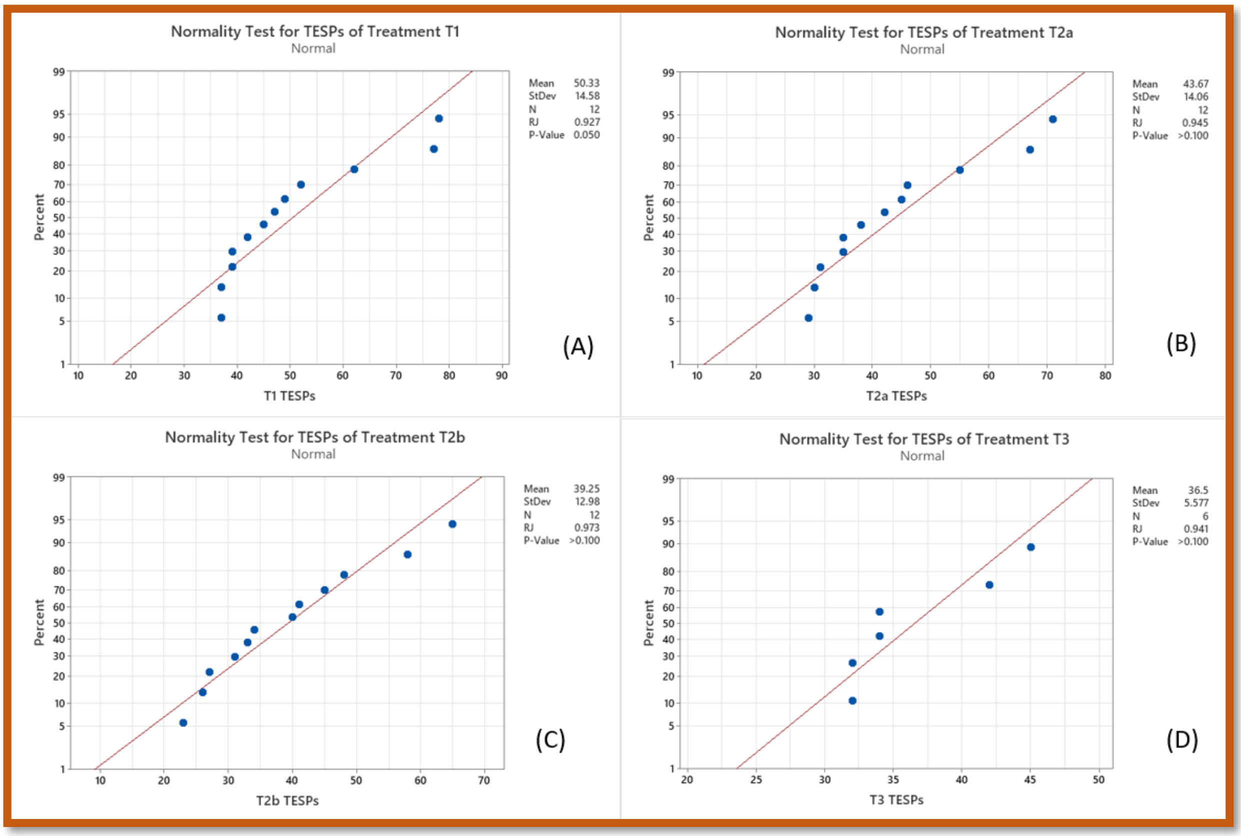


Figure 9.5 Normality Tests of the TESPs as suggested by Treatments T1 to T3

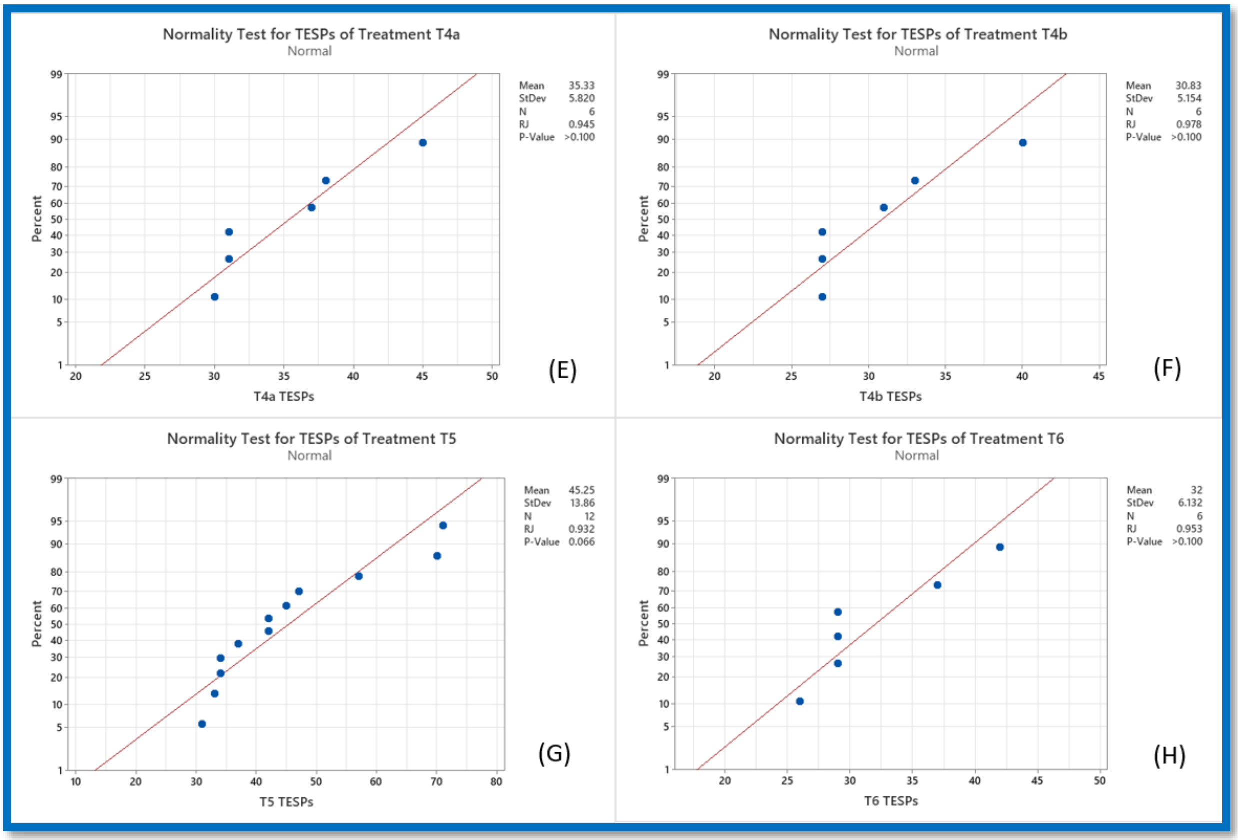


Figure 9.6 Normality Tests of the TESPs as suggested by Treatments T4 to T6

9.4 Hypothesis Evaluations and Analysis

In Chapter 6, the detailed experimental design was described, and a hypothesis was formulated, including Null (H_0) and Alternate (H_1) forms. The acceptance criterion for the Null hypothesis states that if the same results are returned by all the treatments (i.e., equal Mean 'μ') then the Null hypothesis is accepted i.e. H_0: $\mu1 = \mu2 = \mu3 = \mu4 = \mu5 = \mu6$ where the mean is the same for all the treatments; but on the rejection of H_0, the alternative hypothesis is accepted, which means that the results of the treatments are not the same, i.e., H_1: $\mu1 \neq \mu2 \neq \mu3 \neq \mu4 \neq \mu5 \neq \mu6$ for at least one pair.

Appropriate statistical tests are used to test the above hypothesis. They include analysis of variance (ANOVA), a parametric test that makes assumptions about the distribution of the population from which data are drawn, and the Kruskal-Wallis test, a non-parametric test that make no such assumptions. The ANOVA test compares variabilities due to the treatment and the random error as well as the samples of the same Mean values (a summary of the tests and criteria is given in Table 6.1 of Chapter 6).

The six treatments' details are presented briefly here but comprehensive descriptions are covered in Chapter 6: T1 - Planning Poker based estimation (PPBE), an existing methodology which teams were using for their estimations by using expert judgement based on gut feelings, analogy, and/or Top-Down techniques but excluding Bottom-Up methods; T2a - treatment T1 extended with variant-1 of PERT; T2b - treatment T1 extended with PERT's variant-2; T3 - Planning Poker extended with expert judgement by following strictly the Bottom-Up approach based on the work breakdown structure; T4a - treatment T3 extended with PERT's variant-1; T4b - treatment T3 extended with PERT's variant-2; T5 - treatment T1 extended with the COEEMO, the proposed parametric model of this study (whose details are provided in Chapter 3), and T6 - treatment T3 extended with the COEEMO. The six treatments were variously applied in three real industry projects (i.e., Xproject, Yproject, and Zproject). The experiments were conducted by two agile development teams in an Australian based company referred to here as AUSIT. The complete specifics of the projects (i.e., the nature of the projects, number of sprints, types of projects), teams (i.e., team IDs, number of team members), and the treatments, along with their significant data-points, were presented in Chapter 7.

Based on the descriptive statistical analysis reported in Section 9.2 and the results of the normality tests as shown in Figures 9.5 and 9.6 it is asserted that the data of the treatments are normally distributed, therefore, the parametric one-way ANOVA test is applied to evaluate the null and the alternate hypothesis. However, and as noted above, due to the small sample size the non-parametric Kruskal-Wallis test is also applied.

Data Table

Groups:	T1 TESPs	T2a TESPs	T2b TESPs	T3 TESPs	T4a TESPs	T4b TESPs	T5 TESPs	T6 TESPs
Skewness:	1.1722	0.9955	0.7552	0.9808	0.9322	1.3396	1.0517	1.0774
Excess kurtosis:	0.1921	-0.04053	-0.142	-1.1597	0.03894	1.4171	-0.03932	-0.2272
Normality	0.01807	0.0848	0.4847	0.07621	0.2571	0.09334	0.03789	0.1394
Median:	46	40	37	34	34	29	42	29
Mean:	50.33333	43.66667	39.25000	36.50000	35.33333	30.83333	45.25000	32.00000
Sample size (n):	12	12	12	6	6	6	12	6
Rank sum (R):	632.5	490.5	401.5	194	169.5	97.5	531	111.5
R^2/n:	33338.0208	20049.1875	13433.5208	6272.6667	4788.375	1584.375	23496.75	2072.0417

(A)

Source	DF	Sum of Square	Mean Square	F Statistic	P-value
Groups **(between groups)**	7	2804.444454	400.634922	2.810395	0.0128984
Error **(within groups)**	64	9123.499870	142.554685		
Total	71	11927.94432	167.999216		

(B)

One Way ANOVA test, using F distribution df(7,64) (right tailed)

1. H_0 hypothesis

Since p-value<α, H_0 is rejected.

Some of the groups' averages consider to be not equal.

In other words, the difference between the averages of some groups is big enough to be statistically significant.

2. P-value

p-value equals **0.0128984**, [p(x ≤ F) = 0.987102]. This means that the chance of type1 error (rejecting a correct H_0) is small: 0.01290 (1.29%)

The smaller the p-value the stronger it support H_1

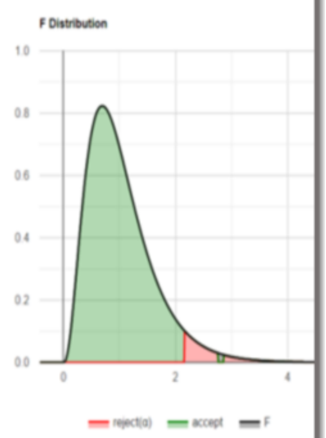
3. The statistics

The test statistic F equals **2.810395**, is not in the 95% critical value accepted range: [-∞ : 2.1564]

4. Effect size

The observed effect size f is **large** (0.55). That indicates that the magnitude of the difference between the averages is large.

The η^2 equals 0.24. It means that the **group** explains 23.5% of the variance from the average (similar to R^2 in the linear regression)



(C)

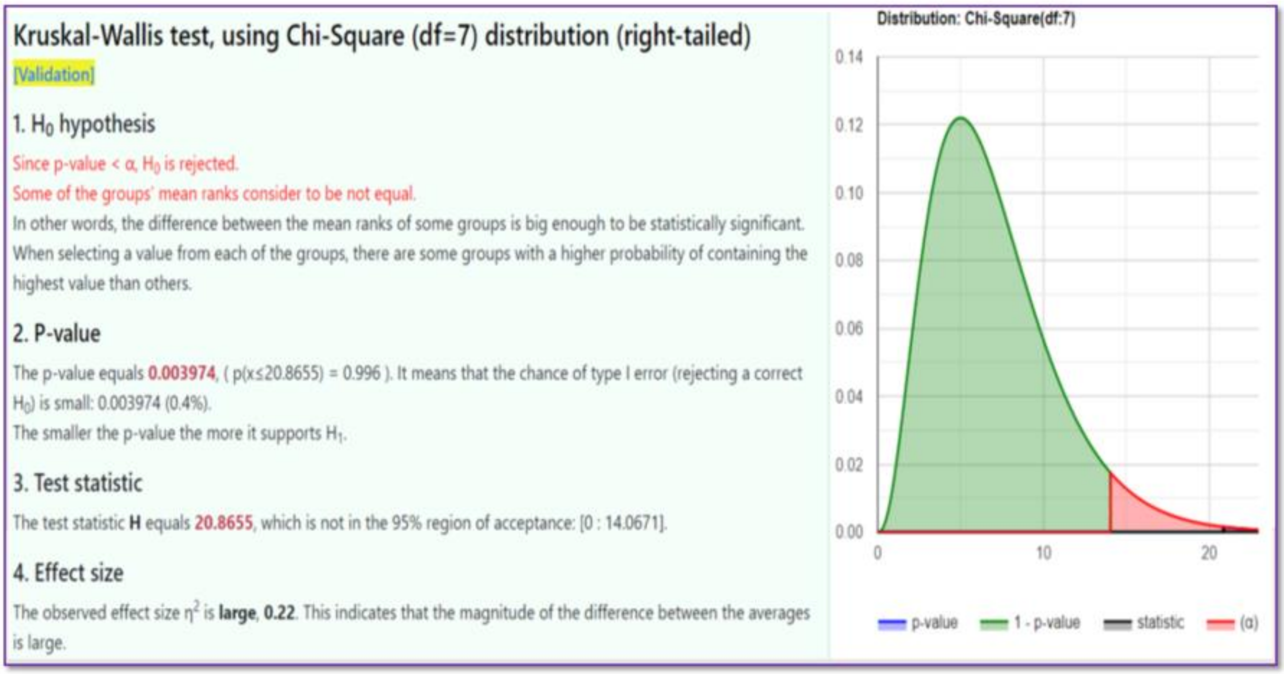
Figure 9.7 One Way ANOVA Test of the TESPs for all Treatments - Xproject, Yproject & Zproject

The results of both tests are shown in Figure 9.7 (A, B, and C) and 9.8 (A, B), which indicate the p-values are less than 0.05. Based on these values the alternate hypothesis (H_1) can be accepted while rejecting the null hypothesis (H_0), which means that Treatments 1-6 have produced statistically significant differences among their results.

Data Table

Groups:	T1 TESPs	T2a TESPs	T2b TESPs	T3 TESPs	T4a TESPs	T4b TESPs	T5 TESPs	T6 TESPs
Skewness:	1.1722	0.9955	0.7552	0.9808	0.9322	1.3396	1.0517	1.0774
Excess kurtosis:	0.1921	-0.04053	-0.142	-1.1597	0.03894	1.4171	-0.03932	-0.2272
Normality	0.01807	0.0848	0.4847	0.07621	0.2571	0.09334	0.03789	0.1394
Median:	46	40	37	34	34	29	42	29
Mean:	50.33333	43.66667	39.25000	36.50000	35.33333	30.83333	45.25000	32.00000
Sample size (n):	12	12	12	6	6	6	12	6
Rank sum (R):	632.5	490.5	401.5	194	169.5	97.5	531	111.5
R^2/n:	33338.0208	20049.1875	13433.5208	6272.6667	4788.375	1584.375	23496.75	2072.0417

(A)



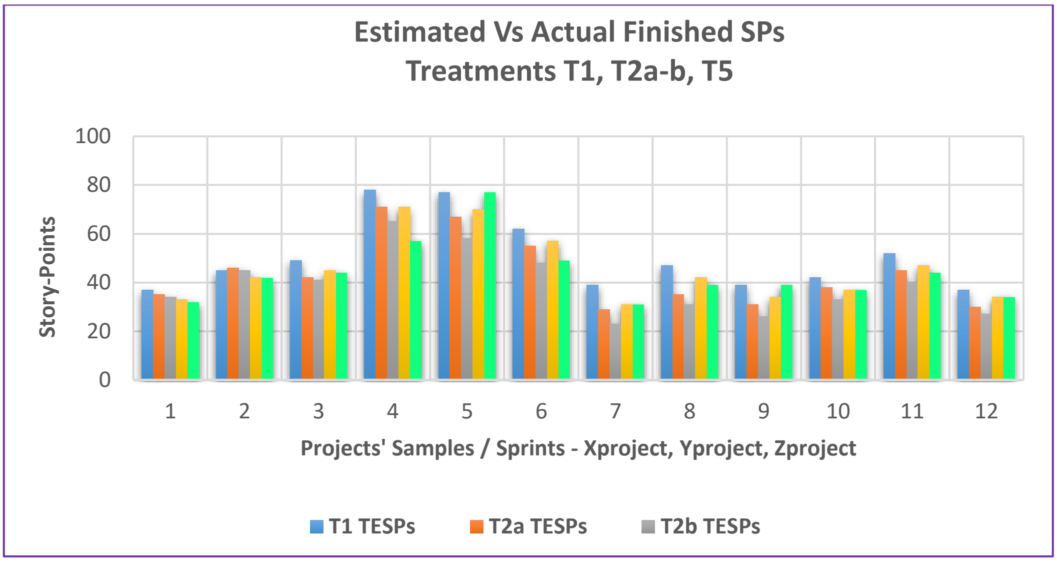
(B)

Figure 9.8 Kruskal-Wallis Test of the TESPs for all Treatments - Xproject, Yproject & Zproject

9.5 Treatments Analysis

9.5.1 Estimated Vs Actual Finished Story-Points Analysis for Treatments T1, T2a-b, and T5

The total estimated SPs (TESPs, as projected by treatments T1, T2a-b, and T5) and the actual finished SPs (AFSPs) are shown in Figure 9.9 as bar graphs. Treatment T5 estimated story-points closest or equal to the actual finished SPs (AFSPs) in 10 of the 12 sprints. Treatments T1 and T2b deviate most frequently from AFSPs: T1 estimates the highest value of the treatments in 11 of the 12 sprints, and T2b generates the smallest value in 9 sprints. Treatment T2a produces almost mixed results, over- and under-estimating on 6 occasions each as compared to the actual finished SPs.

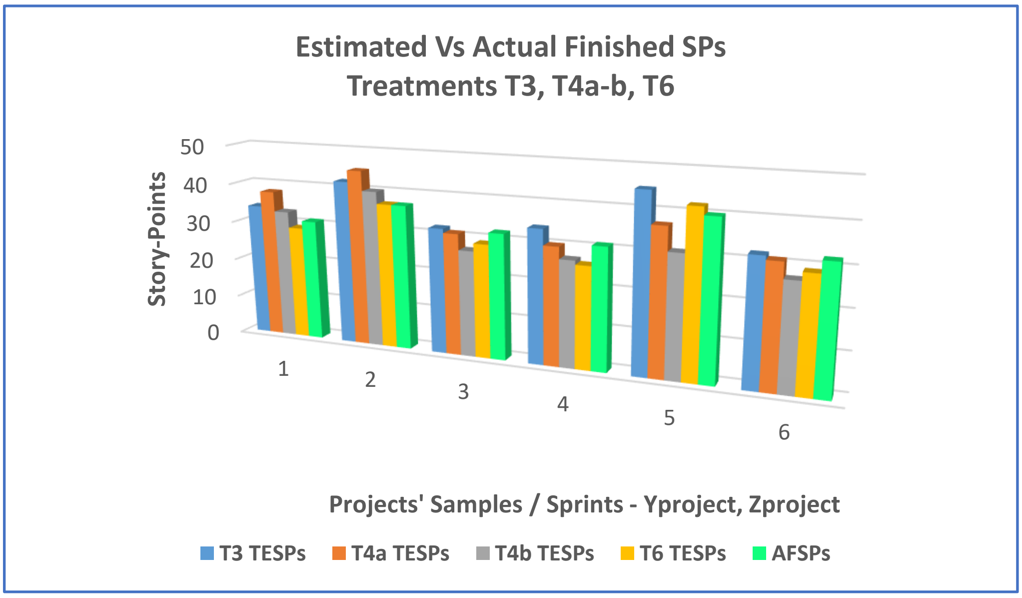


TESPs: Total Estimated Story-Points **T1, T2a, T2b, T5**: Treatment-1, 2 and 5 **AFSPs**: Actual Finished Story-Points

Figure 9.9 Estimated Vs Actual Finished SPs assessments — Treatments T1, T2a-b, T5

9.5.2 Estimated Vs Actual Finished Story-Points Analysis — Treatments T3, T4a-b, and T6

The total estimated SPs (TESPs, as projected by treatments T3, T4a-b, and T6) and the actual finished SPs (AFSPs) are shown in Figure 9.10 as bar graphs. Treatment T6 estimated story-points closest or equal to the actual finished SPs (AFSPs) in 3 of the 6 sprints. Treatments T3 and T4b deviate most frequently from AFSPs: T3 estimates the highest value of the treatments in 3 of the 6 sprints, and T4b generates the smallest value in 4 sprints. Treatment T4a produces closest TESPs in 3 while it demonstrates 2, and 1 sprint with over- and under-estimating TESPs respectively as compared to the actual finished SPs.

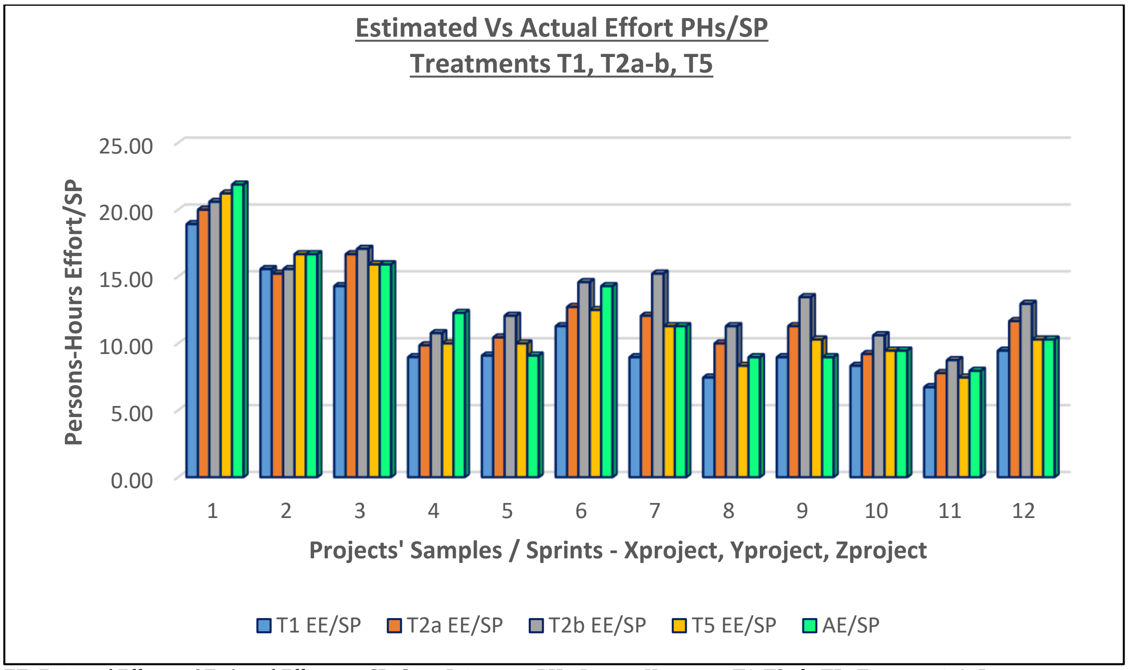


TESPs: Total Estimated Story-Points **T3, T4a, T4b, T6**: Treatment-3, 4 and 6 **AFSPs**: Actual Finished Story-Points

Figure 9.10 Estimated Vs Actual Finished SPs assessments — Treatments T3, T4a-b, T6

9.5.3 Estimated Vs Actual Effort Analysis — Treatments T1, T2a-b, and T5

The effort estimates generated by treatments T1, T2a-b, and T5 and the actual effort in PHs per SP are displayed as bar graphs in Figure 9.11. Treatment T5 produces effort estimates closest or equal to the actual effort in 10 of the 12 sprints. Treatments T1 and T2b deviate most frequently from the actual values: T1 estimates the smallest value of the treatments in 11 of the 12 sprints, and T2b generates the highest value in 9 sprints. Treatment T2a produces mixed results, over- and under-estimating on 6 occasions each as compared to the actual effort/SP.

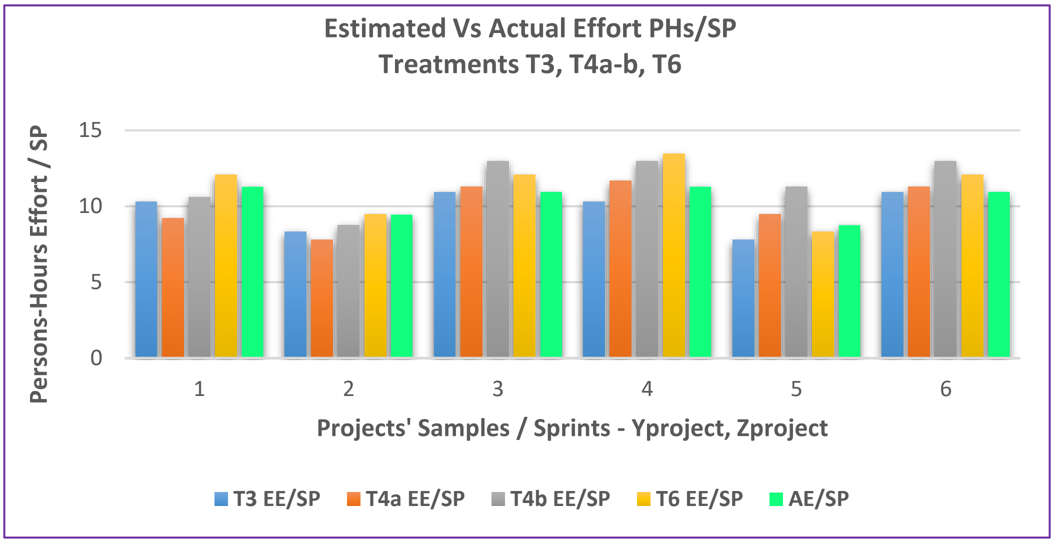


EE: Estimated Effort **AE**: Actual Effort **SP**: Story-Point **PHs**: Persons-Hours **T1, T2a-b, T5** : Treatment-1, 2, 5

Figure 9.11 Estimated Vs Actual Effort assessments — Treatments T1, T2a-b, T5

9.5.4 Estimated Vs Actual Effort Analysis — Treatments T3, T4a-b, and T6

The estimated effort projected by treatments T3, T4a-b, and T6 and the actual effort in PHs per SP are exhibited as bar graphs in Figure 9.12. Treatment T6 results in effort estimates closest or equal to the actual effort in 3 sprints. Treatments T3 produces under-estimates in 3 sprints while T4b displays over-estimates in 4 sprints as compared to the actual effort per SP (AE/P).

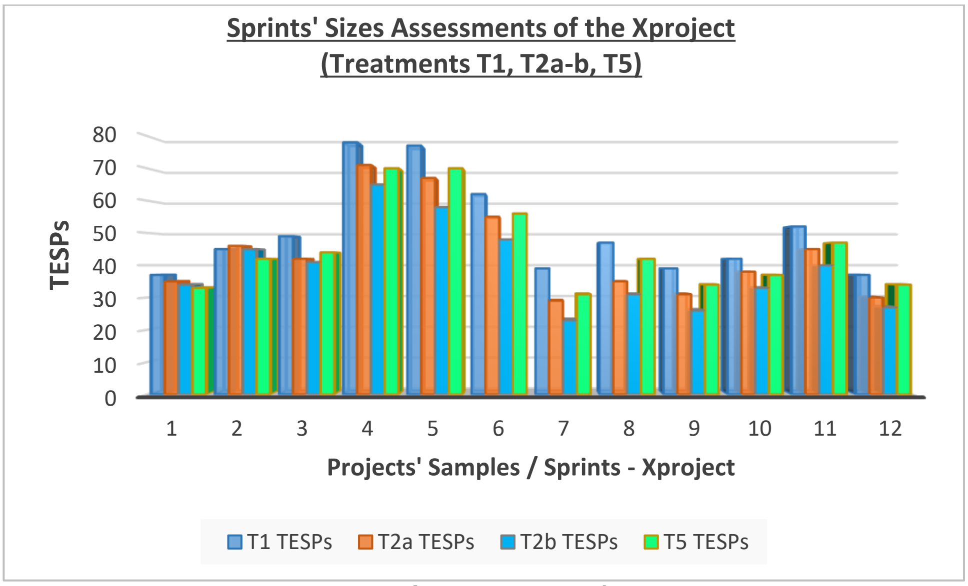


EE: Estimated Effort **AE**: Actual Effort **SP**: Story-Point **MHs**: Man-Hours **T3, T4a-b, T6** : Treatment-3, 4, 6

Figure 9.12 Estimated Vs Actual Effort assessments — Treatments T3, T4a-b, T6

9.5.5 Sprint size Analysis for the Xproject's Treatments — T1, T2a-b, T5

Figure 9.13 presents a bar graph comparisons of the sizes (i.e., total estimated SPs) of the sprints among the treatments (i.e., T1, T2a-b, T5) as employed on the Xproject. The graph shows that Treatments T1 and T2b most often estimate sprints of large and small size (10 sprints in each case) as compared to the other treatments.

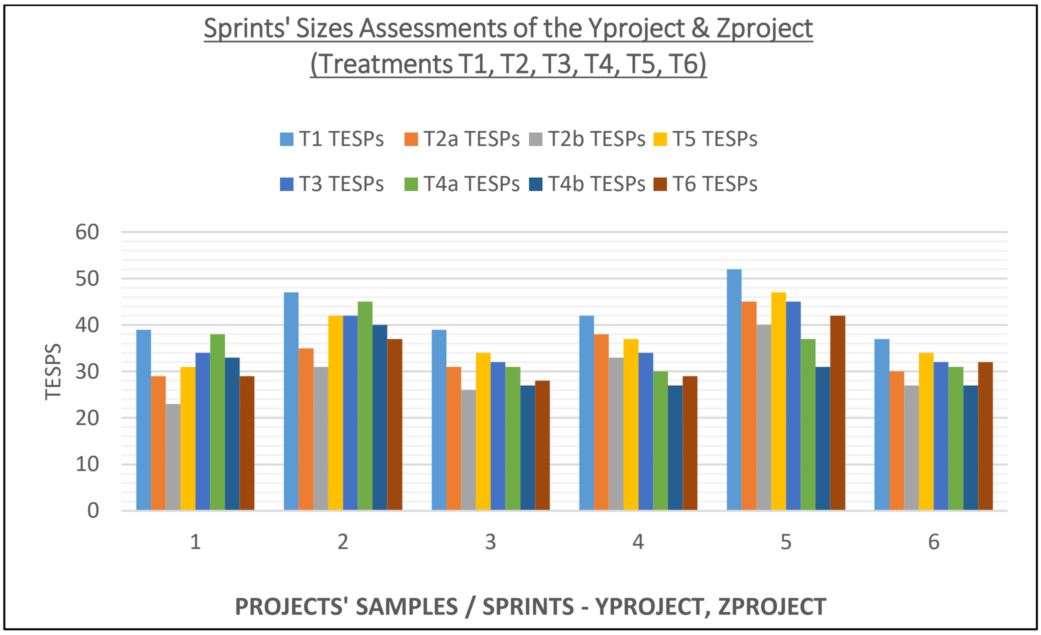


TESPs: Total Estimated Story-Points **T1, T2a, T2b, T5**: Treatment-1, 2 and 5

Figure 9.13 Sprints' sizes comparisons among the Treatments for the Xproject

9.5.6 Sprint size Analysis for the Treatments T1 to T6 — Yproject & Zproject

Figure 9.14 depicts a bar graph comparison for the sprints' sizes (i.e., total estimated SPs) as projected by all Treatments 1-6 for the Yproject and the Zproject. The graph illustrates that, in all the sprints across both projects, Treatment T1 generates larger values while Treatments T2b and T4b tend to estimate smaller-sized sprints.



TESPs: Total Estimated Story-Points **T1, T2, T3, T4, T5, T5**: Treatment-1, 2, 3, 4, 5 and 6

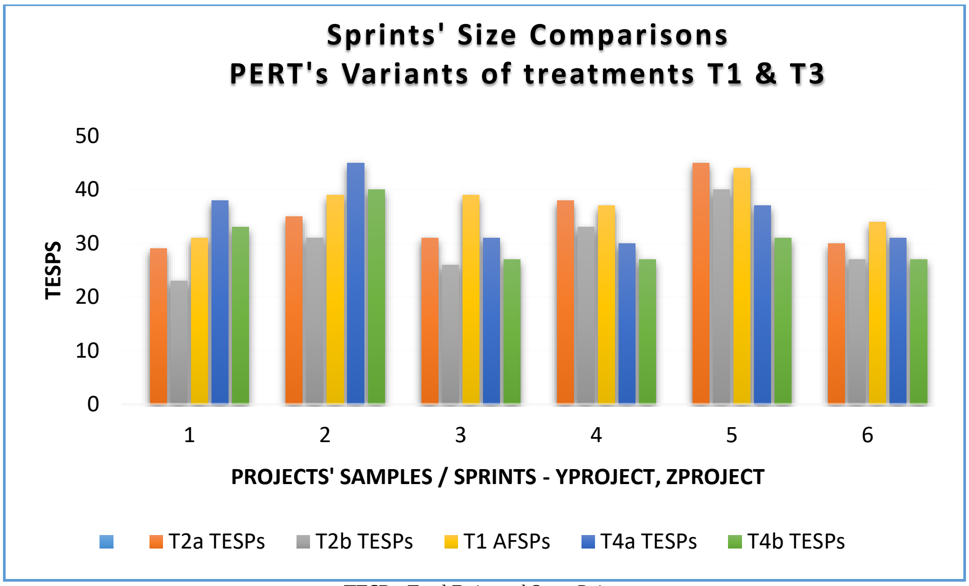
Figure 9.14 Sprints' sizes comparisons among the Treatments — Yproject & Zproject

9.5.7 Analysis of the Treatments Extended with PERT Variants

Treatments-2 and -4 – Planning Poker estimations based on expert judgement using (i) analogy, gut feeling and/or Top-Down methods, or (ii) using only Bottom-Up methods – form the basis of extended treatments using two variants of PERT (i.e., PERT-1 and PERT-2 as described in detail in Chapter 6). In this section, analyses are presented to report the performance of the PERT-1 and PERT-2 variants with respect to Treatment-1 and Treatment-3. For these comparisons only samples drawn from the Yproject and Zproject are considered because the contending treatments, T2a, T2b, T4a, and T4b, are applied on the alternate sprints of these projects.

9.5.7.1 Sprint size analysis for the PERT variants

The bar graph comparisons of both PERT extensions, T2a and T4a for PERT-1 and T2b and T4b for PERT-2, are depicted in Figure 9.15. This reveals that treatment T2a estimated large-sized sprints as compared to treatment T2b, and the same pattern can be viewed in regard to treatments T4a and T4b. However, none of the PERT variants of treatment T1 resulted in large-sized sprints as compared to treatment T1 while just one of the PERT variants of treatment T3 (i.e., T4a) resulted in large-sized sprints (for 2 sprints) as corresponding to treatment T3.

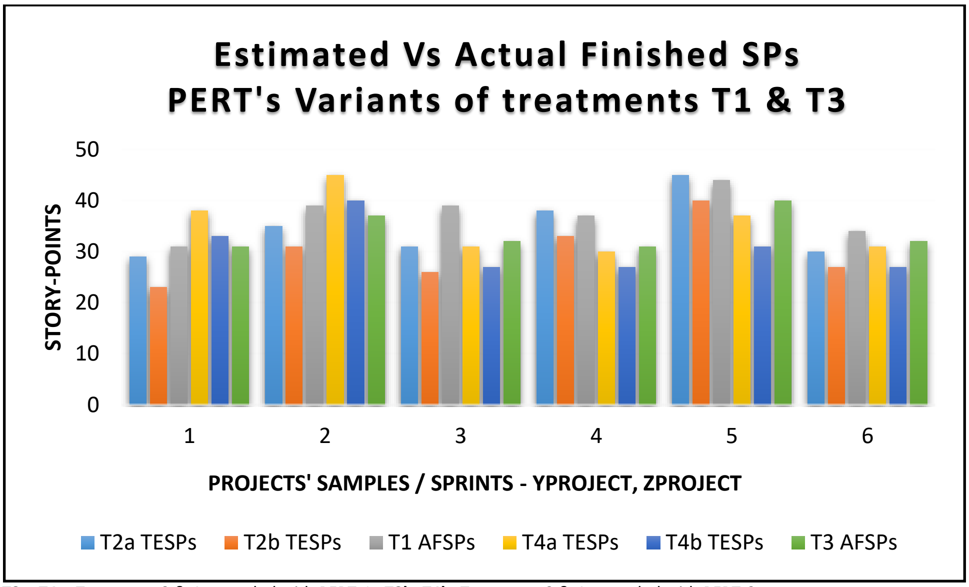


TESPs: Total Estimated Story-Points

Figure 9.15 Sprints' Sizes Comparisons ─ PERT's Variants Vs T1 & T3

9.5.7.2 Estimated Vs Finished Story-Points Analysis for the PERT's Variants

The bar graph presented in Figure 9.16 shows the total estimated story points (TESPs) according to the PERT variants of treatments T1 and T3 with respect to the actual finished story-points (AFSPs). Treatment T2a projects the TESPs of all of the sprints, when compared to the actual finished SPs of Treatment T1, with low to moderate deviations. In comparison, Treatment T2b reveals larger deviations in estimated SPs as compared to Treatment T2a. A similar pattern of deviations as is evident for the PERT variant of Treatment T1 for the estimated SPs is also observed in Treatment T3's variants of the PERT. Furthermore, treatments T2a and T4a show lesser deviations for the estimated SPs as compared to Treatments T2b and T4b, when these treatments are compared with the actual finished SPs. Consequently, PERT variant-1 demonstrates better performance in both treatments (i.e., T1 and T3) as compared to PERT variant-2.



T2a, T4a: Treatment 2 & 4 extended with PERT-1 **T2b, T4b**: Treatment 2 & 4 extended with PERT-2
AFSPs: Actual Finished Story-Points **TESPs**: Total Estimated SPs

Figure 9.16 Estimated Vs Finished SPs assessments for the PERT's variants

9.5.8 Effort Estimation Accuracy Assessments

In this final results section, analysis of the accuracy of the effort estimates produced by the six treatments as compared to the actual effort on the three projects of the AUSIT is reported. In these comparisons, the Magnitude of Relative Error (MRE) that is otherwise the most popular and commonly used metric of the last decade to measure the accuracy of effort estimation models (Usman et al., 2018) is not utilized, due to the criticisms on its usage in numerous studies (Shepperd et al., 2000; Kitchenham et al., 2001; Foss et al., 2002; Jørgensen, 2007; Myrtveit and Stensrud, 2012; Shepperd and MacDonell, 2012). Instead, the Balanced Relative Error (BRE) and Balanced Relative Error Bias (BREbias) metrics are applied due to their balanced performance in situations of over- and under-estimation (Moløkken-Østvold et al., 2008; Mahnic et al., 2012; Usman et al., 2018). The equations for these metrics are shown in Figure 9.17.

$$BRE = \frac{|\text{actual effort} - \text{estimated effort}|}{min(\text{ actual effort, estimated effort })}$$

$$BREbias = \frac{(\text{ actual effort} - \text{estimated effort })}{min(\text{ actual effort, estimated effort })}$$

Figure 9.17 Equations of the Metrics BRE and BREbias (Usman et al., 2018)

9.5.8.1 Accuracy Analysis for the Effort Estimates of Xproject — Treatments T1 to T5

Balanced Relative Error (BRE) and BREbias are calculated and presented in Tables 9.3 and 9.4, the latter table shows under/overestimated effort by the preceding negative and positive signs. The BRE and BREbias of the treatments are computed from the experimental data as given in Table 9.2, and the base formulae as shown in Figure 9.17.

BRE of the Effort Estimates per SP for the Treatments: T1, T2a, T2b, T3, T4a, T4b, T5, and T6									
S#	Spt No	Balanced Relative Error for the Effort Estimates in PHs per SP for the Treatments T1 - T6							
		T1	T2a	T2b	T3	T4a	T4b	T5	T6
Xproject Data: - Subjects/Treatments: Team-A, Agile coach / T1, T2, T5									
1	27	0.16	0.09	0.06	-	-	-	0.03	-
2	28	0.07	0.10	0.07	-	-	-	0.00	-
3	29	0.11	0.05	0.07	-	-	-	0.02	-
4	30	0.37	0.25	0.14	-	-	-	0.25	-
5	31	0.00	0.15	0.33	-	-	-	0.10	-
6	32	0.27	0.12	0.02	-	-	-	0.16	-
Mean BRE		T1 = 0.16	T2a = 0.13	T2b = 0.12	-	-	-	T5 = 0.09	-
SP: Story Point **S#**: Sample Number **Spt No**: Sprint number **BRE**: Balanced Relative Error **T1, T3, T5, T6**: Treatment-1, 3, 5 and 6 while **T2a**, **T4a** are the treatment-2, 4 with PERT variant-1 and **T2b**, **T4b** treatment-2, 4 with PERT variant-2									

Table 9.3 BRE for the Effort Estimates of Xproject – Treatments T1 to T5

The results of the BRE analysis reveal that Treatments T5 (i.e., Treatment T1 extended with COEEMO model) was the best performed method considered on the basis of Mean BRE when applied on Xproject. However, Treatments T2a, and T2b (i.e., both of these treatments used PERT's variants and extended form of Treatment T1) were second and third in order of performances on the basis of Mean BRE. All the three treatments (T2a, T2b, and T5) which were applied on Xproject demonstrated better estimation accuracies as compared to Treatment T1 that was the existing method of effort estimation.

BREbias for the Treatments: T1, T2a, T2b, T3, T4a, T4b, T5, and T6									
S#	Spt No	BREbias for the Treatments T1 - T6							
		T1	T2a	T2b	T3	T4a	T4b	T5	T6
Xproject Data: - Subjects/Treatments: Team-A, Agile coach / T1, T2, T5									
1	27	+0.16	+0.09	+0.06	-	-	-	+0.03	-
2	28	+0.07	+0.10	+0.07	-	-	-	0.00	-
3	29	+0.11	-0.05	-0.07	-	-	-	+0.02	-
4	30	+0.37	+0.25	+0.14	-	-	-	+0.25	-
5	31	+0.00	-0.15	-0.33	-	-	-	-0.10	-
6	32	+0.27	+0.12	-0.02	-	-	-	+0.16	-
SP: Story Point **S#**: Sample Number **Spt No**: Sprint number **BRE**: Balanced Relative Error **T1, T3, T5, T6**: Treatment-1, 3, 5 and 6 while **T2a**, **T4a** are the treatment-2, 4 with PERT variant-1 and **T2b**, **T4b** treatment-2, 4 with PERT variant-2									

Table 9.4 BREbias for the Effort Estimates of Xproject – Treatments T1 to T5

Based on the calculations of BREbias, as shown in Table 9.4, the number of sprints with under-, over- and accurate effort estimations as proposed by the treatments when applied on Xproject are counted and summarized in Table 9.5.

The results reveal that Treatment 1 and 5 resulted in precise accuracy in 1/6 sprints. Further, Treatment T1 produced the highest number of sprints with overestimated effort (i.e., 5/6 sprints) while Treatment T2b had the highest number of sprints with under-estimations of the effort (i.e., 3/6 sprints).

Treatment Name	Total Samples /Sprints	Number of Sprints		
		Over-estimated Effort	Under- estimated Effort	Accurate
T1	6	5	0	1
T2a	6	4	2	0
T2b	6	3	3	0
T3	Not Applied			
T4a				
T4b				
T5	6	4	1	1
T6	Not Applied			

Table 9.5 Treatments Vs Number of accurate / Over/underestimated Sprints for Xproject

9.5.8.2 Accuracy Analysis for Effort Estimates of Yproject and Zproject — Treatments T1-T6

Balanced Relative Error (BRE) and BREbias are calculated and presented in Tables 9.6 and 9.7, the latter table shows under/overestimated effort by the preceding negative and positive signs. The BRE and BREbias of the treatments are computed from the experimental data as given in Table 9.2, and the base formulae as shown in Figure 9.17.

BRE of the Effort Estimates per SP for the Treatments: T1, T2a, T2b, T3, T4a, T4b, T5, and T6									
S#	**Spt No**	**Balanced Relative Error for the Effort Estimates in PHs per SP for the Treatments T1 - T6**							
		T1	**T2a**	**T2b**	**T3**	**T4a**	**T4b**	**T5**	**T6**
Yproject Data: -	**Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6**								
7	1	0.26	0.07	0.35	-	-	-	0.00	-
8	2	-	-	-	0.10	0.23	0.06	-	0.07
9	3	0.21	0.11	0.26	-	-	-	0.08	-
10	4	-	-	-	0.14	0.22	0.08	-	0.00
11	5	0.00	0.26	0.50	-	-	-	0.15	-
12	6	-	-	-	0.00	0.03	0.19	-	0.14
Zproject Data: -	**Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6**								
13	1	-	-	-	0.10	0.03	0.15	-	0.07
14	2	0.14	0.03	0.12	-	-	-	0.00	-
15	3	-	-	-	0.13	0.08	0.29	-	0.05
16	4	0.18	0.02	0.10	-	-	-	0.07	-
17	5	0.09	0.13	0.26	-	-	-	0.00	-
18	6	-	-	-	0.00	0.03	0.19	-	0.10
Mean BRE		T1 = 0.15	T2a = 0.10	T2b = 0.27	T3 = 0.08	T4a = 0.10	T4b = 0.16	T5 = 0.05	T6 = 0.07
SP: Story Point **S#**: Sample Number **Spt No**: Sprint number **BRE**: Balanced Relative Error **T1, T3, T5, T6**: Treatment-1, 3, 5 and 6 while **T2a**, **T4a** are the treatment-2, 4 with PERT variant-1 and **T2b**, **T4b** treatment-2, 4 with PERT variant-2									

Table 9.6 BRE for the Effort Estimates of Yproject and Zproject – Treatments T1 to T6

The results of the BRE analysis disclose that Treatment T5 (i.e., Treatment T1 extended with COEEMO model) was the best performed method considered on the basis of Mean BRE when applied on Yproject and Zproject. However, Treatments T6 (i.e., Treatment T3 extended with COEEMO), and T3 (Planning Poker extended with expert judgement by following strictly the Bottom-Up approach) were second and third in order of performances on the basis of Mean BRE. The treatments (T2a, T4a, T3, T5, and T6) demonstrated better estimation accuracies as compared to Treatments (T1, T2b, and T4b) when applied on Yproject and Zproject.

BREbias for the Treatments: T1, T2a, T2b, T3, T4a, T4b, T5, and T6									
S#	Spt No	BREbias for the Treatments 1-6							
		T1	T2a	T2b	T3	T4a	T4b	T5	T6
Yproject Data: - Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6									
7	1	+0.26	-0.07	-0.35	-	-	-	0.00	
8	2	-	-	-	+0.10	+0.23	+0.06	-	-0.07
9	3	+0.21	-0.11	-0.26	-	-	-	+0.08	-
10	4	-	-	-	+0.14	+0.22	+0.08	-	0.00
11	5	0.00	-0.26	-0.50	-	-	-	-0.15	-
12	6	-	-	-	0.00	-0.03	-0.19	-	-0.10
Zproject Data: - Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6									
13	1	-	-	-	+0.10	-0.03	-0.15	-	-0.19
14	2	+0.14	+0.03	-0.12	-	-	-	0.00	-
15	3	-	-	-	+0.13	-0.08	-0.29	-	+0.05
16	4	+0.18	+0.02	-0.10	-	-	-	+0.07	-
17	5	+0.09	-0.13	-0.26	-	-	-	0.00	-
18	6	-	-	-	0.00	-0.03	-0.19	-	-0.10
SP: Story Point **S#**: Sample Number **Spt No**: Sprint number **BRE**: Balanced Relative Error **T1, T3, T5, T6**: Treatment-1, 3, 5 and 6 while **T2a**, **T4a** are the treatment-2, 4 with PERT variant-1 and **T2b**, **T4b** treatment-2, 4 with PERT variant-2									

Table 9.7 BREbias for the Effort Estimates of Yproject and Zproject – Treatments T1 to T6

Based on the calculations of BREbias, as shown in Table 9.7, the number of sprints with under-, over- and accurate effort estimations as proposed by the treatments when applied on Yproject and Zproject are counted and summarized in Table 9.8.

The results reveal that Treatment T5 resulted in precise accuracy in 3/6 sprints followed by treatments T3, T6, and T1, which achieved the level of the accuracy in 2/6, 1/6 and 1/6 sprints, respectively. The remaining treatments (i.e., T2a-b, and T4a-b) did not deliver any exact estimates.

Further, Treatment T1 produced the highest number of sprints with overestimated effort (i.e., 5/6 sprints) while Treatment T2b had the highest number of sprints with under-estimations of the effort (i.e., 6/6 sprints).

Treatment Name	Total Samples /Sprints	Number of Sprints		
		Over-estimated Effort	Under- estimated Effort	Accurate
T1	6	5	0	1
T2a	6	2	4	0
T2b	6	0	6	0
T3	6	4	0	2
T4a	6	2	4	0
T4b	6	2	4	0
T5	6	2	1	3
T6	6	1	4	1

Table 9.8 Treatments Vs Number of accurate / Over/underestimated Sprints of Yproject and Zproject

9.5.8.3 Accuracy Analysis for the Effort Estimates of All Projects — Treatments T1 to T6

Balanced Relative Error (BRE) and BREbias are calculated and presented in Tables 9.9 and 9.10, the latter table shows under/overestimated effort by the preceding negative and positive signs. The BRE and BREbias of the treatments are computed from the experimental data as given in Table 9.2, and the base formulae as shown in Figure 9.17.

BRE of the Effort Estimates per SP for the Treatments: T1, T2a, T2b, T3, T4a, T4b, T5, and T6									
S#	**Spt No**	**Balanced Relative Error for the Effort Estimates in PHs per SP for the Treatments T1 - T6**							
		T1	**T2a**	**T2b**	**T3**	**T4a**	**T4b**	**T5**	**T6**
Xproject Data: -	**Subjects/Treatments: Team-A, Agile coach / T1, T2, T5**								
1	27	0.16	0.09	0.06	-	-	-	0.03	-
2	28	0.07	0.10	0.07	-	-	-	0.00	-
3	29	0.11	0.05	0.07	-	-	-	0.02	-
4	30	0.37	0.25	0.14	-	-	-	0.25	-
5	31	0.00	0.15	0.33	-	-	-	0.10	-
6	32	0.27	0.12	0.02	-	-	-	0.16	-
Yproject Data: -	**Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6**								
7	1	0.26	0.07	0.35	-	-	-	0.00	-
8	2	-	-	-	0.10	0.23	0.06	-	0.07
9	3	0.21	0.11	0.26	-	-	-	0.08	-
10	4	-	-	-	0.14	0.22	0.08	-	0.00
11	5	0.00	0.26	0.50	-	-	-	0.15	-
12	6	-	-	-	0.00	0.03	0.19	-	0.14
Zproject Data: -	**Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6**								
13	1	-	-	-	0.10	0.03	0.15	-	0.07
14	2	0.14	0.03	0.12	-	-	-	0.00	-
15	3	-	-	-	0.13	0.08	0.29	-	0.05
16	4	0.18	0.02	0.10	-	-	-	0.07	-
17	5	0.09	0.13	0.26	-	-	-	0.00	-
18	6	-	-	-	0.00	0.03	0.19	-	0.10
Mean BRE		T1 = 0.16	T2a = 0.12	T2b = 0.19	T3 = 0.08	T4a = 0.10	T4b = 0.16	T5 = 0.07	T6 = 0.07
SP: Story Point **S#**: Sample Number **Spt No**: Sprint number **BRE**: Balanced Relative Error **T1, T3, T5, T6**: Treatment-1, 3, 5 and 6 while **T2a**, **T4a** are the treatment-2, 4 with PERT variant-1 and **T2b**, **T4b** treatment-2, 4 with PERT variant-2									

Table 9.9 BRE for the Effort Estimates of All Projects – Treatments 1 to 6

The results of the BRE analysis reveal that Treatments T5 (i.e., T1 extended with COEEMO model) and T3 (Planning Poker extended with expert judgement by following strictly the Bottom-Up approach) were the best performed of the eight methods considered on the basis of Mean BRE for all projects (i.e., Xproject, Yproject, and Zproject).

S#	Spt No	BREbias for the Treatments T1 - T6							
		T1	T2a	T2b	T3	T4a	T4b	T5	T6
Xproject Data: - Subjects/Treatments: Team-A, Agile coach / T1, T2, T5									
1	27	+0.16	+0.09	+0.06	-	-	-	+0.03	-
2	28	+0.07	+0.10	+0.07	-	-	-	0.00	-
3	29	+0.11	-0.05	-0.07	-	-	-	+0.02	-
4	30	+0.37	+0.25	+0.14	-	-	-	+0.25	-
5	31	+0.00	-0.15	-0.33	-	-	-	-0.10	-
6	32	+0.27	+0.12	-0.02	-	-	-	+0.16	-
Yproject Data: - Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6									
7	1	+0.26	-0.07	-0.35	-	-	-	0.00	
8	2	-	-	-	+0.10	+0.23	+0.06	-	-0.07
9	3	+0.21	-0.11	-0.26	-	-	-	+0.08	-
10	4	-	-	-	+0.14	+0.22	+0.08	-	0.00
11	5	0.00	-0.26	-0.50	-	-	-	-0.15	-
12	6	-	-	-	0.00	-0.03	-0.19	-	-0.10
Zproject Data: - Subjects/Treatments: Team-B, Agile coach / T1, T2, T3, T4, T5, T6									
13	1	-	-	-	+0.10	-0.03	-0.15	-	-0.19
14	2	+0.14	+0.03	-0.12	-	-	-	0.00	-
15	3	-	-	-	+0.13	-0.08	-0.29	-	+0.05
16	4	+0.18	+0.02	-0.10	-	-	-	+0.07	-
17	5	+0.09	-0.13	-0.26	-	-	-	0.00	-
18	6	-	-	-	0.00	-0.03	-0.19	-	-0.10

BREbias for the Treatments: T1, T2a, T2b, T3, T4a, T4b, T5, and T6

SP: Story Point **S#**: Sample Number **Spt No**: Sprint number **BRE**: Balanced Relative Error
T1, T3, T5, T6: Treatment-1, 3, 5 and 6 while **T2a**, **T4a** are the treatment-2, 4 with PERT variant-1 and **T2b**, **T4b** treatment-2, 4 with PERT variant-2

Table 9.10 BREbias for the Effort Estimates of All Projects – Treatments 1 to 6

Based on the calculations of BREbias for all projects, as shown in Table 9.10, the number of sprints with under-, over- and accurate effort estimations as proposed by all the treatments are counted and summarized in Table 9.11.

The results reveal that Treatment T5 resulted in precise accuracy in 5/12 sprints followed by treatments T1, T3, and T6, which achieved the same level of the accuracy in 2/12, 2/6 and 2/6 sprints, respectively. The remaining treatments (i.e., T2a-b, and T4a-b) did not deliver any exact estimates. Further, Treatment T1 produced the highest number of sprints with overestimated effort (i.e., 10/12 sprints) while Treatment T2b had the highest number of sprints with under-estimations of the effort (i.e., 9/12 sprints).

Treatment Name	Total Samples /Sprints	Number of Sprints		
		Over-estimated Effort	Under- estimated Effort	Accurate
T1	12	10	0	2
T2a	12	6	6	0
T2b	12	3	9	0
T3	06	4	0	2
T4a	06	2	4	0
T4b	06	2	4	0
T5	12	6	2	5
T6	06	1	4	1

Table 9.11 Treatments Vs Number of accurate / Over/underestimated Sprints of All Projects

9.6 Chapter Review

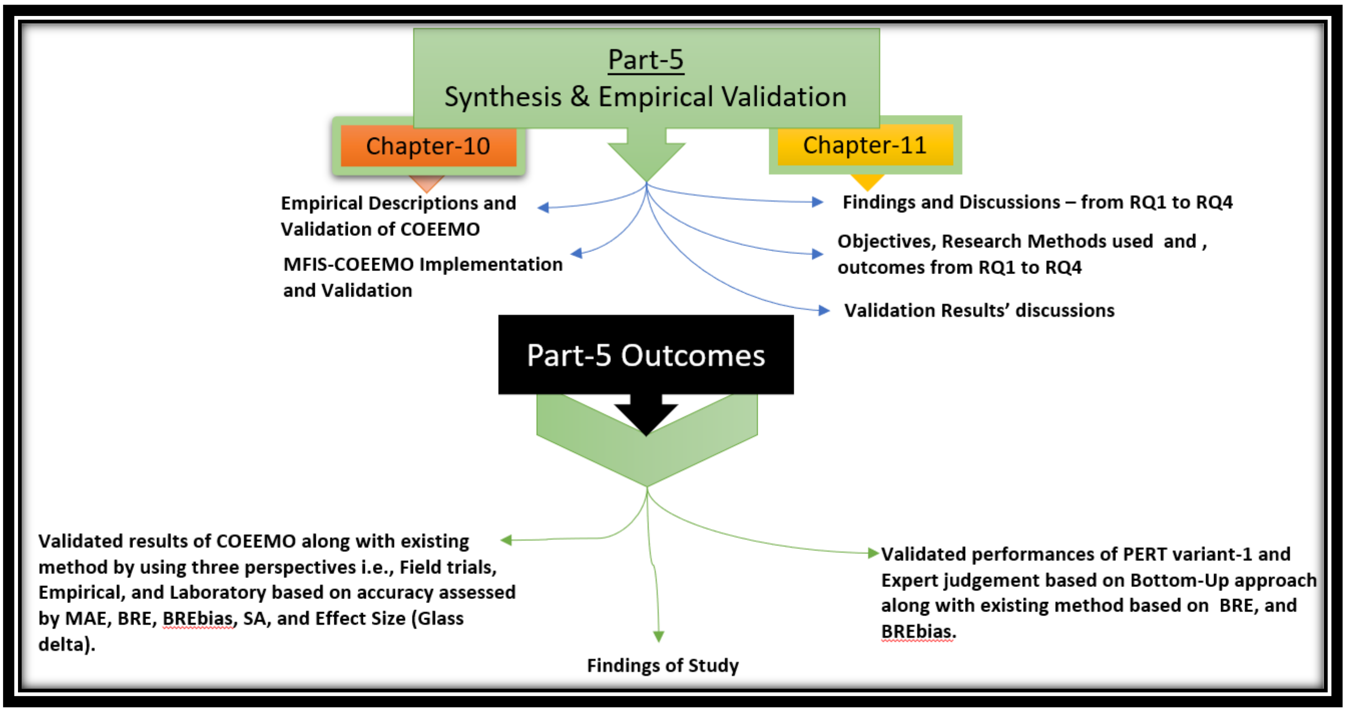
In this chapter, experimental data preparation, descriptive statistics, analysis, normality testing, and hypothesis evaluations were reported. Further, the effect of the various treatments on effort estimation and accuracy analysis was also considered.

In the next chapter, the details of further data collection and analysis from secondary sources, including publications and industry datasets in the public domain, are presented. Moreover, consideration regarding the COEEMO parameters and ratings based on the collected data is provided.

PART-5 SYNTHESIS AND EMPIRICAL VALIDATION

This part consists of two chapters which encompasses the validation of the proposed model based on three perspectives: i.e., empirical, field trials, and laboratory. Further, this reports the discussions and findings of the study.

Detailed views of the research activities with respect to the chapters within this part are presented in the figure as given below. On the top of the diagram, the chapters covered in this part are shown along with their different research endeavours as accomplished within the individual chapters, while the outcomes of Part-5 based on its individual chapters are exhibited at the bottom of the figure.



Chapter-10 Model Validation

***Contents' Summary**: This chapter reports the details of further data collection and analysis from secondary sources including publications and industry datasets in the public domain. There is further consideration of the COEEMO parameters and ratings based on the collected data. The chapter concludes with discussions and comparisons of the Balanced Relative Error (BRE) and BREbias for the accuracies of effort estimations with and without COEEMO and Mamdani Fuzzy inference.*

COEEMO (the **Co**nstructive **E**ffort **E**stimation **Mo**del) is the model put forward to answer RQs 3 & 4 and to deliver against one of the core objectives of this study. COEEMO is proposed as a comprehensive model that can be utilized to estimate/adjust human effort expended for software development projects in agile and non-agile environments. This model estimates and adjusts effort based on what have been shown empirically and in the research literature to be the most significant factors related to team, stakeholder, system, risk, organization's culture, work environment, and others, augmented where useful by traditional and/or contemporary estimation techniques (e.g., Expert judgement, PERT, Top-Down, Bottom-up, planning poker). These factors are considered in five core categories, i.e., Human, System, Organization, Sprint, and Risk. These categories are further divided into 3-4 subcategories which are themselves subdivided based on a scale of five-point fuzzy values ranging from "Very High" to "Very Low". Each fuzzy label is given an intuitively weighted value depending upon the significance of the envisioned factor. The complete list of these categories along with their detailed ratings and weighted values are provided in Table 4.1 of Chapter 4.

COEEMO has been designed to be used in agile and non-agile development environments by teams with different levels of professional experience and backgrounds ranging from well-established, new, migrated, or transitional teams. The high-level conceptual / functional view of this model aligned to the SCRUM framework is shown in Figure 4.1 of Chapter 4.

The validation process considered three different perspectives, i.e., field, empirical and laboratory, to determine the coverage and usefulness of the proposed COEEMO. From the first perspective, the COEEMO was employed in experimental settings on 3 real industry ASD projects, and its results were compared with those drawn from existing methods. Details of the process along with the outcomes of all the field trials were reported in Chapter 9.

The second perspective involves empirical validation in which the model was applied to previously completed ASD projects, and the results are reported in this chapter. To validate the model empirically, data on completed ASD projects were collected from two different sources: published studies and a repository of IT industry data (where the details of both these sources are covered in subsequent sections).

For the laboratory validation perspective, the COEEMO was implemented in MATLAB using the well-established fuzzy model (i.e., Mamdani Fuzzy Inference System – MFIS) which had been used in estimating the effort of numerous software development projects (Nisar et al., 2009; Attarzadeh and Hock, 2009; Muzaffar and Ahmed, 2010; Bathla et al., 2010; Lopez-Martin, 2011; Nassif et al., 2013). Further, the MFIS-based COEEMO and COEEMO alone were applied to the already completed projects and their results were compared.

For all three validation perspectives, two metrics (as shown in Figure 9.17) were used to measure the accuracy and direction of the effort estimates produced by each model/method, i.e., BRE and BREbias, where a model/method resulting in lower values of BRE was considered more accurate. The Mean BRE was calculated by adding up all the individual BREs, then divided by the number of BRE observations. Moreover, two further measures e.g., Standardized Accuracy (SA) and Effect size (Glass delta – Δ) as recommended by Shepperd and MacDonell (2012) were also utilized to test whether the prediction error of any proposed model or method was equivalent to random guessing, and further, how much improvement over random guessing was attained. The equations for both measures are provided in Figure 10.1.

$$SA = 1 - \frac{MAE}{\overline{MAE}_{po}}$$

$$\Delta = \frac{MAE - \overline{MAE}_{po}}{SP_o}$$

Figure 10. 1 Equations for Standardized Accuracy & Glass Delta (Shepperd and MacDonell, 2012)

In the above equations, MAE is the Mean Absolute Error which is calculated by the following formula:

$$\text{MAE} = \sum_{i=1}^{n} | e_i - \hat{e_i} | /n$$

Where:

- e_i, $\hat{e_i}$ are the actual and the estimated effort
- $\overline{MAEp_0}$ and SP_0 are the sample Mean and Standard deviation of random guessing, respectively
- Δ represents the Glass Effect size

10.1 Data of completed ASD projects gathered from published studies

Data relating to ASD projects and the significant factors of the COEEMO were collected from various published studies, as detailed below. While the data for some of the COEEMO factors were given in the referenced studies a few of them were missing; therefore, they were either deduced based on the nature of the reported data in the published study or the average of the maximum and the minimum ratings against each missing factor was used to avoid bias in the outcomes:

a. A comparative study (designated as study-A for reference purposes in the analysis and validation reporting) was conducted to estimate effort for an ASD project (Salmanoglu and Demirors, 2017). A big Turkish software contractor working in the security domain undertook the development of a Web Portal project by employing scrum in an agile development environment. In this study, 10 sprints as developed by 7 developers (i.e., 5 software developers and 2 testers) were evaluated. The relevant data of the sprints and COEEMO's factors as extracted from this published study are shown in Tables 10.1 (a & b) respectively.

ASD Project's Data for study-A

Spt No	TESPs	AFSPs	UFSPs	Total Actual Man-Hours Per sprint	Estimated Man-Hours per SP	Actual Man-Hours per SP
1	96	88	08	512	5.33	5.82
2	156	151	05	896	5.74	5.93
3	150	139	11	960	6.40	6.91
4	82	82	0	832	10.15	10.15
5	151	135	16	768	5.09	5.69
6	161	161	0	960	5.96	5.96
7	140	132	8	960	6.86	7.27
8	195	182	13	1024	5.25	5.63
9	196	175	21	1024	5.22	5.85
10	188	188	0	1152	6.13	6.13

TESPs: Total Estimated Story-Points **AFSPs**: Actual Finished Story-Points **UFSPs:** Unfinished Story-Points **Spt**: Sprint

(a)
ASD Project's Data of 10 Sprints for study-A (Salmanoglu and Demirors, 2017)

COEEMO factors' Data for study-A

Factors	Given / Assumed (Y/N)	Ratings	Remarks
Number of Team members	Y		This was a large Turkish company with domain expertise in security. This type of expertise normally was difficult to attain by inexperienced, less competent developers and teams who had the casual status of employment. Therefore, it was assumed that their team might have from moderate to very high levels of expertise and professional experience. Further, keeping in view the status of the contractor and the type of the project/employment, it was assumed that the developers might have very strong team relationships. Furthermore, the culture & the work environment were also considered in the strong category from moderate to very high-level. Moreover, it appeared to be a private company working in the security domain, and therefore, strong organization compliance might be observed
Team Members' Expertise	Y	0.36	
Team Members' Experience	Y	0.28	
Stakeholder Coordination	Y	0.35	
Team Relationship & Spirit	Y	1.06	
System Type	Y	Sys-4 Category	
System Complexity	Y	2.15	
Organization's Type	Y	0.33	
Organization's Ratings	Y	0.25	
Organization's Culture	Y	0.25	
Organization's Work Environment	Y	0.30	
Known mitigated risks	Y	2.3	
Known unmitigated risks		-0.4	
Unknown risks		-1.3	
Sprint Numbers	Y	1, 2, …,10	

			during software development from high to very high level. The complexity of the system was assumed very high because the web portal might have to incorporate some security aspects. Further, because of the security features, the impact of the project's risks was also thought to be from moderate to very high-level. Furthermore, the stakeholder coordination was assumed moderate in the Fully category.
Sprint(s) Complexity	Y	0.15,0.45, 0.59, 0.63, 0.68, 0.72, 0.77, 0.81, 0.86, 0.90	There were 10 sprints, the complexities for the sprints 1-3 were given while for sprints 4-10 they were calculated as per the given formula in subsection 4.4.5.1 of Chapter 4. Further, the complexities of the sprints were assumed moderate to avoid bias.
Note: CEAF, EF, TF, EnvF, and LF calculated as per method given in chapter-4 subsection 4.6.1.6			
TF: Technical Factor **EnvF**: Environmental Factor **EF**: Effort Factor **LF**: Learning Factor **CEAF:** COEEMO Effort Adjustment Factor			

(b)
COEEMO factors' Data for study-A

Table 10.1(a, b) ASD and COEEMO Factors' data for study-A

b. A case study (specified as study-B for reference purposes in the analysis and validation reporting) was undertaken in the academic year 2009/10 (over 15 weeks during the summer term) in the University of Ljubljana as a part of one of the courses for the capstone in software engineering (Mahnic, 2011). In this study 52 university students were divided into 13 groups of four to work as self-organized/managed Scrum teams to develop a Web-based system for student records. The system was required to have certain core modules to support: enrollment, applications / records for the exam, data maintenance, and a module to perform some statistical reviews. The professor who taught this course played the Product Owner role. The data for 39 project sprints (i.e., each agile team worked on the development over three sprints) and the COEEMO factors relevant as extricated from the published study are presented in Tables 10.2 (a, b, c, and d) respectively.

Planned and actual achievement in Sprint-1

Team	Velocity [Story Points]		Plan fulfillment [%]	Work spent [hours]	Hours worked per Story Point
	Planned	Actual			
T01	35.50	1.50	4.23	240.00	160.00
T02	30.50	11.00	36.07	141.00	12.82
T03	52.00	43.00	82.69	146.50	3.41
T04	13.00	13.00	100.00	74.00	5.69
T05	22.00	12.00	54.55	132.70	11.06
T06	23.00	8.00	34.78	127.00	15.88
T07	36.50	7.50	20.55	242.00	32.27
T08	14.00	5.00	35.71	103.00	20.60
T09	32.00	14.00	43.75	138.00	9.86
T10	25.00	6.00	24.00	126.50	21.08
T11	20.00	13.00	65.00	114.00	8.77
T12	25.00	7.00	28.00	134.50	19.21
T13	12.00	2.00	16.67	83.00	41.50

(a)
ASD project's Data for 13 teams for sprint-1 of study-B (Mahnic, 2011)

Planned and actual achievement in Sprint-2

Team	Velocity [Story Points]		Plan fulfillment [%]	Work spent [hours]	Hours worked per Story Point
	Planned	Actual			
T01	43.00	23.00	53.49	140.00	6.09
T02	46.00	31.00	67.39	255.00	8.23
T03	46.50	46.50	100.00	109.00	2.34
T04	20.00	12.50	62.50	160.00	12.80
T05	36.00	36.00	100.00	180.00	5.00
T06	33.50	23.00	68.66	170.50	7.41
T07	35.50	14.50	40.85	148.00	10.21
T08	36.00	35.00	97.22	181.00	5.17
T09	25.00	20.50	82.00	116.50	5.68
T10	40.00	26.00	65.00	137.50	5.29
T11	33.50	25.50	76.12	199.00	7.80
T12	33.50	21.50	64.18	155.00	7.21
T13	18.50	18.50	100.00	108.00	5.84

(b)

ASD project's Data of 13 teams for sprint-2 of study-B (Mahnic, 2011)

Planned and actual achievement in Sprint-3

Team	Velocity [Story Points]		Plan fulfillment [%]	Work spent [hours]	Hours worked per Story Point
	Planned	Actual			
T01	35.00	35.00	100.00	181.00	5.17
T02	34.50	31.50	91.30	175.00	5.56
T03	7.50	7.50	100.00	29.00	3.87
T04	15.00	14.00	93.33	94.00	6.71
T05	25.50	25.50	100.00	60.50	2.37
T06	36.50	30.50	83.56	116.10	3.81
T07	25.00	15.00	60.00	125.00	8.33
T08	20.00	20.00	100.00	98.00	4.90
T09	24.00	23.00	95.83	111.00	4.83
T10	29.50	22.50	76.27	93.00	4.13
T11	36.00	36.00	100.00	169.00	4.69
T12	23.50	23.50	100.00	106.00	4.51
T13	29.00	27.00	93.10	144.25	5.34

(c)

ASD project's Data of 13 teams for sprint-3 of study-B (Mahnic, 2011)

COEEMO factors' Data for the study-B

Factors	Given / Assumed (Y/N)	Ratings	Remarks
Number of Team members	Y	4	Detailed descriptions were provided in the study regarding various COEEMO factors meaning that many factor ratings could be found. To avoid biasedness, the moderate values of missing factors were utilized e.g., in risk factor. For the sprint complexity, it was assumed that the complexity for all the sprints was very high because all team members were students / learners without prior development experience. Further, every team had 4
Team Members' Expertise	Y	1.83	
Team Members' Experience	Y	2.4	
Stakeholder Coordination	Y	0.25	
Team Relationship & Spirit	Y	1.33 for other teams but 1.6 for the team T07	
System Type	Y	Sys-4 Category	
System Complexity	Y	2.15	
Organization's Type	Y	Joint Sector	
Organization's Type Ratings	Y	0.7	
Organization's Culture	Y	0.65	
Organization's Work Environment	Y	0.6	
Known mitigated risks	Y	2.5	

Known unmitigated risks		-0.3	members, and for equal distribution of the expertise and experience ratings across all the teams, two members from each team were considered as beginners in terms of experience and to have inadequate expertise with "Very High" ratings while the other 2 members at the same level of experience and expertise had "Very Low" ratings. For all teams, the relationships were assumed to be at the very high level within the "Good" category except for one team, i.e., team T07, for this team was considered at the very low level in the category of "Poor" as per the study indications.
Unknown risks		-1.1	
Sprint Numbers	Y	1, 2, 3	
Sprint(s) Complexity	Y	0.15, 0.45, 0.9	There was a total of 3 sprints and the 3rd sprint was the last sprint of the project; therefore, the SptNCpxy's rating of the sprint was considered the complexity for the sprint 3 as given in Table 4.1 of Chapter 4. Further, the complexities of the sprints were assumed moderate to avoid bias.
Note: CEAF, EF, TF, EnvF, and LF calculated as per method given in chapter-4 subsection 4.6.1.6			
TF: Technical Factor **EnvF**: Environmental Factor **EF:** Effort Factor **LF**: Learning Factor **CEAF:** COEEMO Effort Adjustment Factor			

(d)
COEEMO factors' Data of 39 sprints for the study-B

Table 10. 2 (a, b, c, d) ASD and COEEMO Factors' data for study-B

10.2 Data of completed ASD projects collected from the IT industry

Data relating to eight completed ASD projects were collected from two Pakistani-based medium-sized IT companies (i.e., ICT-1 and ICT-2) working in agile software product development contexts. The COEEMO's factors as identified for all eight projects were rated by the companies involved (using sample excel sheet/templates and guidelines provided to the organizations attached at Appendix-E) as shown in Tables 10.3, 10.4, and 10.5 respectively. The real names of the projects and the companies were not used here to comply with the rules of the AUT ethics committee (AUTEC). The eight software projects / products were named as SPA, SPB, SPC, SPD, SPE, SPF, SPG, and SPH.

Data of Projects from IT Industries / Companies										
P#	**PID**	**ITCID**	**Spt No**	**US**	**TESPs**	**FSPs**	**UFSPs**	**TMs**	**Spt Duration (weeks)**	**Person-Hours**
1	SPA	ITC-1	1	4	14	11	3	1	2	63
			2	5	22	19	3			
			3	3	6	4	2			
2	SPB		1	3	7	5	2	2	2	126
			2	3	9	8	1			
3	SPC		1	4	14	11	3	4	2	252
			2	4	11	9	2			
			3	5	19	11	8			
			4	5	18	15	3			
4	SPD		1	6	23	21	2	3	2	189
			2	5	19	18	1			
			3	8	32	31	1			
5	SPE	ITC-2	1	7	15	11	4	2	2	140
			2	13	27	21	6	3		210
			3	25	58	56	2			
			4	34	64	60	4			
			5	24	54	46	8			
			6	15	32	30	2	4		280
			7	19	42	38	4			
			8	19	39	37	2			
			9	22	58	56	2			
			10	32	57	37	20	3		210
			11	32	47	27	20			
			12	17	37	31	6			
			13	19	38	26	12			
			14	9	21	19	2	1		70
	SPF		1	5	17	12	5	4	2	200
			2	4	15	12	3			
			3	7	22	15	7			
			4	6	23	15	8			
	SPG		1	9	39	30	9	4		240
			2	10	61	46	15			
			3	6	34	29	5			
	SPH		1	5	14	9	5	4		200
			2	7	36	28	8			
			3	7	59	41	18			

P#: Project Number **ITCID**: IT company ID **Spt No**: Sprint number **TMs**: Team members **US**: User Stories
TESPs: Total Estimated Story-Points **FSPs**: Finished Story-Points **UFSPs:** Unfinished Story-Points **Spt**: Sprint

Table 10 3 ASD Projects' Data from IT Companies

The data regarding one of the ITC-2 products (SPE) was incomplete. SPE was a web-based product developed in 36 sprints (where each sprint duration was 2 weeks) but the data of only 14 sprints were presented here, due to some technical issues and/or missing significant information: total estimated SPs, finished and unfinished SPs, team members who developed/estimated the stories/features along with their expertise and the experience details, and others. The team lead was consulted to include any missing information that was significant for the validation.

COEEMO Factors' Data for the ITC-1 Projects i.e., SPA, SPB, SPC, and SPD:		
COEEMO's Factors	**Factors' Levels / Ratings**	**Values as per Table 4.1 of Chapter 4**
Team Members' Expertise	1 Member of "Medium expertise" with "Very High" rating (worked for SPA, SPB, SPC, SPD projects)	1.4
	1 Member of "Adequate expertise" with "High" rating (worked for SPB, SPC, SPD projects)	0.74
	1 Member of "Inadequate expertise" with "High" rating (Worked for SPC project)	1.99
	1 Member of "High expertise" with "Very High" rating (Worked for SPC, and SPD projects)	0.3
Team Expertise - calculated as per method given in Chapter-4 subsection 4.6.1.1		
Team Members' Experience	1 Members of "Experienced level" with "Very High" rating (worked for SPA, SPB, SPC, SPD projects)	0.8
	1 Members of "Experienced level" with "High" rating (Worked for SPB, SPC, and SPD projects)	0.85
	1 Member of "Beginner level" with "Low" rating (Worked for SPC projects)	2.4
	1 Members of "Proficient level" with "Very High" rating (Worked for SPC and SPD projects)	0.2
Team Experience **-** calculated as per method given in Chapter-4 subsection 4.6.1.2		
Stakeholder Coordination	3 Stakeholders are "Fully" cooperative with "High" rating for SPA project	0.3
	5 Stakeholders are "Poorly" cooperative with "Moderate" rating for SPB project	1.85
	4 Stakeholders are "Occasionally" cooperative with "Very High" rating for SPC project	1.15
	2 Stakeholder are "Poorly" cooperative with "Low" rating for SPD project	1.5
Stakeholder Coordination **-** calculated as per method given in Chapter-4 subsection 4.6.1.3		
Team Relationship & Spirit	"Excellent" with "High" rating for SPA project "Good" with "Moderate" rating for SPB project "Excellent" with "Very High" rating for SPC project "Good" with "Moderate" rating for SPD project	1.06 1.39 1.03 1.39
System Complexity	"Sys-4" with complexity "Moderate for SPA", "High for SPB", "Low for SPC" "Sys-2" with complexity "Very Low for SPD"	2.35 for SPA 2.25 for SPB 2.45 for SPC 1.4 for SPD
Organization's Type	Organization type is "Private" with "Moderate" rating	0.4
Organization's Culture	Organization culture is "Strong" with "Moderate" rating	0.35
Organization's Work Environment	Work environment is "Best" with "Moderate" rating	0.35
Project's Risks	Known Planned risks "Moderate for SPA", "Low for SPB, SPC, and SPD"	2.5, 2.7
	Known Unplanned risks "Moderate for SPA, SPB, SPC" and "Very High for SPD"	-0.3, -0.5
	Unknown risks "Low" for SPA, "Moderate for SPB, SPC" and "Very High" for SPD grading	-0.9, -1.1, -1.5
Sprint(s)-wise Story Points (SPs) – CEAF, EF, TF, EnvF, LF calculated as per method given in chapter-4 subsection 4.6.1.6		
Sprint(s) Complexity	Spt 1-3 "High for SPA", Spt 1-2 "Moderate" for SPB, Spt 1, 3 "High" and Spt 2,4 "Moderate for SPC, Spt 1 "High" and Spt 2-3 "Low" for SPD	0.12, 0.4, 0.85 for SPA, 0.15, 0.45 for SPB, 0.12, 0.45, 0.79 and 0.9 for SPC, 0.12, 0.5, and 0.94 for SPD
Note: CEAF, EF, TF, EnvF, and LF calculated as per method given in chapter-4 subsection 4.6.1.6		
TF: Technical Factor **EnvF**: Environmental Factor **EF:** Effort Factor **LF**: Learning Factor **CEAF:** COEEMO Effort Adjustment Factor		

Table 10.4 COEEMO Factors' Data for the ITC-1 Projects – SPA, SPB, SPC, SPD

COEEMO Factors' Data for the ITC-2 Projects i.e., SPE, SPF, SPG, SPH		
COEEMO's Factors	**Factors' Levels / Ratings**	**Values as per Table 4.1 of Chapter 4**
Team Members' Expertise		
Project: SPE	1 Member of "Medium expertise" with "Very High" 2 Members of "High expertise" with "Very High" rating 1 Member of "High expertise" with "High" rating	1.4 0.3 0.34
Project: SPF	1 Member of "Adequate expertise" with "Moderate" rating 2 Members of "Medium expertise" with "Moderate" and "Very Low" ratings 1 Member of "Inadequate expertise" with "Very High" rating	0.82 1.48, 1.7 1.95
Project: SPG	1 Member of "Adequate expertise" with "High" rating 2 Members of "Medium expertise" with "Moderate" ratings 1 Member of "Inadequate expertise" with "Moderate" rating	0.74 1.48 2.07
Project: SPH	3 Members of "Medium expertise" with "Very Low" ratings 1 Member of "Adequate expertise" with "Moderate" rating	1.7 0.82
Team Expertise - calculated as per method given in Chapter-3 subsection 4.6.1.1		
Team Members' Experience		
Project: SPE	3 Members of "Experienced level" with "Very High" ratings 1 Member of "Intermediate level" with "High" rating	0.8 1.6
Project: SPF	3 Members of "Intermediate level" with "Moderate" Ratings 1 Member of "Beginner level" with "High" rating	1.7 2.1
Project: SPG	1 Member of "Experienced level" with "Moderate" Ratings 2 Members of "Intermediate level" with "Moderate" Ratings 1 Member of "Beginner level" with "Very High" rating	0.95 1.7 2.1
Project: SPH	1 Member of "Experienced level" with "Moderate" Rating 2 Members of "Intermediate level" with "Very Low" Ratings 1 Member of "Intermediate level" with "Very High" Rating	0.95 2.05 1.55
Team Experience - calculated as per method given in Chapter-4 subsection 4.6.1.2		
Stakeholder Coordination	2 Stakeholder "Fully" cooperative with "Very High" rating for SPE 1 Stakeholder "Fully" with "Moderate" rating (Project: SPF) 1 Stakeholder "Partially" with "Very High" rating (Project: SPG) 1 Stakeholder "Partially" with "Very Low" rating (Project: SPH)	0.25 0.35 0.6 0.8
Stakeholder Coordination - calculated as per method given in Chapter-4 subsection 4.6.1.3		
Team Relationship & Spirit	"Very Good" with "Very High" rating (Project: SPE) "Good" with "Low" rating (Project: SPF, SPG and SPH)	1.18 1.42
System Complexity	Category "Sys-4" with complexity "Moderate" rating Category "Sys-4" with complexity "Very High" rating (Projects: SPF, and SPG) Category "Sys-2" with complexity "Very High" rating (Project: SPH)	2.35 2.15 0.8
Organization's Type	Organization type is "Private" with "Moderate" rating for all projects	0.4
Organization's Culture	Organization culture is "Semi-Strong" with "Moderate" rating for all projects	0.85
Organization's Work Environment	Work environment is "Semi-Best" with "Moderate" rating for all projects	0.7
Project's Risks	Known Planned risks "High to Moderate" (Project: SPE) Known Planned risks "Very High" (Project: SPF, SPG, SPH)	2.4 2.1
	Known Unplanned risks "Very High" for all projects	-0.5
	Unknown risks "Very High" for all projects	-1.5
Sprint(s) Complexity	The sprints numbers which were used for SPE were not representing the true order of the sprints in the actual project (this product was developed in 36 sprints) therefore to calculate the complexity of each sprint the "High" SptNCpxy factor i.e., 0.85 was considered. While this value of SptNCpxy and 0.4 which was the value of "High" complexity for the sprint# 2 as given in the table 4.1 of Chapter 4 against the selected SptNCpxy were utilized by the formula to calculate the sprints' complexities (this formula was demonstrated in subsection 4.4.5.1 of Chapter 4).	0.85 (SptNCpxy)
	Spt# 1-3 "Very High Complexity for SPF, SPG, SPH and Spt# 4 "Very High" for SPF which was calculated based on SptNCpxy i.e., 1 as per the formula given in subsection 4.4.5.1 of Chapter 4.	0.1, 0.3 (SPF, SPG, SPH for Spt 1-2) 0.75 for Spt-3 SPG, and SPH
Note: CEAF, EF, TF, EnvF, and LF calculated as per method given in chapter-4 subsection 4.6.1.6		
TF: Technical Factor **EnvF**: Environmental Factor **EF:** Effort Factor **LF**: Learning Factor **CEAF:** COEEMO Effort Adjustment Factor		

Table 10.5 COEEMO Factors' Data for the ITC-2 Projects – SPE, SPF, SPG, SPH

10.3 Mamdani Fuzzy Inference System (MFIS)

The relevant MATLAB toolbox was employed to implement COEEMO through a Mamdani Fuzzy Inference System (MFIS). Gaussian and Triangular memberships functions (MFs) were utilized to present input and output variables of the MFIS, while 12 Fuzzy rules as given in Figure 10.3 were devised to make the MFIS functional. Gaussian and Triangular MFs were chosen based on their superior performance as compared to other types of MF (Sadollah, 2018).

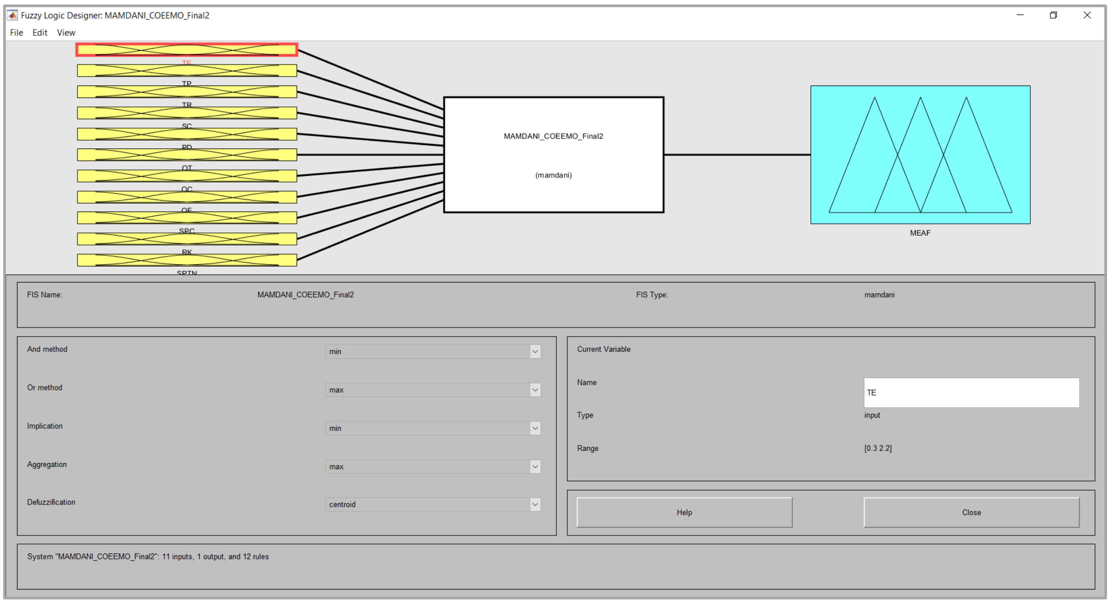


Figure 10. 2 Fuzzy Logic Designer Interface – MFIS-COEEMO

The detailed MFIS algorithm that realized the COEEMO is given at Appendix-F while screen shots of the MFIS (i.e., Fuzzy Logic Designer, MFs Rules editor, surface, and Rule viewers) are provided in Figures 10.2, 10.3., and 10.4 respectively.

S#	COEEMO's Factors / Metrics	Represented in MFIS	Values in Ranges [Min – Max]	No. / Type of Membership Functions / DE-Fuzzified Method
Input Variables				
1	Team Expertise	TE	[0.3 – 2.2]	5 Gaussian membership functions are considered against each input variable which includes: VH – Very High, H – High; M – Moderate, L – Low, and VL – Very Low.
2	Team Experience	TP	[0.2 – 2.6]	
3	Team Relationship & Spirit	TR	[1.03 – 1.6]	
4	Stakeholder Coordination	SC	[0.25 – 1.7]	
5	Org. Culture	OC	[0.15 – 2.05]	
6	Org. Work Environment	OE	[0.25 – 1.5]	
7	Org. Type	OT	[0.3 – 1.1]	
8	System Types' Complexity	PD	[0.1 – 2.5]	
9	Sprint's Complexity**	SPC	[0.1 – 0.98]	De-Fuzzified Method Name: Centroid
10	Project's Risks***	RK	[-1.5 – 2.9]	
Output Variable				
1	Effort Adjustment Factor	MEAF	[0.35 – 1.05]	9 Triangulated membership functions are envisaged against an output variable which includes: VVL – Very Very Low, VL – Very Low, L – Low, M – Moderate, H – High, VH – Very High, VVH – Very Very High, EH – Extra High, and XH – Extreme High

**Sprint's complexity (SPC) was given based on the sprint number from 1-3 or more. To adjust the sprint number in the MFIS, SPTN was used as an input variable for MFIS and therefore, three ranges (0 – 0.95, 0.95 – 1.95, 1.95 – 2.99) were used for sprint-1, 2, and 3 respectively to calculate MEAF. If any project / product is completed in more than 3 sprints, then sprints are equally divided into these given ranges of SPTN, the MEAF was calculated based on each sprint aligned to its own range. The value of MEAF was chosen which is highest within that relevant ranges of the SPTN.
***Project risks' ratings (i.e., Known mitigated, known non-mitigated, and unknown) were aggregated first and then given to MFIS for MEAF calculations.

Table 10. 6 MFIS-COEEMO Input / Output Variables / Membership Functions

The Gaussian input and triangular output variable representations along with their values in ranges are shown in Table 10.6. These variables reflect the 10 most significant factors of the COEEMO. For the purpose of the implementation of a MFIS based on COEEMO factors / ratings, the names of the COEEMO's factors' subcategories were ignored and only their ratings / values in ranges (i.e., minimum and maximum) were contemplated.

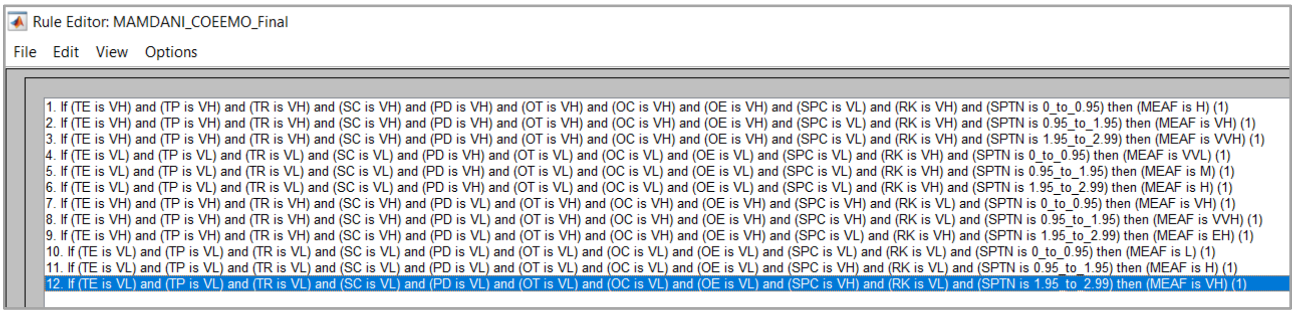


Figure 10. 3 Fuzzy Rules for MFIS-COEEMO

The data of the ASD projects as reported in Tables 10.1 to 10.3 were given as Fuzzy sets to the implemented MFIS to calculate the MEAF (MFIS Effort Adjustment Factor) by using the Rule viewer of the Fuzzy designer, as presented in Figure 10.5. The MFIS generated MEAF and further calculations of the total estimated SPs as proposed by the MFIS are reported in Tables 10.5 to 10.7 for the purpose of empirical evaluation and comparison.

Figure 10. 4 MFIS-COEEMO Ruler

10.4 Effort Estimates' Accuracy Analysis

The critical data for the effort estimates as projected by the team, the COEEMO and the Mamdani FIS (MFIS) are given in Table 10.7 (for the case study-A), 10.8 to 10.9 (for the case study-B), and 10.10 to 10.11 (for the industry data). These report accuracy comparisons in the effort estimates using the two important metrics BRE and BREbias. The details of each are considered in the following subsections.

10.4.1 Effort Estimates' Accuracy Analysis for the ASD projects of Case Study-A

Tables 10.7 (a, b) set out the COEEMO and MFIS suggested TESPs (i.e., the product of the COEEMO / MFIS effort adjustment factor and the team's TESPs per sprint) along with the effort estimates, BRE and BREbias. The effort adjustment factors (i.e., CEAFs and MEAFs) for the 10 sprints were calculated by using the COEEMO and MFIS. The COEEMO parameters along with their relevant gradings / ratings as identified from the case study-A are shown in Tables 10.1 (a-b), and these were utilized as inputs to both models.

The results demonstrated that the COEEMO outperformed the other approaches in 4 sprints when the team's estimates were adjusted by the CEAF, while the team alone and the MFIS estimates were better in 4 and 2 sprints respectively. In one sprint the estimations by the MFIS and the COEEMO were the same. The mean BRE for the accuracy of the COEEMO's effort estimates (i.e., 5%) was better than the team and MFIS's MBREs (i.e., 6% and 8%).

Project	[1]Spt #	[2]TESPs	[3]AFSPs	[4]CEAF	[5]CTESPs	[6]MEAF	[7]MTESPs
Case Study-A	1	96	88	0.86	83	0.74	71
	2	156	151	0.89	139	0.83	129
	3	150	139	0.91	137	0.83	125
	4	82	82	0.92	75	0.96	79
	5	151	135	0.93	140	0.96	145
	6	161	161	0.94	151	0.96	155
	7	140	132	0.95	133	0.96	134
	8	195	182	0.96	187	0.96	187
	9	196	175	0.97	190	0.96	188
	10	188	188	0.99	186	0.96	180

1: Sprint Number 2: Total Estimated Story-Points (SPs) 3: Actual Finished SPs 4: COEEMO Effort Adjustment Factor
5: COEEMO Total Estimated SPs 6: Mamdani FIS Effort Adjustment Factor 7: Mamdani FIS Total Estimated SPs

(a)
Team, COEEMO, and MFIS suggested TESPs for the Case Study-A

Project	[1]Spt #	[2]TEE /SP	[3]CEE /SP	[4]MEE /SP	[5]AE /SP	[6]BRE			[7]BREbias		
						TEE	CEE	MEE	TEE	CEE	MEE
Case Study-A	1	5.33	6.20	7.21	5.82	0.09	0.07	0.24	+0.09	-0.07	-0.24
	2	5.74	6.45	6.92	5.93	0.03	0.09	0.17	+0.03	-0.09	-0.17
	3	6.40	7.03	7.71	6.91	0.08	0.02	0.12	+0.08	-0.02	-0.12
	4	10.15	11.03	10.57	10.15	0.00	0.09	0.04	0.00	-0.09	-0.04
	5	5.09	5.47	5.30	5.69	0.12	0.04	0.07	+0.12	+0.04	+0.07
	6	5.96	6.34	6.21	5.96	0.00	0.06	0.04	0.00	-0.06	-0.04
	7	6.86	7.22	7.14	7.27	0.06	0.01	0.02	+0.06	+0.01	+0.02
	8	5.25	5.47	5.47	5.63	0.07	0.03	0.03	+0.07	+0.03	+0.03
	9	5.22	5.39	5.44	5.85	0.12	0.09	0.08	+0.12	+0.09	+0.08
	10	6.13	6.19	6.38	6.13	0.00	0.01	0.04	0.00	-0.01	-0.04
					MBRE	0.06	0.05	0.08			

1: Sprint Number 2: Team Estimated Effort per SP 3: COEEMO Estimated Effort per SP 4: Mamdani FIS Estimated Effort per SP
5: Actual Effort per SP 6: Balanced Relative Error 7: Balanced Relative Error Bias

(b)
Team, COEEMO, and MFIS suggested Effort Estimates / BRE for the Case Study-A

Table 10.7 Team, COEEMO, MFIS suggested TESPs, Effort Estimates / BRE for Case Study-A

10.4.2 Effort Estimates' Accuracy Analysis for the ASD projects of Case Study-B

Tables 10.8 and 10.9 show the COEEMO / MFIS recommended TESP values (i.e., the product of the COEEMO / MFIS effort adjustment factor and the teams' TESPs for sprint-1, 2 and 3) along with the effort estimates, BRE and BREbias individually. The effort adjustment factors (CEAF / MEAF) for the three sprints were calculated by using the COEEMO and MFIS. The COEEMO parameters along with their relevant gradings / ratings, as identified from the case study-B, were displayed in Tables 10.2 (a-d) and were the inputs for both models COEEMO and MFIS.

The findings for sprint-1 and 2 indicated that COEEMO and MFIS outperformed the effort estimates of 21 and 20 teams, respectively, after adjusting their estimates by using the CEAF / MEAF, as compared to the teams' own estimates without the CEAF / MEAF. However, the estimation results for the teams, COEEMO and MFIS were similar in sprint-3. The MBREs for the accuracies of the teams' estimations from sprint 1-3 are 3.51, 0.42, and 0.11 respectively as compared to 1.91, 0.25 and 0.10 in the case of COEEMO. MFIS resulted in higher MBRE values than those of COEEMO (i.e., 2.40 Vs 1.91, 0.31 Vs 0.25 for sprint 1 and 2). The overall improvements in effort estimate accuracy were approximately 44% and 30% when the teams' provided estimates were adjusted by using the COEEMO and MFIS effort adjustment factors (i.e., CEAF and MEAF) individually.

Project	[*]TID	[1]Sp#	[2]TESPs	[3]AFSPs	[4]CEAF	[5]CTESPs	[6]MEAF	[7]MTESPs
Case Study-B	T01		35.5	1.5	0.62	22.0	0.75	26.6
	T02		30.5	11	0.62	18.9	0.75	22.9
	T03		52	43	0.62	32.2	0.75	39.0
	T04		13	13	0.62	8.1	0.75	9.8
	T05		22	12	0.62	13.6	0.75	16.5
	T06		23	8	0.62	14.3	0.75	17.3
	T07	1	36.5	7.5	0.59	21.5	0.65	23.7
	T08		14	5	0.62	8.7	0.75	10.5
	T09		32	14	0.62	19.8	0.75	24.0
	T10		25	6	0.62	15.5	0.75	18.8
	T11		20	13	0.62	12.4	0.75	15.0
	T12		25	7	0.62	15.5	0.75	18.8
	T13		12	2	0.62	7.4	0.75	9.0
	T01		43	23	0.74	31.8	0.88	37.8
	T02		46	31	0.74	34.0	0.88	40.5
	T03		46.5	46.5	0.74	34.4	0.88	40.9
	T04		20	12.5	0.74	14.8	0.88	17.6
	T05		36	36	0.74	26.6	0.88	31.7
	T06		33.5	23	0.74	24.8	0.88	29.5
	T07		35.5	14.5	0.72	25.6	0.79	28.0
	T08		36	35	0.74	26.6	0.88	31.7
	T09		25	20.5	0.74	18.5	0.88	22.0
	T10	2	40	26	0.74	29.6	0.88	35.2
	T11		33.5	25.5	0.74	24.8	0.88	29.5
	T12		33.5	21.5	0.74	24.8	0.88	29.5
	T13		18.5	18.5	0.74	13.7	0.88	16.3
	T01		35	35	0.96	33.6	0.92	32.2
	T02		34.5	31.5	0.96	33.1	0.92	31.7
	T03		7.5	7.5	0.96	7.2	0.92	6.9
	T04		15	14	0.96	14.4	0.92	13.8
	T05		25.5	25.5	0.96	24.5	0.92	23.5
	T06		36.5	30.5	0.96	35.0	0.92	33.6
	T07	3	25	15	0.95	23.8	0.83	20.8
	T08		20	20	0.96	19.2	0.92	18.4
	T09		24	23	0.96	23.0	0.92	22.1
	T10		29.5	22.5	0.96	28.3	0.92	27.1
	T11		36	36	0.96	34.6	0.92	33.1
	T12		23.5	23.5	0.96	22.6	0.92	21.6
	T13		29	27	0.96	27.8	0.92	26.7

1: Sprint Number **2: Total Estimated Story-Points (SPs)** **3: Actual Finished SPs** **4: COEEMO Effort Adjustment Factor**
5: COEEMO Total Estimated SPs **6: Mamdani FIS Effort Adjustment Factor** **7: Mamdani FIS Total Estimated SPs** *** Team-ID**

Table 10. 8 Team, COEEMO, and MFIS suggested TESPs for the Case Study-B

Project	*TID	[1]Sp #	[2]TEE /SP	[3]CEE /SP	[4]MEE /SP	[5]AE /SP	[6]BRE			[7]BREbias		
							TEE	CEE	MEE	TEE	CEE	MEE
Case Study-B	T01	1	6.76	10.90	9.01	160	22.67	13.67	16.75	+22.67	+13.67	+16.75
	T02		4.62	7.46	6.16	12.82	1.77	0.72	1.08	+1.77	+0.72	+1.08
	T03		2.82	4.54	3.76	3.41	0.21	0.33	0.10	+0.21	-0.33	-0.10
	T04		5.69	9.18	7.59	5.69	0.00	0.61	0.33	0.00	-0.61	-0.33
	T05		6.03	9.73	8.04	11.06	0.83	0.14	0.38	+0.83	+0.14	+0.38
	T06		5.52	8.91	7.36	15.88	1.88	0.78	1.16	+1.88	+0.78	+1.16
	T07		6.63	11.24	10.20	32.27	3.87	1.87	2.16	+3.87	+1.87	+2.16
	T08		7.36	11.87	9.81	20.60	1.80	0.74	1.10	+1.80	+0.74	+1.10
	T09		4.31	6.96	5.75	9.86	1.29	0.42	0.71	+1.29	+0.42	+0.71
	T10		5.06	8.16	6.75	21.08	3.17	1.58	2.13	+3.17	+1.58	+2.13
	T11		5.70	9.19	7.60	8.77	0.54	0.05	0.15	+0.54	-0.05	+0.15
	T12		5.38	8.68	7.17	19.21	2.57	1.21	1.68	+2.57	+1.21	+1.68
	T13		6.92	11.16	9.22	41.50	5.00	2.72	3.50	+5.00	+2.72	+3.50
						Sprint-1 MBRE	**3.51**	**1.91**	**2.40**			
	T01	2	3.26	4.40	3.70	6.09	0.87	0.38	0.65	+0.87	+0.38	+0.65
	T02		5.54	7.49	6.30	8.23	0.48	0.10	0.31	+0.48	+0.10	+0.31
	T03		2.34	3.17	2.66	2.34	0.00	0.35	0.14	+0.00	-0.35	-0.14
	T04		8.00	10.81	9.09	12.80	0.60	0.18	0.41	+0.60	+0.18	+0.41
	T05		5.00	6.76	5.68	5.00	0.00	0.35	0.14	+0.00	-0.35	-0.14
	T06		5.09	6.88	5.78	7.41	0.46	0.08	0.28	+0.46	+0.08	+0.28
	T07		4.17	5.79	5.28	10.21	1.45	0.76	0.93	+1.45	+0.76	+0.93
	T08		5.03	6.79	5.71	5.17	0.03	0.31	0.10	+0.03	-0.31	-0.10
	T09		4.66	6.30	5.30	5.68	0.22	0.11	0.07	+0.22	-0.11	+0.07
	T10		3.44	4.65	3.91	5.29	0.54	0.14	0.35	+0.54	+0.14	+0.35
	T11		5.94	8.03	6.75	7.80	0.31	0.03	0.16	+0.31	-0.03	+0.16
	T12		4.63	6.25	5.26	7.21	0.56	0.15	0.37	+0.56	+0.15	+0.37
	T13		5.84	7.89	6.63	5.84	0.00	0.35	0.14	0.00	-0.35	-0.14
						Sprint-2 MBRE	**0.42**	**0.25**	**0.31**			
	T01	3	5.17	5.39	5.62	5.17	0.00	0.04	0.09	0.00	-0.04	-0.09
	T02		5.07	5.28	5.51	5.56	0.10	0.05	0.01	+0.10	+0.05	+0.01
	T03		3.87	4.03	4.20	3.87	0.00	0.04	0.09	0.00	-0.04	-0.09
	T04		6.27	6.53	6.81	6.71	0.07	0.03	0.01	+0.07	+0.03	-0.01
	T05		2.37	2.47	2.58	2.37	0.00	0.04	0.09	0.00	-0.04	-0.09
	T06		3.18	3.31	3.46	3.81	0.20	0.15	0.10	+0.20	+0.15	+0.10
	T07		5.00	5.26	6.02	8.33	0.67	0.58	0.38	+0.67	+0.58	+0.38
	T08		4.90	5.10	5.33	4.90	0.00	0.04	0.09	+0.00	-0.04	-0.09
	T09		4.63	4.82	5.03	4.83	0.04	0.00	0.04	+0.04	0.00	-0.04
	T10		3.15	3.28	3.43	4.13	0.31	0.26	0.21	+0.31	+0.26	+0.21
	T11		4.69	4.89	5.10	4.69	0.00	0.04	0.09	0.00	-0.04	-0.09
	T12		4.51	4.70	4.90	4.51	0.00	0.04	0.09	0.00	-0.04	-0.09
	T13		4.97	5.18	5.41	5.34	0.07	0.03	0.01	+0.07	+0.03	-0.01
						Sprint-3 MBRE	**0.11**	**0.10**	**0.10**			

1: Sprint Number **2: Team Estimated Effort per SP** **3: COEEMO Estimated Effort per SP** **4: Mamdani FIS Estimated Effort per SP**
5: Actual Effort per SP **6: Balanced Relative Error** **7: Balanced Relative Error Bias**

Table 10.9 Team, COEEMO, and MFIS suggested Effort estimates / BRE for the Case Study-B

10.4.3 Effort Estimates' Accuracy Analysis for the ASD projects from industry

Tables 10.10 and 10.11 report the COEEMO and MFIS proposed TESP values (i.e., the product of the COEEMO / MFIS effort adjustment factors and the teams' TESPs for the eight industrial projects, comprising 36 sprints altogether) along with the effort estimates, BRE and BREbias values respectively. The effort adjustment factors (i.e., CEAF and MEAF) for the 36 sprints of the projects were calculated by using the COEEMO and MFIS. The COEEMO parameters, along with their relevant gradings / ratings as identified from the industry projects / organizations, were presented in Tables 10.3 to 10.5, and were utilized as inputs to both models, COEEMO and MFIS. The findings showed that the teams' estimates for 28 sprints out of 36 were more accurate when their estimates were adjusted by using the CEAF and MEAF, as compared to the teams' estimates without CEAF and MEAF.

Further, the MBRE values for the accuracies of the teams' estimates adjusted with CEAF and MEAF were 29% and 37% individually, as compared to 49% without the stated adjustments.

Projects	[1]Spt#	[2]TESPs	[3]AFSPs	[4]CEAF	[5]CTESPs	[6]MEAF	[7]MTESPs
SPA	1	14	11	0.79	11	0.72	10
	2	22	19	0.85	19	0.94	21
	3	6	4	0.97	6	0.96	6
SPB	1	7	5	0.74	5	0.53	4
	2	9	8	0.84	8	0.74	7
SPC	1	14	11	0.76	11	0.72	10
	2	11	9	0.83	9	0.83	9
	3	19	11	0.91	17	0.88	17
	4	18	15	0.96	17	0.96	17
SPD	1	23	21	0.74	17	0.74	17
	2	19	18	0.84	16	0.88	17
	3	32	31	0.98	31	0.79	25
SPE	1	15	11	0.79	12	0.83	12
	2	27	21	0.84	23	0.83	22
	3	58	56	0.86	50	0.83	48
	4	64	60	0.87	56	0.83	53
	5	54	46	0.88	48	0.92	50
	6	32	30	0.87	28	0.91	29
	7	42	38	0.87	37	0.89	37
	8	39	37	0.88	34	0.88	34
	9	58	56	0.90	52	0.96	56
	10	57	37	0.91	52	0.96	55
	11	47	27	0.92	43	0.96	45
	12	37	31	0.93	34	0.96	36
	13	38	26	0.94	36	0.96	36
	14	21	19	0.95	20	0.96	20
SPF	1	17	12	0.67	11	0.74	13
	2	15	12	0.73	11	0.96	14
	3	22	15	0.83	18	0.96	21
	4	23	15	0.88	20	0.96	22
SPG	1	39	30	0.68	27	0.7	27
	2	61	46	0.73	45	0.92	56
	3	34	29	0.82	28	0.96	33
SPH	1	14	9	0.68	10	0.74	10
	2	36	28	0.73	26	0.88	32
	3	59	41	0.82	48	0.92	54

1: Sprint Number 2: Total Estimated Story-Points (SPs) 3: Actual Finished SPs 4: COEEMO Effort Adjustment Factor
5: COEEMO Total Estimated SPs 6: Mamdani FIS Effort Adjustment Factor 7: Mamdani FIS Total Estimated SPs

Table 10.10 Team, COEEMO, and MFIS suggested TESPs for the Industry Data

Projects	[1]Spt#	[2]TEE /SP	[3]CEE /SP	[4]MEE /SP	[5]AE /SP	[6]BRE			[7]BREbias		
						TEE	CEE	MEE	TEE	CEE	MEE
SPA	1	4.50	5.70	6.25	5.73	0.27	0.01	0.09	+0.27	+0.01	-0.09
	2	2.86	3.37	3.05	3.32	0.16	0.02	0.09	+0.16	-0.02	+0.09
	3	10.50	10.82	10.94	15.75	0.50	0.46	0.44	+0.50	+0.46	+0.44
SPB	1	18.00	24.32	33.96	25.20	0.40	0.04	0.35	+0.40	+0.04	-0.35
	2	14.00	16.67	18.92	15.75	0.13	0.06	0.20	+0.13	-0.06	-0.20
SPC	1	18.00	23.68	25.00	22.91	0.27	0.03	0.09	+0.27	-0.03	-0.09
	2	22.91	27.60	27.60	28.00	0.22	0.01	0.01	+0.22	+0.01	+0.01
	3	13.26	14.57	15.07	22.91	0.73	0.57	0.52	+0.73	+0.57	+0.52
	4	14.00	14.58	14.58	16.80	0.20	0.15	0.15	+0.20	+0.15	+0.15
SPD	1	8.22	11.10	11.10	9.00	0.10	0.23	0.23	+0.10	-0.23	-0.23
	2	9.95	11.84	11.30	10.50	0.06	0.13	0.08	+0.06	-0.13	-0.08
	3	5.91	6.03	7.48	12.73	0.03	0.01	0.23	+0.03	+0.01	-0.23
SPE	1	9.33	10.73	11.24	10.00	0.36	0.08	0.13	+0.36	+0.08	+0.13
	2	7.78	8.64	9.37	3.75	0.29	0.08	0.07	+0.29	+0.08	+0.07
	3	3.62	3.98	4.36	3.50	0.04	0.12	0.16	+0.04	-0.12	-0.16
	4	3.28	3.61	3.95	4.57	0.07	0.08	0.13	+0.07	-0.08	-0.13
	5	3.89	4.27	4.23	9.33	0.17	0.03	0.08	+0.17	+0.03	+0.08
	6	8.75	9.62	9.62	7.37	0.07	0.08	0.03	+0.07	-0.08	-0.03
	7	6.67	7.49	7.49	7.57	0.11	0.04	0.02	+0.11	-0.04	-0.02
	8	7.18	8.07	8.16	5.00	0.05	0.08	0.08	+0.05	-0.08	-0.08
	9	4.83	5.42	5.03	5.68	0.04	0.07	0.01	+0.04	-0.07	-0.01
	10	3.68	4.14	3.84	7.78	0.54	0.40	0.48	+0.54	+0.40	+0.48
	11	4.47	4.96	4.65	6.77	0.74	0.60	0.67	+0.74	+0.60	+0.67
	12	5.68	6.24	5.91	8.08	0.19	0.11	0.15	+0.19	+0.11	+0.15
	13	5.53	6.07	5.76	3.68	0.46	0.37	0.40	+0.46	+0.37	+0.40
	14	3.33	3.58	3.47	12.73	0.11	0.05	0.06	+0.11	+0.05	+0.06
SPF	1	11.76	18.10	15.90	33.33	1.83	0.90	1.10	+1.83	+0.90	+1.10
	2	13.33	18.52	13.89	15.38	0.15	0.19	0.11	+0.15	-0.19	+0.11
	3	9.09	11.51	9.47	22.22	1.44	1.03	1.35	+1.44	+1.03	+1.35
	4	8.70	10.74	9.06	20.00	1.30	1.02	1.21	+1.30	+1.02	+1.21
SPG	1	6.15	9.32	8.79	10.43	0.70	0.15	0.19	+0.70	+0.15	+0.19
	2	3.93	5.39	4.28	6.49	0.65	0.20	0.52	+0.65	+0.20	+0.52
	3	7.06	8.82	7.35	11.43	0.62	0.33	0.55	+0.62	+0.33	+0.55
SPH	1	14.29	21.65	19.31	33.33	1.33	0.59	0.73	+1.33	+0.59	+0.73
	2	5.56	7.72	6.31	12.50	1.25	0.64	0.98	+1.25	+0.64	+0.98
	3	3.39	4.29	3.68	10	1.95	1.42	1.71	+1.95	+1.42	+1.71
					[8]**MBRE**	**0.49**	**0.29**	**0.37**			

1: Sprint Number 2: Team Estimated Effort in Man-Hours (MHs)/SP 3: COEEMO Estimated Effort in MHs/SP 4: Mamdani FIS Estimated Effort in MHs/SP

5: Actual Effort in MHs/SP 6: Balanced Relative Error for Team, COEEMO, and Mamdani FIS 7: BRE biased for Team, COEEMO, and Mamdani FIS 8: Mean BRE

Table 10.11 Team, COEEMO, and MFIS suggested Effort Estimates/BRE for the Industry Data

10.5 Effect size, SA and MAE Analysis of the ASD projects (Industry + Published Data)

For the purpose of comparison among PPBE, PPBE extended with the COEEMO, and MFIS, the effect size (i.e., Glass Δ), mean absolute error (MAE), standardized accuracy (SA), and mean balanced relative error (MBRE) values were computed for the 85 sprints of the ten published and industry ASD projects (shown across Tables 10.7 to 10.11). The calculated values of all these metrics are given in Table 10.12. The method PPBE alone had an effect size of 0.18 (Sample's Mean based) which was 0.52 and 0.30 smaller than the same measure for PPBE extended with the

COEEMO and MFIS, respectively. The variations from 0.30 to 0.52 exhibited small to medium effect size. Moreover, the effect size differences based on the pooled means were 0.23 (i.e., a small effect as indicated by COEEMO) and 0.11 (i.e., very small effect size as exhibited by MFIS). Consequently, it could be concluded that the results achieved in using the proposed method (i.e., the COEEMO) were not only statistically significant but also were of practical significance. In a similar vein the other metrics MAE, SA, and MBRE result in better performance of the PPBE extended with the COEEMO than the PPBE alone and MFIS.

Metrics Analysis i.e., SA, MBRE, Effect Size, and MAE				
Methods	**MAE**	**SA**	**MBRE**	**Δ**
PPBE	5.89	22%	0.83	0.18 (PPBE Sample's Mean based) 0.37 (Pooled Mean based)
COEEMO	4.85	36%	0.47	0.70 (COEEMO Sample's Mean based) 0.59 (Pooled Mean based)
MFIS	5.36	29%	0.60	0.48 (MFIS Sample's Mean based) 0.48 (Pooled Mean based)
SA is calculated based on the Pooled Mean of three methods.				

Table 10.12 Various Metrics' Analysis of the Methods for the ASD Projects

10.6 Chapter Review

In this chapter, the details of further data collection and analysis from secondary sources, including publications and industry datasets in the public domain, were presented. Moreover, consideration regarding the COEEMO parameters and ratings based on the collected data was provided. In concluding the chapter, discussions and comparisons were given to reflect the accuracies of effort estimations with and without COEEMO and Mamdani Fuzzy inference, using Balanced Relative Error (BRE) and BREbias metrics. In next chapter, the study's findings and implications in relation to all four research questions are reported.

Chapter-11 Findings and Discussion

Contents' Summary: *This chapter considers in detail the outcomes of the research in the context of each of three important phases, including problem Identification, solution development and validation. Further, the findings are discussed in relation to prior knowledge in the field of effort and cost estimation.*

Organizations that develop software as their core business typically deploy considerable resources to drive innovation and transformation. However, enduring evidence indicates that many software projects fail to deliver within time or budget and do not achieve business objectives. One of the reasons underlying such failure has been an inability to provide accurate forecasts of the time and resources needed for project completion. Numerous estimation techniques have been employed in Traditional Software Development (TSD) to forecast project size, the human effort required, the likely developmental cost and the project schedule, but their effectiveness in Agile Software Development (ASD) is unknown. ASD is currently the most popular development approach, and it brings with it comparatively simple forecasting methods; however, these have several gaps and limitations. This research sought to understand and improve the utility of forecasts of size, effort, cost and schedule for ASD projects through variants of existing approaches based on a range of indicators, including but not limited to stories' size (estimates of task size in relative terms, often in unit-less story points), numbers of sprints (development cycles), stakeholder's coordination, teams' velocity (story points completed in a sprint by a team), experts' domain knowledge, System and the organizational context.

Problem identification was the first phase of this study, in which the extant literature was critically reviewed. Further, based on the outcomes of that review, an industrial review was conducted through in-depth interviews with 21 IT experts. This enabled the discovery of some additional perceptions from the practitioner point-of-view, in addition to validating the problems acknowledged in the literature review. The next phase was the design and development of a solution, in which a framework of novel model for estimation along with the usage of PERT (with recommended modifications) and Expert judgement based on bottom-up approaches were suggested as three artefacts of the conducted study. The third phase of the research was the validation, to determine the coverage and usefulness of the above stated three artefacts (i.e., COEEMO − a novel model of estimation, PERT with suggested modifications, and Expert judgement based on Bottom-up approach). These all three artefacts were evaluated by field trials within industrial settings against the three real-industry projects. However, COEEMO was also assessed empirically against previously completed ASD projects and further, also validated in the laboratory where COEEMO was implemented in MATLAB using the well-established Mamdani Fuzzy Inference System (MFIS).

Figure 11.1 represents in a single diagram the findings of each research phase in summarized format, and these are further deliberated in subsequent paragraphs of this chapter. The items given in cloud-callouts are the outcomes/findings of each phase while the black rectangle displays the novel contributions of the research that have emerged from this study.

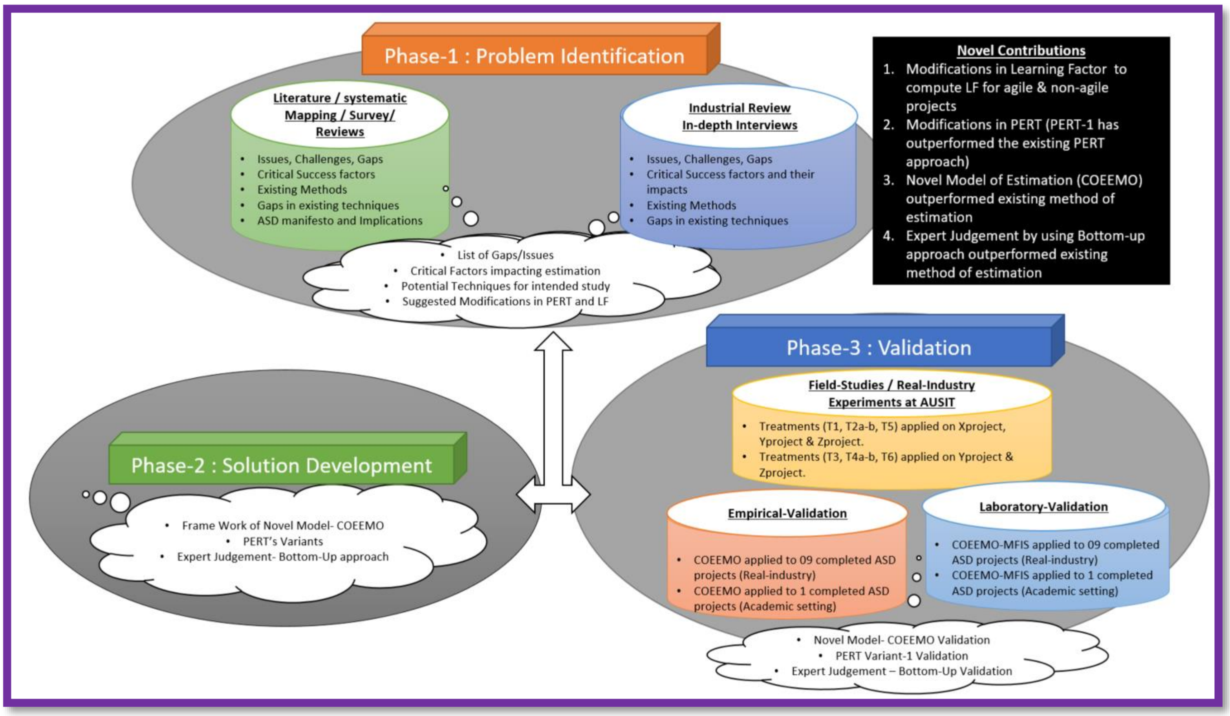


Figure 11.1 Research Phases and Findings

11.1 Problem Identification – Phase 1

The problem identification phase was conducted in two steps to determine the current state of effort estimation in agile development contexts. The intent was to identify all of the factors that could impact effort estimation. At this stage all factors irrespective of development methodologies, i.e., Agile or non-agile, were considered. The first step was envisioned to ascertain the state-of-the-art through reviews of the existing research literature. Moreover, some recent systematic review studies were consulted to provide a comprehensive overview of the existing knowledge in the context of the intended study. A wide-ranging list of the most important factors was prepared. Drawing on the results of the literature review, the second step was anticipated as an exploration of effort estimation in industry, emphasising the challenges faced by practitioners and any success factors they encountered while estimating effort, based on 21 face-to-face interviews with IT practitioners. Figure 11.1 displays the research findings. The literature reviews are presented in Chapters 2 and 3 while the industrial analyses are reported in Chapter 8. Aspects of the findings discussed in the sections of this chapter build on those previously reported results within the domain of effort estimation.

11.1.1 Findings and Discussion – RQ1

11.1.1.1 RQ1's Description

What are the gaps and limitations in contemporary estimation and scheduling techniques when applied in agile development contexts?

11.1.1.2 RQ1's Objectives

To study prevailing estimation techniques employed (in ASD) to forecast size, human effort and schedule for a software project and to identify their weaknesses and limitations, which may restrict them to deliver products within time, or budget constraints.

11.1.1.3 RQ1's Research Methods

Literature reviews of previous studies from academia and practice were conducted to identify issues, gaps and limitations of estimation methods, especially in the context of agile development. Further, some systematic review studies (e.g., Usman et al. 2014 & 2015; Dantas et al., 2018 and others) from academia and practice were also evaluated to understand the prevailing view and the challenges confronted in the domain of estimation for ASD projects.

11.1.1.4 RQ1's Outcomes and Discussion

RQ-1 is a question of enduring importance given that IT practitioners in industry are still having accuracy issues in the estimation process, even though this specific domain of practice has been given attention over many decades. In spite of the development of some quite sophisticated models, tools, and/or methods to address the challenges of estimation accuracy in software development, with some used successfully for a time, there remains some way to go, especially since the emergence of agile software development (ASD) methodologies. Although variants of ASD have been used in multiple forms since the late 1950's, it is only relatively recently that ASD has received global recognition and adoption (Hamdan et. al., 2013). However, besides the success stories of ASD, there remains the fact that the IT industry still has problems delivering on time for complex software development projects resulting in budgeting and scheduling issues. A recent SLR study presents an updated review of the state of the art on estimating effort in ASD since 2014 (Dantas et al., 2018) and it states even the best level of accuracy is not sufficient for good practice. Further, another quite recent study states that the time prediction's average error remains stubbornly at about 20-30% (Halkjelsvik & Jorgensen, 2018).

The core outcome of this step in the research surfaced ten current issues and problems faced by industry in agile development, reported in detail in Chapter-1 under the problem statement.

11.1.2 Findings and Discussion – RQ2

11.1.2.1 RQ2's Description

What is the minimum data required for 'accurate-enough' agile estimation and scheduling? Which factors are influential in agile estimation and so should be measured, and which can be ignored?

11.1.2.2 RQ2's Objectives

To identify and classify factors based on their relevance, significance and utility for forecasting of size, effort, cost and scheduling of projects, as indicated by academia and practice in agile and traditional software development. Further, the aim is to include only those factors that are relevant and may improve the forecasting accuracy of size, effort, cost and scheduling of projects in both contexts, but the principal focus is given to the agile development context given its growing predominance in practice. The core aim is to improve accuracy by considering factors that are (most) relevant/significant in terms of their impact on development effort and cost.

11.1.2.3 RQ2's Research Methods

Literature reviews of previous studies from academia and practice were conducted to identify factors based on their significance and their (perceived) usefulness in effort estimation. In particular this included close review of some more recent studies (Usman et al. 2014 & 2015; Ahimbisibwe et al., 2015; Tanveer et al, 2017; Dantas et al., 2018) in order to understand the impacts of the various factors on effort estimation outcomes. To search the relevant literature, both manual and automatic searches were performed on the basis of relevant keywords (e.g., Effort estimation, agile based estimation, Gaps/Limitations in Agile methods, Critical factors for effort estimations, Significant factors for software development projects, Cost drivers for software projects, Estimation techniques/Methods, Traditional and non-traditional methods of effort estimations, Weaknesses of traditional methods, COCOM, UCP, FPA). Snowballing techniques were also utilized to search for relevant papers from the aforementioned SLR studies. Initially, 550 relevant papers were gathered from four digital libraries, i.e., IEEE Xplore, ACM, SCOPUS, and Google. Later, after applying the exclusion criteria, 350 relevant papers were finalized to include in this study. In addition, individual in-depth interviews were conducted with the industry practitioners to collect data relating to their perceptions of the factors thought to be relevant to estimation and scheduling.

11.1.2.4 RQ2's Outcomes and Discussion

Multiple factors, e.g., organizational, team, customer, project and others, have an impact on the success of software projects, i.e., project completion within the allocated budget and schedule with successful achievement of its goals. A total of 37 factors were revealed in one prior study based on 148 previous articles, considered critical for the success of software projects, and shown in Table 2.1 of Chapter 2. For this study, the ten most critical factors, being the team's experience, expertise, and working relationships, stakeholder coordination, sprint and system complexity, organization culture, working environment and type, and the project's risks, were selected from the pool of numerous factors. The filtration of these factors was based on the twelve principles in the Agile Manifesto; derived from practice evidence; and drawn from the researcher's own domain knowledge and software industry experience. Further, their individual relevance was validated through reviews of the literature from academia and practice reported in Chapter-2. Furthermore, to understand the views of IT industry practitioners and to revalidate the given parameters, face-to-face interviews were conducted individually with practitioners to collect data relating to their perceptions of the factors that were thought to be relevant to effort estimation. The intent of this step was to analyse their relative impact with respect to estimates of size, effort, and cost of projects undertaken in agile development contexts. Further, to seek their point-of-view in relation to the provided list of factors that were considered significant in the context of estimating size, effort, and cost of development projects /

products in an agile development context, along with their proposed descriptions, categories/subcategories, and scaling.

All 21 respondents agreed that the provided factors were vital in relation to effort estimation and had an impact on the accuracy of such estimates. No additional factors were suggested as needing to be considered for this purpose. A small number of the respondents proposed additions in some of the existing sub-categories of the system type (i.e., IoT, Data analytics, and Machine Learning), working environment and the culture categories so that various additional scenarios could be accommodated (e.g., in the work environment initially the "Best" and "Worst" subcategories were provided for the ratings but after feedback from some of the interviewees, two more subcategories e.g. "Semi-best" and "Poor" were added to the factor). The detailed industrial review is placed in Chapter-8.

11.1.3 Problem Identification Phase outcomes

The findings of the literature and industrial reviews verified that there were still accuracy issues in the estimation process, even though this specific domain of practice had been given extensive attention over several decades. In spite of the development of quite sophisticated models, tools, and/or methods to address the challenges of estimation accuracy in software development, with some used successfully for a time, there remained some way to go, especially since the emergence of agile software development (ASD) methodologies. Although some of the existing models based on Artificial Neural Networks and other forms of Machine Learning had been shown to be useful in agile contexts but they were not commonly used by industry due to many inherent issues, including usage and/or implementation intricacy. As a result, the majority of IT teams were still using, or had reverted to using, expert judgement as their main technique for estimating effort.

11.2 Solution Development – Phase 2

The core objective of phase 2 of this study was to provide solutions for (at least some of) the identified gaps and limitations as conceived from the problem identification phase to improve ASD estimation accuracy. To accomplish this single objective, the choices as given below were considered, and finally two of them (i.e., option-1 and -3) were chosen for further exploration, analysis, implementation, and validation, to analyse their impacts on effort estimation. These alternatives were chosen based on factors that include: acceptability within the IT industry; the synopsis of literature from academia and practice within the domain of effort estimation; resources available (i.e., opportunities for real-industry data for experimentation, and validation); and researcher expertise (i.e., researcher's own professional experience and domain expertise). The contexts and precise reasons against each option for their selection or rejection to be used for this phase are given as follows:

a. Option-1: To use existing or modified well-established estimation techniques. In this option, only two techniques were considered, i.e., Expert Judgement (based on gut feelings, analogy, Top-Down and/or Bottom-Up approaches) and (two variants of) PERT. Expert judgement was chosen because evidence suggests this was the most popular technique being utilized in effort estimation for agile software development (ASD), while PERT was selected due to in-depth knowledge and experience of the researcher regarding its usefulness in effort estimation under a plan-driven approach, but its efficacy was unknown in an agile development context.

b. Option-2: To use existing algorithmic models, potentially tailored using Machine Learning and/or soft computing methods that have been shown or suggested to be useful for agile development, perhaps via customization of the traditional algorithmic models: including COCOMO, UCP, FPA, or others that have exhibited efficacy for plan-driven approaches. This choice was not pursued due to a range of issues, core among them being usage and/or implementation intricacy, challenges in understanding the base formulas/techniques, an absence of relevant cost drivers in some of them (e.g., organizational and risk contexts, iteration complexity, team learning curve, and others), or consideration of too many cost drivers.
c. Option-3: To construct a novel algorithmic model for estimation. This option was chosen to address some of the core issues identified in relation to current or prior models as stated under option-2. Further, this envisaged model for estimation was expected to have additional positive features, e.g., being easy for agile teams to understand, configure/reconfigure, and implement; using fewer cost drivers than some other models; being based on simple, transparent rules; using cost drivers relevant to agile development.

11.2.1 Findings and Discussions – RQ3

11.2.1.1 RQ3's Description

How can the significant factors as discovered in RQ-2 be effectively calibrated to build a dynamic model of estimation on its own and in combination with other subjective assessments to further improve the accuracy of the estimations?

11.2.1.2 RQ3's Objectives

To develop an algorithmic model that improves the utility of forecasts of size, effort, cost and schedule for ASD projects, based on the significant/relevant factors identified in RQ-2. The proposed values of these factors are further calibrated to accommodate their impacts on the accuracy of forecasts of size, effort, cost and schedule for ASD projects both alone and in combination with other subjective assessments / techniques.

11.2.1.3 RQ3's Research Methods

As noted previously, reviews of the literature from academia and practice were conducted to identify the factors relevant to effort estimation. Consequently, the most important factors (ten in number) were finalized and revalidated based on the responses of 21 IT practitioners secured through individual in-depth interviews. Further, based on the stated factors, an algorithmic model named the COEEMO (Constructive Effort Estimation Model) was developed.

11.2.1.4 RQ3's Outcomes and Discussion

COEEMO (the Constructive Effort Estimation Model) is a novel contribution of this research. This model has demonstrated statistical and practical significance. The detailed salient features and the scenario-based implementations of the model are given in Chapter 4. This model has shown its practical significance when the AUSIT teams adjusted estimated story-points for their three projects with the COEEMO Adjustment Factor (CEAF). Further, COEEMO can also be used in situations where teams do not want to provide estimated story-points and desire that COEEMO should predict story-points on its own (the details of such scenarios are demonstrated in Chapter 4, section 4.6) along with the CEAF for a project's sprint/iteration. However, the efficacy of this specific feature of COEEMO could not be evaluated because of restrictions on the scope of the study. Furthermore, COEEMO can also be employed for estimating effort for non-agile software projects where their implementations are undertaken with or without iterations/sprints (e.g., RUP, Waterfall and others). Though the effectiveness of COEEMO to estimate effort in non-agile projects is unknown this aspect could be evaluated in the future research endeavours.

The results of the experimentation as reported in Tables 9.2 & 9.3 revealed that COEEMO (represented as treatments T5 & T6) outperformed the other treatments (T1, T2a-b, T3 and T4a-b). It delivered an overall accuracy improvement of 56% (i.e., treatments T1 Vs T5) in effort estimation when the team suggested effort by using the existing method (i.e., Planning Poker based estimation (PPBE) by using expert judgement based on gut feeling, analogy or Top-Down methods but excluding the Bottom-up approach ─ indicated as PPBE or T1) but tweaked by the COEEMO adjustment factor (CEAF). Further, out of 12 sprints, as given in Table 9.4, PPBE and COEEMO overestimated 10 and 6 sprints, respectively.

Treatment-3 (i.e., PPBE extended with the Bottom-Up Approach ─ indicated as $PPBE_{BUA}$ or T3) as reported in Tables 9.2 & 9.3, demonstrated better performance than PPBE alone (i.e., MBRE 8% Vs 15%). Approximately 47% improvement in the accuracy of effort estimates was observed while using $PPBE_{BUA}$ over PPBE alone. These results approximately mirror those found by Jørgensen (2005) when comparing expert judgement techniques, i.e., Top-Down with Bottom-Up approach. Further, $PPBE_{BUA}$ resulted in 4/6 sprints as overestimated, while PPBE have shown overestimation in 10/12 sprints. However, the COEEMO based treatment T6 performed close to $PPBE_{BUA}$ (i.e., MBRE 9% Vs 8).

11.2.2 Findings and Discussions – RQ4

11.2.2.1 RQ4's Description

How can expert judgement, as applied in existing estimation methods for ASD projects, be augmented by applying PERT and Bottom-Up approaches to analyse their utility on forecasts of size, effort, cost and schedule for ASD projects?

11.2.2.2 RQ4's Objectives

To analyse the utility of expert judgement methods in terms of accuracy of forecasts of size, effort, cost and schedule for ASD projects. Evidence has shown that expert judgement is one of the dominating techniques in current methods of estimation for ASD projects. This prevailing technique was extended by two other well established methods, i.e., PERT and Bottom-Up.

11.2.2.3 RQ4's Research Methods

The research method used for RQ4 was a quasi-experiment. The purpose of taking such an approach was to systematically evaluate the impact of a range of forecasting techniques and their variants Bottom-Up and PERT. This enabled a comparative analysis of the existing methods of estimation along with the proposed model of estimation, i.e., COEEMO. The performance of the employed treatments was assessed using best practice error measures, Balanced Relative Error (BRE) and BREbias. The detailed experimental planning, which included the design of the experiment, the hypothesis formulation, and the independent & dependent variables of all the treatments, are reported in Chapter 6.

11.2.2.4 RQ4's Outcomes and Discussion

The results of a one-way ANOVA test (parametric) and Kruskal-Wallis test (a non-parametric test used due to the small sample size) applied on the data of Treatments 1-6 are shown in Figures 9.6 and 9.7 of Chapter-9. Based on these results the alternate hypothesis (H_1) was accepted, while rejecting the null hypothesis (H_0), which meant that Treatments 1-6 had statistically significant variations among their outcomes.

The six treatments were as follows: T1 - Planning Poker based estimation (PPBE), this was an existing methodology that teams were using in their estimations by using expert judgement based on gut feeling, analogy or Top-Down techniques but excluding a Bottom-Up approach; T2a - treatment T1 extended with variant-1 of the PERT); T2b - treatment T1 extended with PERT's variant-2; T3 - Planning Poker extended with expert judgement by following strictly the Bottom-Up approach based on the work break down structure; T4a - treatment T3 extended with PERT's variant-1); T4b - treatment T3 extended with PERT's variant-2; T5 - treatment T1 extended with the COEEMO, the proposed parametric model of this study; and T6 - treatment T3 extended with the COEEMO. All six were applied on three real industry projects (i.e., Xproject, Yproject, and Zproject of AUSIT). The relative errors of these treatments were measured by using the two metrics, BRE and BREbias.

The Mean BRE results for all treatments are shown in Table 9.2. Treatments T5 and T3 resulted in lower MBRE values, i.e., 7% and 8% individually, as compared to all other treatments. In comparing the two treatments (i.e., T2a-b, & T4 a-b) based on PERT's variants, it is noted that the treatments employing PERT's variant-1 resulted in better MBRE values for estimate accuracy as compared to the treatments using variant-2 of PERT. Further, PERT's variant-1-based treatment T2a performed much better when the team suggested effort estimates are assessed by using PPBE as corresponding to the $PPBE_{BU,}$ where PERT's variant-1-based treatment T4a performed inadequately. Further, PERT's variant-2-based treatments T2b & T4b performed very poorly when the team suggested effort estimates are assessed either by the PPBE and/or the $PPBE_{BU}$. Furthermore, $PPBE_{BU}$ performance is much better in delivering more accurate effort estimates than the PPBE, and the findings have shown overall a 47% enhancement in the estimates' accuracy as corresponding to the PPBE. However, use of the COEEMO resulted in a 56% improvement in estimates' accuracy (w.r.t the treatment PPBE).

11.2.3 Solution Development Phase Outcomes

Three artefacts, i.e., COEEMO, the novel model for estimation; PERT-1 (a modified approach of PERT); and Expert Judgement using a Bottom-up approach, were developed or used in response to the aggregated findings of the literature and industrial reviews to provide estimation accuracy in agile development. COEEMO can be used for estimating effort for agile and non-agile projects. Moreover, this can also be utilized in a situation where a team needs COEEMO to provide the projected effort estimates of its own accord, along with estimates' adjustments for a project's sprint/iteration. However, the effectiveness of COEEMO in estimating effort for non-agile projects and to provide projected effort estimates on its own still require appraisal as these aspects were out of scope of the current study. Mapping of these artefacts along with identified gaps/issues (presented in Chapter-1), and the provision of their solutions against each, are exhibited in Figure 11.2. The single arrow in the figure signifies the solution of a unique identified problem or gap while the multiple arrows from the artefacts indicate a number of gaps and problems that the particular artefacts resolve.

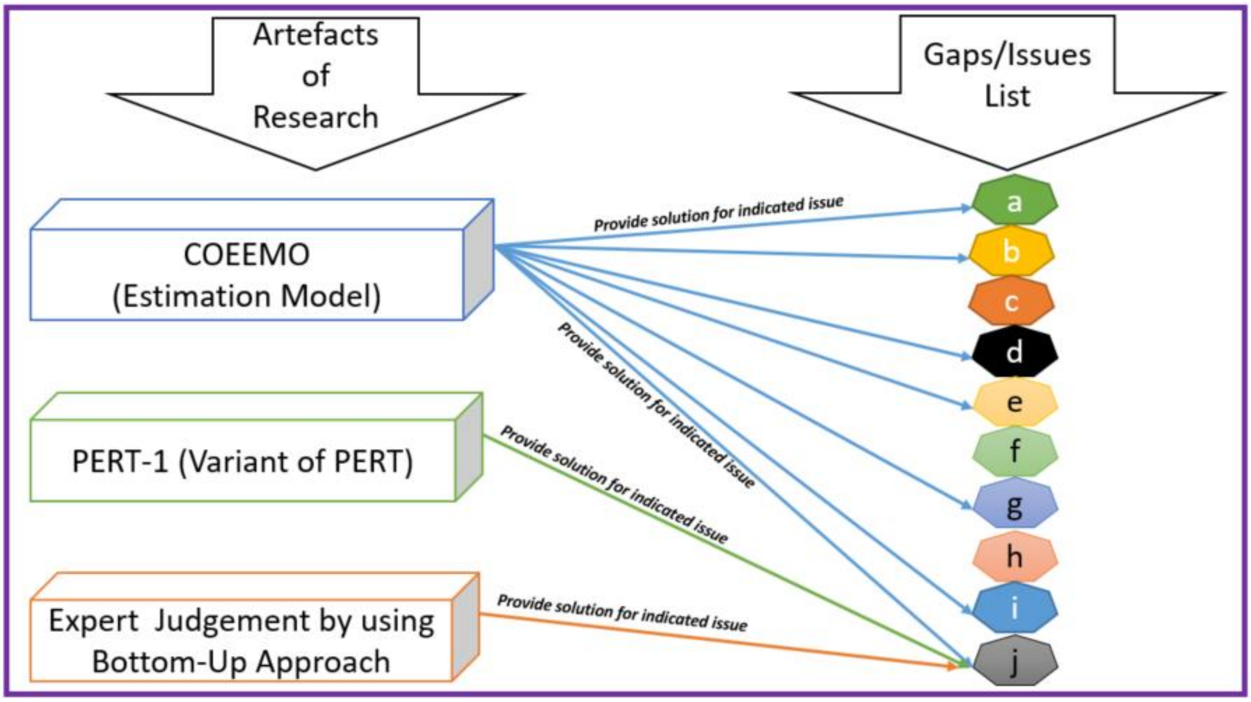


Figure 11. 2 Mapping of Research Artefacts Providing Solution(s) for the Indicated Gaps/Issues

11.3 Validation Phase

This phase was conducted to further evaluate, and quantitatively validate, the findings of the solution development phase of the research. To validate the artefacts of the second phase, trials from three distinct perspectives were employed, i.e., field, empirical and laboratory. In the first context, the COEEMO, PERT with two variants, and Expert judgement with Bottom-up approach were applied in experimental settings on real industry ASD projects (i.e., Xproject, Yproject and Zproject) and their results were compared with those obtained from existing methods. Details of the testing along with the outcomes of all the field trials/experimentation are provided in Chapter 9.

The second perspective, empirical validation, was accomplished by utilizing the COEEMO model on the data of already completed ASD projects collected from two different sources: published studies and the IT industry. The outcomes of the empirical validation are discussed in detail in Chapter 10.

For the context of laboratory validation, the COEEMO was implemented by using a well-established fuzzy inference model (i.e., the Mamdani model in the MATLAB toolbox) and both models, COEEMO and Mamdani, were applied to the already completed projects, and their results were compared. In all three validation perspectives, two metrics were used to measure the accuracy rate (BRE) and the direction of any error (BREbias) for each effort estimation. A model/method resulting in a lesser value of BRE compared to another such model is considered more accurate.

11.3.1 Empirical Validation results – COEEMO Vs PPBE and MFIS

In the case study-A of the published data, when the team's suggested effort estimates of the ten sprints were modified with the CEAF (COEEMO-based Effort Adjustment Factor), more accurate results were achieved for 6 sprints in that their estimates were close to the actual values (exhibited in Table 10.5). The MBRE shown by the COEEMO for estimate accuracy was 5% as compared to 6% by the PPBE alone. When comparing the effort estimates as given by COEEMO and MFIS with the actual estimates of the case study-A, it is noted that COEEMO performance was better in 6 sprints as compared to MFIS, that could perform well in only 4 sprints. Moreover, the MBRE was only 5% for the COEEMO-suggested effort estimates as compared to 8% shown by the MFIS.

Regarding the case study-B of the published data, COEEMO and MFIS both exhibited better estimates in 21 and 20 sprints respectively as compared to the PPBE. However, the estimation results for the teams, COEEMO and MFIS were similar in sprint-3. The sprint-wise MBREs as demonstrated by the COEEMO, MFIS and the PPBE were 1.91, 2.40 and 3.51 for the sprint-1 and 0.25, 0.31 and 0.42 for the sprint-2 (presented in Table 10.6). The MBREs for the sprint-3 in case of COEEMO and MFIS were 0.1 while in PPBE alone this was 0.11. Further, use of the COEEMO and MFIS resulted in 44% and 30% overall enhancement in the accuracies, respectively, for the teams' estimates of the case study-B as compared to the PPBE alone.

The results regarding the industry data, as exhibited in Table 10.7, showed that the teams' estimates for 28 sprints out of 36 (i.e., the collective number of sprints for the 8 ASD projects) delivered greater accuracy when their estimates were adjusted by using the CEAF and MEAF, as compared to the teams' estimates without CEAF and MEAF. Further, the MBREs for the teams' estimates adjusted with CEAF and MEAF were 29% and 37% respectively, as compared to 49% without the stated adjustments. Moreover, the COEEMO and MFIS generated overall improvements of 41% and 24% in the accuracies for the teams' estimates of the industry data as compared to the PPBE alone.

Further, the effect sizes (i.e. Glass Δ), mean absolute error (MAE), standardized accuracy (SA), and mean balanced relative error (MBRE) were computed for the 85 sprints of the ASD projects (with the data of all these projects / sprints shown in Tables 10.5, 10.6 and 10.7). The calculated values of all these metrics are given in Table 10.8. The method PPBE alone had given the effect size of 0.18 (Sample's Mean based) which was approximately 0.52 and 0.3 smaller than the PPBE extended with the COEEMO and MFIS, respectively. The variations from 0.3 to 0.52 exhibited a medium effect size. Moreover, the effect size differences based on the pooled means were 0.23 (i.e., a small effect as indicated by COEEMO) and 0.11 (i.e., a very small effect size as demonstrated by MFIS). Consequently, it can be concluded that the proposed model of estimation (COEEMO), is not only statistically more accurate but that also has the difference in accuracy is of practical significance. The other metrics considered, i.e., MAE, SA, and MBRE, demonstrated the similarly better performance of the PPBE extended with COEEMO than the PPBE alone and MFIS.

11.3.2 Validation Phase Outcomes

This phase comprised the quantitative assessment of COEEMO against 10 ASD projects across a total of 85 sprints. Both the overall and comparative analysis provided strong evidence that COEEMO should successfully support effort estimation for ASD projects.

11.4 Chapter Review

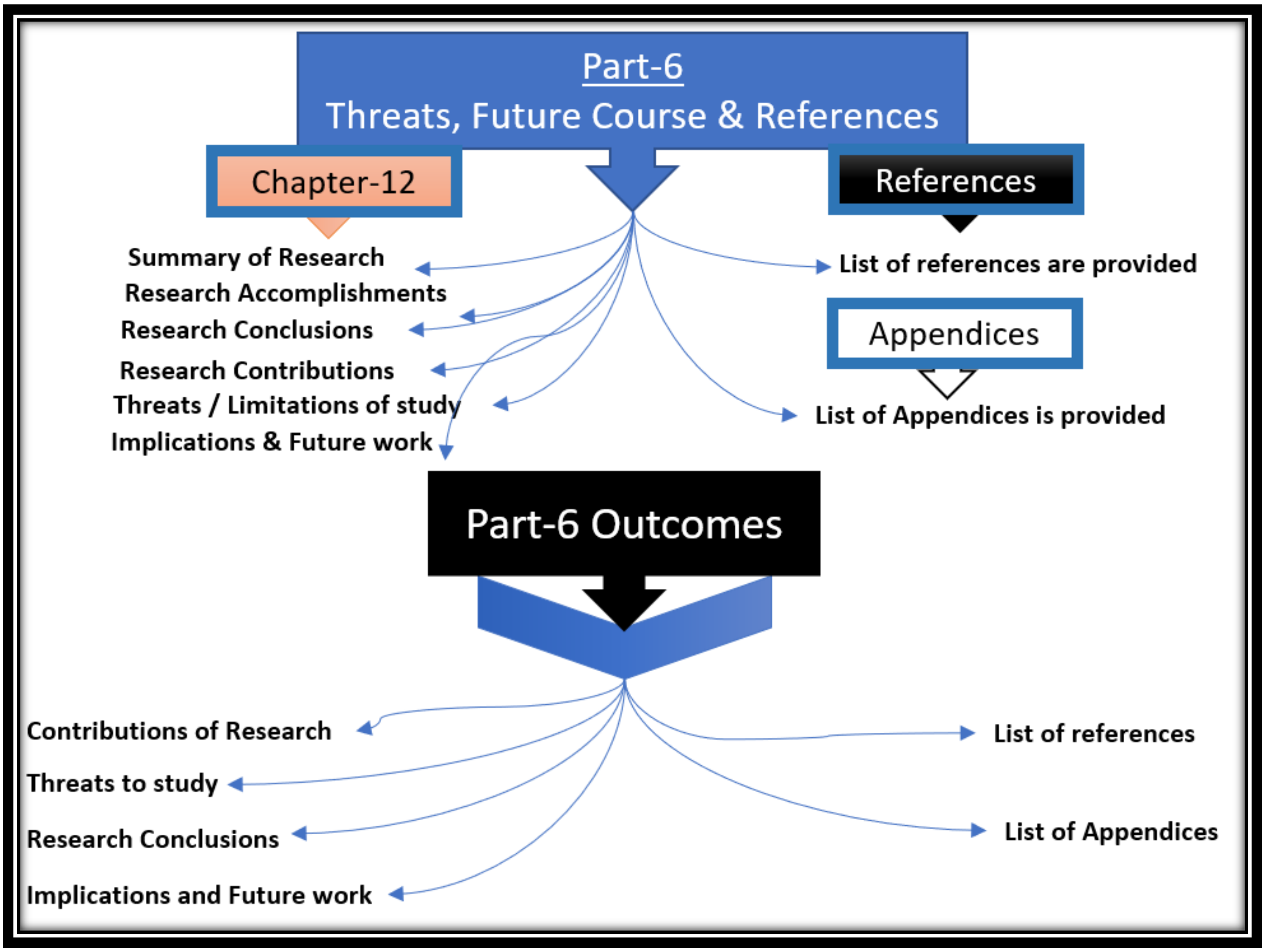
In this chapter, three research phases comprising problem identification, solution design & development and validation were discussed. In the problem identification phase, the underlined gaps/issues of the intended research were explained along with the details of the industrial reviews, in which face-to-face interviews were conducted. In the solution development phase, three artefacts were discussed which could be utilized to improve effort estimation in agile development. At the end of this chapter, the validation phase was discussed, in which assessments based on three perspectives were given for the quantitative validation of COEEMO against 10 completed ASD projects over 85 sprints.

In the next chapter, the conclusion of this thesis and the research reported in it are described. Further, a summary of the research work undertaken along with a description of the key contributions are provided. Details regarding threats to the validity of the research along with the implications of this research and future research prospects are presented.

PART-6 THREATS, FUTURE COURSE AND REFERENCES

This part consists of one chapter, references, and appendices. This part includes summary of the study, along with the core contributions of this research. Further, this part also discusses the implications, future work and limitations of the study. Furthermore, the lists of references and appendices are presented.

Detailed views of the research activities with respect to the chapter within this part are presented in the figure as given below. On the top of the diagram, the chapter, references, and appendices covered in this part are shown along with their different research endeavours as accomplished within the individual chapter, while the outcomes of Part-6 based on its individual chapter are exhibited at the bottom of the figure.



Chapter-12 Conclusion

***Contents' Summary**: This chapter concludes this thesis and the various research endeavours reported in it. It outlines the work undertaken, demonstrates the key contributions made, and acknowledges threats to the validity of the research. It also contemplates the implications of this research along with future research prospects.*

12.1 Summary

The research reported in this thesis has explored software development effort estimation, with a particular focus on agile development contexts. The research was undertaken in three phases following a design science research methodology: identification of the problem, solution design and development, and evaluation. The study addressed two core objectives:

- first, to understand a range of aspects relevant to effort estimation in agile development, including challenges, constraints, critical factors, prevailing estimation techniques and their built-in complexities and issues, along with their benefits, conducted primarily during the first research phase;
- and second, to recommend, appraise and refine three artefacts: COEEMO a novel model for effort estimation, and two variants of existing techniques, being PERT with suggested modifications and Expert Judgement using a Bottom-Up approach (EJBU), as performed during the second and third research phases.

The key motivations for this research were based on the findings of the literature reviews, which indicated that there continue to be accuracy issues in the estimation process, even though this domain of software practice has been given attention over many decades. In spite of the development of some quite sophisticated models, tools, and/or methods to address the challenges of estimation accuracy in software development, enduring evidence indicates that many software projects still fail to deliver within time or budget and do not achieve business objectives.

The phase of problem identification for this research was divided into two portions so that sequential examinations of the state-of-the-art and the state-of-practice might be possible. The prevailing literature was studied to determine the state-of-the-art regarding effort estimation. To search the relevant literature, both manual and automatic searches were performed. Snowballing techniques were also utilized and initially 550 relevant papers were gathered from four digital libraries, i.e., IEEE Xplore, ACM, SCOPUS, and Google. Later, after applying the exclusion criteria, 350 relevant papers were finalized to include in this study. While researchers might conduct a systematic mapping study in the area of their research to determine current opportunities this was not done in this case because of the availability of two systematic studies on the topic of interest (Usman et al. 2014; 2015) in the years preceding that when the intended study was commenced (2016).

The results of the literature review indicated that effort estimation still needs attention, with accuracy proving difficult to improve with existing models. One of the core reasons for this inaccuracy could be models' failure to consider the impact of factors that can affect the cost of a project, including the desired reliability and usability of the software, team experience and culture, corporate culture, workplace factors, use of modern tools/practices, offshore development, and others. Interestingly, many of these factors had been considered critical in traditional estimation approaches, e.g., Function Point Analysis, UCP and COCOMO and the like, but how such factors were or could be accommodated in ASD sizing and estimation was not yet clear.

The investigation of the current state-of-the-art also contributed to a refinement of the primary research goal, that is, “To provide a method / technique, or a combination of more than a single method/technique, for estimating effort which should be dynamic to cater for different situations and that may provide useful features, e.g., being easy for an agile team to understand, configure, reconfigure, and implement; using fewer cost drivers than some other models so that team may readily complete data-entry requirements; being based on simple, transparent rules, using cost drivers relevant to agile development; supporting the team in their estimation activities without replacing them”.

Part-2 of the problem identification phase addressed the state-of-practice, wherein an industrial review was conducted. The intent of the industrial review was to determine the importance of the range of identified factors and their relative influence on effort estimation in agile development, and to investigate the challenges practitioners faced during estimation, and further, what solutions they might adopt to resolve them. During this phase of the research data was collected through face-to-face interviews with 21 practitioners. The interviews were conducted using two types of questions, i.e., open-ended questions so that practitioners could provide as much information as possible and closed questions so that they could succinctly and quantitatively report likely levels of influence of each factor in the context of estimation accuracy on the conveyed grading/scaling. The data collected from interviewees was transcribed and daily interpretive analysis (DIA) was utilized to assemble and interpret the information as collected from the interviewees at the end of every day of interviewing.

The conclusions of the industrial review, building on the outcomes of the literature review, informed a preliminary framework of the proposed model for estimation based on a set of factors along with their likely influences on effort estimation. In addition, one well-established technique was chosen for analysis, i.e., Expert Judgment by using a Bottom-Up approach. All other types of expert judgement approaches, based on gut feeling, analogy, Top-Down approaches, and others, were not selected because they were already described as being used in practice. A third technique, i.e., PERT (modified with two variants) was also to be employed as a further alternative to address the underlying issues of estimation inaccuracy.

The core aim for the second phase of the research was the design and development of a suitably lightweight model for effort estimation that would succinctly and successfully support the estimation activities of agile teams. Further, to suggest modifications in PERT and also to devise guidelines so that the usefulness and support of PERT and EJBU in the estimation pursuits could be assessed in the next phase.

The final phase (i.e., the evaluation phase) of the study sought to substantiate the developed solutions (i.e., COEEMO – Constructive Effort Estimation Model, PERT with two variants, and EJBU). To execute this phase, trials from three distinct perspectives were employed, i.e., field, empirical and laboratory. The core objective of using multiple perspectives was to broadly evaluate the perceived usefulness of artefacts(s) within effort estimation across a range of project contexts.

The field perspective of COEEMO had not only exhibited statistical significance but also resulted in 56% overall improvement in estimation accuracy while the PERT variant-1 and Expert judgement based on Bottom-up approaches had exhibited 25% and 47% enhancement in accuracy of estimates respectively over the existing method for the ASD projects of AUSIT. In order to address the other two perspectives, one empirical and the other laboratory, COEEMO was implemented using the well-established Mamdani fuzzy inference model in the MATLAB toolbox and applied to the data of 85 sprints across ten already completed ASD projects. In all three perspectives of evaluation, COEEMO outperformed the other techniques/methods of estimation for the projects that were used for validation. Further, tests of effect size indicated that COEEMO was also significantly different to the other methods in a practical sense.

12.2 Research Accomplishment and Conclusion

Accurate estimation of effort has long been held to be important in the planning and management of a software development project and it remains an ongoing challenge in the field of software engineering. Thus, the central objective of this research was to provide a solution to reduce inaccuracies in effort estimation for agile software development projects. To succeed in this core objective, four sub-objectives were stated for each phase of this research, given as follows.

Objective-1: To study prevailing estimation techniques employed (in ASD) to forecast the size, human effort and schedule of a software project and to identify any weaknesses and limitations that may constrain the delivery of products within time or budget constraints.

Addressed in Chapters 1, 2 and 3: A comprehensive literature review of previous studies from academia and practice was conducted to identify issues, gaps and limitations of estimation methods, especially in the context of agile development. Further, recent systematic review studies (e.g., Usman et al. 2014 & 2015; Dantas et al., 2018 and others) from academia and practice were also evaluated to understand the prevailing views and the challenges confronted in the domain of estimation for ASD projects.

Objective-2: To identify and classify factors based on their relevance, significance and utility for the forecasting of size, effort, cost and scheduling of projects as indicated by academia and practice in agile and traditional software development. Further, to include only those factors that are relevant and may impact/improve the forecasting accuracy of size, effort, cost and scheduling of the projects in TSD and ASD contexts, although the principal focus is given to agile development. The core aim is to improve accuracy by considering the factors that are (most) relevant/significant in terms of their impact on development effort and cost.

Addressed in Chapters 2, 4 and 8: Multiple factors related to human, organization, process, system, tool, product, and the physical environment have an impact on the success of a software project, i.e., project completion within the allocated budget and schedule with successful achievement of its goals. A total of 37 factors critical for the success of software projects as identified in a prior study based on 148 previous articles from the state-of-the-art and state-of-the-practice were also considered significant for this study. Subsequently, the ten most critical factors, being the team's experience, expertise, and working relationships, stakeholder coordination, sprint and system complexity, organization culture, working environment and type, and the project's risks, were selected from the pool of factors based on the twelve principles in the Agile Manifesto; and / or were derived from practice evidence; and / or were drawn from the researcher's own domain knowledge and software industry experience. The relevance of each of these preferred factors was indicated through reviews of the literature from academia and practice.

However, there was a need to ascertain current assessments of practitioners from the IT industry regarding prevailing estimation methods, challenges and constraints, the importance of the designated factors, their likely impacts on estimation, and the relevance of any additional factors. Therefore, face-to-face interviews were conducted individually with 21 IT practitioners to collect data relating to their perceptions of the factors that were thought to be relevant to effort estimation. The core purpose of this industrial review was to understand the challenges that the practitioners faced when estimating their own products and to analyse the relevance of the preferred factors, along with their likely influence on estimates of size, effort, and cost of projects undertaken in agile development contexts. Further, the intent was to seek practitioners' point-of-view regarding the categories and subcategories of the designated factors and the subjective criteria for their selection and scaling.

Objective-3: To develop an algorithmic model that improves the utility of forecasts of size, effort, cost and schedule for ASD projects, based on the significant/relevant factors identified under Objective 2. The proposed values of these factors are further calibrated to accommodate their impacts on the accuracy of forecasts of size, effort, cost and schedule for ASD projects both alone and in combination with other subjective assessments / techniques.

Addressed in Chapter 4, 6-10: Through comprehensive reviews of the state-of-the-art and state-of-the-practice a list of ten factors impacting estimation was deliberated and, further, validated by an industrial review. The Constructive Effort Estimation Model, COEEMO, was proposed. The model's framework, significant components, factors classified into categories/subcategories, criteria to choose a specific subcategory within a category, their relationships and scaling/ratings, were developed. Fundamental metrics were designed to compute the COEEMO Effort Adjustment Factor (CEAF), Effort, Technical, Environmental, and Learning factors. Moreover, various scenarios for COEEMO's implementation in agile and non-agile contexts were exhibited.

Objective-4: To analyse the utility of expert judgement methods in terms of accuracy of forecasts of size, effort, cost and schedule for ASD projects. Evidence has shown that expert judgement is one of the predominant methods of estimation for ASD projects. This prevailing technique was extended by two other well established methods, i.e., PERT and Bottom-Up.

Addressed in Chapters 2 and 6-10: The research method designed to achieve this objective was a quasi-experiment. The purpose of taking this approach was to systematically evaluate the impact of a range of forecasting techniques and their variants, Bottom-Up and PERT. This enabled a comparative analysis of the existing methods of estimation along with the proposed model of estimation, i.e., COEEMO. The performance of the employed treatments (that were six in numbers) was assessed using best practice error measures, Balanced Relative Error (BRE) and BREbias. Based on the results of one-way ANOVA tests (parametric) and Kruskal-Wallis tests (a non-parametric test used due to the small sample size) the alternate hypothesis (H_1) was accepted, while rejecting the null hypothesis (H_0), which means that all of the approaches had variations among their outcomes.

To validate the envisaged solution, trials from three distinct perspectives were employed, i.e., field, empirical and laboratory. In the first context, the COEEMO along with Expert Judgement and PERT's two variants were applied in experimental settings on three real industry ASD projects of AUSIT and their results were compared with those obtained from existing methods. The experimental results indicated 56% and 25% improvements in accuracy for COEEMO and PERT's variant-1-based effort estimates respectively, as compared to the existing method of effort estimation for real-industry ASD projects. While for the same projects, COEEMO-based effort estimates produced 13% accuracy improvement over the method of estimation using EJBU. EJBU approach presented a 47% improvement in estimation accuracy over the existing method of estimation (i.e., expert judgement based on gut feeling, analogy, and Top-Down approaches). PERT's variant-2 failed to show any improvements in accuracy as compared to the existing method of effort estimation, while both PERT variants also did not demonstrate accuracy improvements as compared to estimation based on EJBU.

The second perspective, empirical validation, was addressed by utilizing only the COEEMO model on the data of already completed ASD projects collected from two different sources: published studies and the IT industry (The other two techniques, i.e., EJBU and PERT's two variants could not be applied because of the absence of the relevant data). For the context of laboratory validation, the COEEMO was implemented using the well-established Mamdani fuzzy inference model (MFIS) in the MATLAB toolbox. Both models, COEEMO and MFIS, were applied to the already completed projects, and their results were compared. Validation from both perspectives (i.e., Empirical, and Laboratory) concluded that COEEMO-based effort estimates exhibited overall a 43% improvement while MFIS resulted in a 28% improvement in accuracy for those ASD projects as compared to the existing method of estimation.

Based on the achievement of each of the research objectives (1 to 4) just reported it is asserted that the core research objective of the study has been accomplished. Two novel techniques for effort estimation (i.e., COEEMO and PERT's variant-1) have been developed, and their efficacies in terms of improved estimation accuracy have been established for ASD projects as compared to the existing method of effort estimation. Moreover, the use of one existing technique, that is, expert judgement based on a Bottom-Up approach, has also exhibited improvements in accuracy over ASD projects when compared to the existing method of estimation.

12.3 Research Contributions

The research conclusions conveyed in this thesis contribute to the body of knowledge on effort estimation in agile development as well as to software estimation practice more generally, as described in the succeeding paragraphs. There are three primary contributions deriving from this research in the area of effort estimation with the intent to improve accuracy: design and development of the novel COEEMO model (comprising COEEMO's basic concept, functional view, components, categories/subcategories, subjective criteria for subcategory selection, scaling, and various scenarios for implementation, as reported in Chapter 4); a tailored approach of PERT, referred to as PERT's Variant-1 (detailed in Chapter 6); and the use of EJBU approach as opposed to expert judgement based on gut feeling, Top-Down, analogy and others, to seek higher accuracy in effort estimates for ASD projects (where details are provided in Chapters 6).

12.3.1 Research contribution-1

A constructive model of effort estimation (COEEMO) was devised. This model comprises ten important factors grouped into categories and subcategories (detailed in Chapter 3). This model was designed and developed while keeping in view the following key requirements:

a. Provide improvements in effort estimation accuracy
b. Dynamic in nature
    i. Capable to adjust the team's effort estimates for agile and non-agile projects
    ii. Should provide estimates on its own for ASD projects, e.g., where a team is new and/or migrating to agile development
    iii. Configurable/reconfigurable based on improvements or changes in team expertise, experience, relationship, culture and/or team reflections per sprints

c. Using fewer cost drivers (10 factors) as compared to prior models so that the team may readily complete the model's data entry requirements
d. Based on simple, transparent rules
e. Using cost drivers relevant to agile development
f. Supporting the team in their estimation activities without replacing them.

Some of the above stated features of COEEMO were verified through experimentation and empirical validation e.g., the features stated at para a, b ─ for only agile project and b (iii); para f ─ model provides adjustments on team provided estimates. In addition, two of the features were verified through reviews of literature from academia and practice e.g., para c ─ when compared to the number of factors considered within COCOMO (15 factors are considered for Intermediate COCOMO), and UCP (21 factors along with consideration of Use case and Actors' ratings); para e ─ all factors except Organization's Type (considered significant as per the researcher's own experience) were deliberated as relevant to the agile context based on previous studies (detailed in Chapter 2). However, the features at para b (i) ─ COEEMO usefulness for estimating effort for non-agile projects, b (ii), and d were not yet evaluated due to the inherently limited scope of this study (although effort has been made to use simple formulas and transparent rules to calculate various factors of the model ─ COEEMO Effort Adjustment Factor, Effort Factor, Technical Factor, Environmental Factor and Learning Factor). COEEMO provided 50% improvements overall when teams' effort estimates of 97 sprints (for 13 ASD projects including 3 real-industry projects of AUSIT) were adjusted by using the COEEMO Effort Adjustment Factor, as compared to their existing method of estimation.

In terms of practice, the model could be useful in reducing or better managing project costs due to improved estimation accuracy. Further, the continuous use of COEEMO by an agile team in their sprint planning meetings may help the team to gain greater confidence in the model, and therefore, it can be expected that use of the model may reduce 25-50% duration for time-boxed meetings. This could provide the team with extra time as an offset benefit of using this model, which the team can then utilize for their actual development. However, this aspect is yet to be validated.

12.3.2 Research contribution-2

Program Evaluation and Review Technique (PERT) is an established method used to estimate the amount of time required to complete a project task. This technique was modified via two variants to be used for estimating story-points for ASD projects (detailed in Chapter 6). PERT's variant-1 outperformed the existing method of estimation, providing a 25% improvement in accuracy for the three real-industry projects of AUSIT. This contribution is beneficial for industry in situations where agile teams would prefer not to use an algorithmic model – they can use this technique as an alternate option. However, it is important to reiterate that this technique's accuracy results for effort estimation, as shown by the experiments, are inferior to the results achieved through the use of the COEEMO.

12.3.3 Research contribution-3

EJBU approach, following a work breakdown structure (WBS), was utilized on the three real-industry ASD projects of AUSIT during the experimentation stage of this study. EJBU demonstrated a 47% overall improvement in accuracy for effort estimation as compared to AUSIT's existing method. This contribution is valuable for industry and could be helpful in saving project costs due to improvements in accuracy, if the team decide to use this technique for estimating the effort of their ASD projects. However, as with EJBU, this technique's accuracy results as shown by the experiments are inferior to the results of the same demonstrated by COEEMO.

12.4 Limitations

All finite empirical studies have limitations that can threaten the validity of the results obtained; this study considers these threats in terms of internal, conclusion, and external validity. Details of the currently acknowledged threats are explained in the subsequent sections.

12.4.1 Internal Validity

Internal validity covers numerous concerns related to the effects of treatments, their results, and the background of the participants which may influence outcomes, conclusions and deductions of a study (Runeson and Höst, 2009).

To conduct an empirical study rigorously and so reach definitive conclusions it is preferable to select and/or assign samples randomly from a large population. However, in this study such an approach was not possible because of the limited availability of teams and projects. A quasi-experimental research design was chosen (as detailed in Chapter-6) in real industry settings and as such the study was reliant on the willingness of industry organisations to take part, wherein we were able to apply treatments in actual ASD projects. To partially mitigate this threat, estimation activities were applied through a number of cycles and during the execution of (a number of) projects, across the full development life-cycle. In addition, other project data relating to a number of already completed ASD projects of different types (e.g., MIS, mobile applications, financial systems) were also collected from a number of source organizations and these too were analysed, enabling us to better cover a wider range of projects and teams in industry.

12.4.2 Conclusion Validity

Given that in a real-life industry setting treatments are expected to be applied by the trained industry team itself it is likely that this inadequacy may have had some effect on the conclusions. To mitigate this threat, close virtual interactions with the teams concerned were employed in a range of forms e.g., the researcher was available via SMS/phone, and 44 times chats/calls were exchanged, 5 video meetings were conducted over Skype, and around 150 Whatsapp chats/calls occurred so that they could follow the right method to apply on treatments. This enabled the researcher to answer any queries of the team/Capability Lead and to provide clarity on the applications of the proposed treatments. During the empirical validations of the proposed model (COEEMO) the data from published studies and industry for previously completed ASD projects were used. The values for some of the factors important for COEEMO were not provided in the published studies and therefore, their values were supposed and/or deduced from other factors and information reported, which could have an impact on the conclusions. COEEMO works on the values of the different factors as selected by development teams based on the subjective criteria for different levels within each factor. Inappropriate selection of the levels by the team for any factor could impact conclusion validity.

12.4.3 External Validity

This aspect of validity is related to the extent to which the outcomes of a study can be generalized and be of interest to other people (Runeson and Höst, 2009). During this study, two teams (Team-A & Team-B) from AUSIT applied treatments on two types of systems (i.e., ERP and Web-based) and therefore, results cannot be generalized to a larger population directly because of the limited teams and projects considered. In order to make the results more generalizable, further study would be needed on more heterogeneous samples where many IT teams working in organizations, preferably from different regions of the world, may apply the recommended artefacts of this study on different types of IT projects.

12.5 Implications and Future work

The results of this research suggest a range of implications for researchers and practitioners – a list of future tasks / actions and guidelines are given as follows:

a) In the current study, both COEEMO and PERT's variant-1 were applied to effort estimates that were provided by co-located teams, therefore, their effectiveness could also be explored for effort estimates provided by teams working on the same ASD project but from different locations.

b) COEEMO can also provide estimates on its own for ASD projects, but this aspect was not validated in the current study; therefore, an exploratory study is required to analyze COEEMO's effectiveness in this additional context.

c) In cases where an agile team uses COEEMO for their estimating activities and so gains confidence in the model's outcomes then a decrease of between 25-50% in the duration of time-boxed planning meetings per sprint/cycle is anticipated, as an offset benefit of using this model. This specific aspect should be explored to more fully determine the overall benefit of using COEEMO in estimation endeavours.

d) In the current study COEEMO was applied to a limited range of system types, i.e., ERP, Web based applications, mobile applications. Therefore, it is required to use COEEMO on a broader range of system types, e.g., Decision Support Systems, Mission-critical systems, Executive Information Systems, Internet-Of-Things and others, to better analyze its usefulness.

e) COEEMO should be applied in IT organizations in different geographical locations to analyze its effectiveness in terms of its various factors, e.g., culture, work environment, teams' relationships, stakeholder coordination, and others.

f) The COEEMO model could be integrated with applications that are already used by the IT industry in their ASD project management, e.g., JIRA, so that agile teams can more easily use COEEMO in their estimation activities through the same interface of applications with which they are comfortable.

g) PERT's variant-1 suggested estimates could be extended by using COEEMO to analyze its effectiveness with respect to accuracy improvements for agile and non-agile projects.

h) COEEMO can also provide adjusted estimates for non-agile projects, therefore empirical validation is required to analyze its benefits in terms of improvements in estimation accuracy for non-agile software projects.

12.6 Final Statement

The research conveyed in this thesis has targeted the issue of inaccuracy in effort estimation in general, but specifically in relation to agile development. Three solutions have been suggested to address this long-standing issue. There are expectations that the devised solutions will support teams to overcome in part the issue of inaccuracy in effort estimation which will ultimately assist the IT industry in reducing and better managing their effort and cost due to more frequent on-time delivery of their software products.

References

Abrahamsson, P., Fronza, I., Moser, R., Vlasenko, J., Pedrycz, W. (2011). Predicting development effort from user stories. Empirical Software Engineering & Measurement (ESEM). DOI. 10.1109/ESEM.2011.58.

Abrahamsson, P. and Koskela, J. (2004). Extreme programming: a survey of empirical data from a controlled Case study. Empirical Software Engineering, ISESE '04 Proceedings, pp. 73-82.doi. 10.1109/ISESE.2004.1334895

Adler, P.S., Clark, K.B., 1991. Behind the learning curve: a sketch of the learning process. Management Science 37 (3), 267-281.

ADT─ Anderson-Darling Test for Normality. (2018, November 15). Retrieved February 1, 2020, from https://www.spcforexcel.com/knowledge/basic-statistics/anderson-darling-test-for-normality

Agahi, H., and Quirk, J. (1998). Estimating Software Productivity and Cost for NASA Projects. Journal of Parametrics, pp. 59-71.

Agile Alliance. (2001). Principles behind the Agile Manifesto. Retrieved May 31, 2017, from http://www.agilemanifesto.org/principles.html

Ahimbisibwe, A., Daellenbach, U., and Cavana, R.Y. (2017) "Empirical comparison of traditional plan-based and agile methodologies: Critical success factors for outsourced software development projects from vendors' perspective", Journal of Enterprise Information Management, Vol. 30 Issue: 3, pp.400-453, https://doi.org/10.1108/JEIM-06-2015-0056

Ahimbisibwe, A., Cavana,R.Y., and Daellenbach, U.(2015). A contingency fit model of critical success factors for software development projects: A comparison of agile and traditional plan-based methodologies", Journal of Enterprise Information Management, Vol. 28 Issue: 1, pp.7-33, https://doi.org/10.1108/JEIM-08-2013-0060

Ahmed M.A., and Muzaffar Z., 2009. Handling imprecision and uncertainty in software development effort prediction: A type-2 fuzzy logic-based framework. Information and Software Technology 51. Pp. 640–654.

Albrecht, A.J. (1979) Measuring Application Development Productivity. Proceedings of IBM Applications Development Symposium, Monterey, 14-17 October 1979, 83.

Alostad, J. M., Abdullah, L. R. A., & Aali, L. S. (2017). A Fuzzy based model for effort estimation in scrum projects. International Journal of Advanced Computer Science 145 and Applications, 8(9), 270–277. Retrieved from www.ijacsa.thesai.org

Allen, P., Ramachandran, M., & Abushama, H. (2003). PRISMS: an approach to software process improvement for small to medium enterprisesIEEE. Symposium conducted at the meeting of the Quality Software, 2003. Proceedings. Third International Conference on

Ambler, S. W. (2002). Agile modeling. New York, John Wiley & Sons, Inc.

Amerongen, R. van. (2018, April 19). 7 Agile Estimation Techniques – beyond Planning Poker. Retrieved from https://technology.amis.nl/2016/03/23/8-agile-estimation-techniques-beyond-planning-poker/.

Andreou, A. S., and Papatheocharous, E. (2008). Software Cost Estimation using Fuzzy Decision Trees. 23rd IEEE/ACM International Conference on Automated Software Engineering, pp. 371 – 374.

Anderson, J.R., 1982. Acquisition of cognitive skill. Psychological Review 89 (4), 369-406.

Arora, M., Verma, S., and Kavita. (2018). An efficient effort and cost estimation framework for Scrum Based Projects. International Journal of Engineering & Technology, 7 (4.12) 52-57.

Argote, L. (1999). Organizational Learning: Creating, Retaining and Transferring Knowledge. Springer, New York.

Askin, R., Goldberg, J. (2001). Design and Analisys of Lean Production Systems. John Wiley & Sons, New York.

Attarzadeh, I., & Ow, S. H. (2010). Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks. In 2nd International Conference on Computer Engineering and Technology (ICCET1) (Vol. 3, pp. 487-491). IEEE.

Attarzadeh I. and Hock O. S. (2009). Software Development Effort Estimation Based on a New Fuzzy Logic Model. International Journal of Computer Theory and Engineering, Vol. 1, No. 4.

Augen, N.C. (2006). An Empirical study of using Planning Poker for User Story estimation. Proceeding of AGILE IEEE conference.

Awad, M.A. (2005).A Comparison between Agile and Traditional Software Development Methodologies. Report submitted as partial fulfilment for honours program of school of Computer Science and Software Engineering. University of Western Australia.

Azzeh, M., Neagu, D., and Cowling, P.I. (2010). Fuzzy grey relational analysis for software effort estimation, Empir Software Eng (2010), pp. 60-90, 2010.

Azzeh, M., & Nassif, A.B. (2016). A hybrid model for estimating software project effort from Use Case Points. Allied Soft Computing 49 981-989. http://dx.doi.org/10.1016/j.asoc.2016.05.008

B. (2021, February 6). 10 Characteristics of an Ideal Work Environment. OpEx Managers. Retrieved February 28, 2021, from https://opexmanagers.com/2020/09/01/characteristics-of-an-ideal-work-environment/

Badiru, A.B., 1992. Computational survey of univariate and multivariate learning curve models. IEEE Transactions on Engineering Management 39 (2), 176-188.

Basri, S., Kama, N., Sarkan, H.M, Adli, S., & Haneem, F. (2016). An Algorithmic-based Change Effort Estimation Model for Software Development. 23rd Asia-Pacific Software Engineering Conference, IEEE. DOI 10.1109/APSEC.2016.57

Bardsiri, V.K., Jawawi, D.N.A, Bardsiri, A.K., and Khatibi, E. (2013). LMES: A localized multi-estimator model to estimate software development effort. Engineering Applications of Artificial Intelligence.

Banker, D.R., Data, M.S., and Kemerer, F.C. (1987). Factors affecting software maintenance productivity: An exploratory study. Proceedings of the Eighth International Conference on Information Systems, 1987, p. 160-175.

Balich, I. K., & Martin, C. L. (2010). Applying a feed-forward neural network for predicting software development effort of short-scale projects. In Eighth ACIS International Conference on Software Engineering Research, Management and Applications (pp. 269-275). IEEE Computer Society. doi:10.1109/SERA.2010.41

Bathla R., Singh S., Vishav M., Gupta R. (2010). Innovative scenario in software development effort estimation based on a new fuzzy logic model. International Journal of Information Technology and Knowledge Management. Volume 2, No. 2, pp. 361-364.

Baumeister, H. (Eds.), Extreme Programming and Agile Processes in Software Engineering (Vol. 3092, pp. 139-147): Springer Berlin / Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-540-24853-8_16. doi:10.1007/978-3-540-24853-8-16

Bayer, S. and Highsmith, J. (1994). “RADical software development.” American Programmer 7(6): 35-42.

Beath, C. M., and Orlikowski, W. J., (1994). The contradictory structure of systems development methodologies: Deconstructing the IS-User relationship in information engineering. Information Systems Research, (5:4): 350-377.

Beck, K. (1999). “Embracing change with extreme programming.” IEEE Computer: 70-77.

Beck, K. (2000). Extreme programming explained: Embrace change. Reading, MA., Addison Wesley Longman, Inc.

Beck, K. and M. Fowler (2001). Planning extreme programming. New York, Addison-Wesley.

Benbasat, I., & Zmud, R. W. (1999). Empirical research in information systems: the practice of relevance. MIS Quarterly, 3-16.

Benediktsson, O., Dalcher, D., Reed, K., and Woodman, M. (2003). COCOMO-based Effort Estimation for Iterative and Incremental software development. Software Quality Journal, vol. 11, pp. 265-281.

Bhalerao, S., and Ingle, M. (2009). Incorporating Vital factors in agile estimation through algorithmic method. International Journal of Computer Science and Computer applications.

Biller, C. (2000). Adaptive Bayesian regression splines in semiparametric generalized linear models. Journal of Computational and Graphical Statistics, 122–140.

Bilgaiyan, S., Mishra, S., & Das, M. (2016). A review of software cost estimation in agile software development using soft computing techniques. 2016 2nd International Conference on Computational Intelligence and Networks (CINE), 112–117. https://doi.org/10.1109/CINE.2016.27

Bilgaiyan, S., Sagnika, S., Mishra, S., & Das, M. (2017). A systematic review on software cost estimation in agile software development. Journal of Engineering Science and Technology Review, 10(4), 51–64. https://doi.org/10.25103/jestr.104.08

Balzer, R., Cheetham, T.E. and Green, C. (1983), “Software technology in the: using a new paradigm”, Computer, November, pp. 39-45.

Bloomenthal, A. (2019, December 8). Ringing the Bell Curve. Retrieved from

Bloch, M., Blumberg, S. and Laartz, J. (2013). Delivering large-scale IT projects on time, budget and value. https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value [Accessed on 30-Oct-2017]

Boehm, B., Turner, R. (2004). Balancing Agility & Discipline. Addison-Wesley.

Boehm, B.W. (2000). Software Cost Estimation with COCOMO II. Prentice Hall.

Boehm, B. (1988). “A Spiral Model of Software Development and Enhancement.” IEEE Computer 21: 61-72.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77-101. https://doi.org/10.1191/1478088706qp063oa

Braz, M.R., and Vergilio, S.R. (2006). Software Effort Estimation on Use Cases. Proc. COMPSAC’06.

Briand, L.C., Emam , K. E., and Surmann, D., and Wieczorek, I. (1999). An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques, Fraunhofer Institute for Experimental Solhare Engineering IESE. Volume (1) pp.1-10.
https://www.investopedia.com/terms/b/bell-curve.asp [Accessed on 12-Jan-2020]

Buffardi, L.C., Fleishman, E. A., Morath, R. A., and McCarthy,P. M. (2000). Relationships between ability requirements and human errors in job tasks. Journal of Applied Psychology, pp. 551-564.

Caballero, E., Calvo-Manzano. (2012). J.A. A Practical Approach to Project Management in a Very Small Company. Retrieved from https://link.springer.com/chapter/10.1007/978-3-642-31199-4_28

Camci, A., and Kotnour, T. (2006) Technology complexity in projects: Does classical project management work? PICMET 2006 Proceedings, Turkey, pp. 2181–2186.

Carbonera1, C. E., Farias, K. , and Bischoff .(2018). Software development effort estimation: A systematic mapping study. IET Research Journals, pp. 1–14. ISSN 1751-8644 doi: 0000000000 www.ietdl.org

Clarke, P. and O'Connor R.V. The situational factors that affect the software development process: Towards a comprehensive reference framework, Journal of Information Software and Technology, Vol. 54, Issue 5, May 2012. pp. 433-447.

Charette, R. N. (2005). Why software fails. IEEE Spectrum, 42(9), 42-49.

Chulani, S., Boehm, B., and Steece, B. (1999). Multiple Regression to Bayesian Analysis for Calibrating COCOMO. Journal of Parametrics, vol. 15(2), pp. 175-188.

Chamberlain, P., Bonsiepe, G., Cross, N., Keller, I., Frens, J., Buchanan, R.,Jonas, W. (2007). Design research now: Essays and selected projects: Walter de Gruyter

Chen, T. (2009). Enhancing the yield competitiveness of a semiconductor fabrication factory with dynamic capacity re-allocation. Computers & Industrial Engineering 57 (3), 931-936.

Chow, T., Cao, D. (2007). A survey study of critical success factors in agile software projects. The Journal of Systems and Software 81 (2008) 961–971.

Chow, T., and Cao, D.-B. 2008. A Survey of Critical Success Factors in Agile Software Projects. The Journal of Systems and Software (81:6), pp. 961-971.

Cockburn, A. (1998). Surviving Object-Oriented Projects: A Manager's Guide, Addison Wesley Longman.

Cockburn, A. (2000a). "Selecting a project's methodology." IEEE Software 17: 64-71.

Cockburn, A. (2000b). Writing Effective Use Cases, The Crystal Collection for Software Professionals, Addison-Wesley Professional.

Cockburn, A. (2002). Agile software development. Boston, MA., Pearson Education

Coad, P., E. LeFebvre and J. De Luca (2000). Java Modeling In Color With UML: Enterprise Components and Process. Englewood Cliffs, NJ, Prentice Hall.

Cockburn, A. (2002). Agile Software Development, Addison-Wesley, Boston.

Cohn, M. (2010). Succeeding with Agile: Software Development using Scrum. Pearson Education.

Cohn, M. (2009). Succeeding with Agile: Software Development Using Scrum, Addison-Wesley Professional, Boston, MA (2009).

Cohn, M. (2005). Agile Estimating and Planning, Prentice Hall, New Jersey, USA, (2005).

Cohn, M. (2004). User Stories Applied: For Agile Software Development, 1 ed., Addison-Wesley, Boston, MA (2004).

Cohn, User Stories Applied: For Agile Software Development, Addison R Xesley, Doston, _A, 2004

Commeyne, C., Abran, A., and Djouab, R. (2016). Effort Estimation with Story Points and COSMIC Function Points - An Industry Case Study. Software Measurement News. Vol. 21, No. 1, pages 25-36.

Conboy, K., Coyle, S., Wang, X., and Pikkarainen, M. (2011). People Over Process: Key People Challenges in Agile Development. http://hdl.handle.net/10344/639.

Coplien, J.O. and Harrison, N.B. (2005). Organisational Patterns of Agile Software Development, Pearson Prentice Hall, New Jersey.

Collaris, Remi-Armand and Dekker, E. (2011). Early Estimation with Stakeholder. Agile Records. Volume (1) pp. 1-5.

Cook, J.A., 1991. Competitive model of the Japanese firm. Journal of Policy Modelling 13 (1), 93-114.

Cross, N. (2007). Forty years of design research. Design Studies, 28(1), 1-4. doi: http://dx.doi.org/10.1016/j.destud.2006.11.004

Cisneros-Puebla, Cesar A.; Faux, Robert & Mey, Günter (2004). Qualitative researchers—stories told, stories shared: The storied nature of qualitative research. An introduction to the special issue: FQS Interviews I. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research, 5(3), Art. 37, http://nbn-resolving.de/urn:nbn:de:0114-fqs0403370 [Accessed: 27 March, 2021].

Creswell, J. W., and Clark, V. L. P. 2007. Designing and Con- ducting Mixed Methods Research, Thousand Oaks, CA: Sage Publications.

Cuadrado-Gallego, J.J., Rodríguez-Soria, R., and Martín-Herrera, B.(2010). Analogies and Differences between Machine Learning and Expert Based Software Project Effort Estimation. 11th ACIS International Conference on Software Engineering, Artificial Intelligence. Networking and Parallel/Distributed Computing, London. UK. pp. 269-276. doi: 10.1109/SNPD.2010.47.

Dantas, E., Perkusich, M., Dilorenzo, E., Santos, D.F.S., Almeida, H., and Perkusich, A. (2018). Effort Estimation in Agile Software Development: An Updated Review. International Journal of Software Engineering and Knowledge Engineering, Vol. 28, Nos. 11&12, 1811–1831. DOI: 10.1142/S0218194018400302.

Dave, V. S., & Dutta, K. (2014). Neural network based modelsforsoftware effort estimation: A review, Artificial Intelligence Review. International Review, 42(2), 295–307.

DeBakker, K., Boonstra, A., and Wortmann, H. (2010). Does risk management contribute to IT project success? A meta-analysis of empirical evidence. Int. J. Proj. Manag. 28, 493–503. http://dx.doi.org/10.1016/j.ijproman.2009.07.002.

Diev,S. (2006). Use cases modeling and software estimation: Applying Use Case Points. ACM Software Engineering Notes. Volume (31) pp.1-4.

Doody, Owen & Noonan, Maria (2013). Preparing and conducting interviews to collect data. *Nurse Researcher*, *20*(5), 28-32.

Downey, S. and Sutherland, J. (2013). Scrum metrics for hyper productive teams: how they fly like fighter aircraft. System Sciences (HICSS), 46th Hawaii International Conference on. IEEE, 2013.

Dragicevic, S., Celar, S., and Turicc, M. (2017). Bayesian network model for task effort estimation in agile software development. Journal of Systems and Software, Journal of Systems and Software, volume 127, Pages 109-119. http://dx.doi.org/10.1016/j.jss.2017.01.027

Dingssøyr, T., Dyba, T., and Moe, N.B. (2010). Agile Software Development: Current Research and Future Directions. ISBN 978-3-642-12574-4. DOI 10.1007/978-3-642-12575-1.

Dönmez, D., & Grote, G. (2018). Two sides of the same coin – how agile software development teams approach uncertainty as threats and opportunities. Information and Software Technology, 93, 94–111. https://doi.org/10.1016/j.infsof.2017.08.015

DSDM Consortium (1997). Dynamic Systems Development Method, version 3. Ashford, Eng., DSDM Consortium.

Duka, D. (2013). Adoption of Agile Methodology in Software Development. MIPRO, May 20-24. Opatija, Croatia.

Eckstein, J., Shukla, R., and Marwala, T. (2013). Modeling of Software Development Effort, Special Issue: Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM 2013, pp: 52-57, July 2013.

ECMS- Faculty of Engineering, Mathematicall and Computer Science, AUT. Retrieved on 07-Feb-2019 from https://www.aut.ac.nz/study/study-options/engineering-computer-and-mathematical-sciences/courses/bachelor-of-computer-and-information-sciences.

Eder, W.E., Hubka, V., and Hosnedl, S. 2007. Design Engineering: A Manual for Enhanced Creativity. CRC Press

Faria, P. and Miranda, E. (2012). Expert Judgment in Software Estimation during the Bid Phase of a Project – An Exploratory Survey. Joint Conference of the 22nd International Workshop on Software Measurement and the 2012 Seventh International Conference on Software Process and Product Measurement. DOI 10.1109/IWSM-MENSURA.2012.27

FERNÁNDEZ-DIEGO, M., 1 , MÉNDEZ, E.R., GONZÁLEZ-LADRÓN-DE-GUEVARA, F., ABRAHÃO, S., and INSFRAN, E. (2020) . An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review. IEEEAccess.10.1109/ACCESS.2020.3021664

Fitsilis, P., Kameas, A. and Anthopoulos, L. (2011). Classification of Software Projects' Complexity. W.W. Song et al. (eds.), Information Systems Development. DOI 10.1007/978-1-4419-7355-9_13

Fitriani, W.R., Rahayu, P., and Sensuse, D.I. (2016). Challenges in agile software development: A systematic literature review. *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Malang, 2016, pp. 155-164, doi: 10.1109/ICACSIS.2016.7872736.

Finnie, G. R., Wittig, G. E., & Desharnais, J. M. (1997). A comparison of software effort estimation techniques: using function points with neural networks, case-based reasoning and regression models. Journal of Systems and Software, 39(3), 281–289. doi:10.1016/S0164-1212(97)00055-1

Fleishman, E. A., and Quaintance, M. K. (1984). Taxonomies of human performance: The description of human tasks, Orlando, FL: Academic Press, 1984.

Flanagan, D. P., Harrison, P. L. (2012). Contemporary Intellectual Assessment: Theories, Tests, and Issues, Guilford Press, 2012.

Fleishman, E. A., Kinkade, R. G. and Chambers, A. N. (1968). Development ofa taxonomy of human performance: A review of the first year's progress (No. AIR-726-11/68-TPRI), American Institutes for Research, Pittsburgh, 1968.

Flyvbjerg, B., and Budzier, A. (2011). Why Your IT Project May Be Riskier than You Think. Harvard Business Review, vol. 89, no. 9, September 2011, pp. 23-25.

Fontana, R.M., Rheinehr, S., and Malucelli, A. (2015). "Agile Compass: A Tool for Identifying Maturity in Agile Software-Development Teams", IEEE Software, vol. 32.6, Nov-dec. 2015, pp. 20-23, doi: 10.1109/MS.2015.135

Foss, T., Stensrud, E., Kitchenham, B., and Myrtevit, I. (2002). A Simulation Study of the Model Evaluation Criterion MMRE. The Norwegian School of Management. Keele University.ISSN: 0807-3406.

Frost, J. (2019, October 8). Boxplots vs. Individual Value Plots: Graphing Continuous Data by Groups. Statistics By Jim. https://statisticsbyjim.com/basics/graph-groups-boxplots-individual-values/

Fuggetta, A., and Di Nitto, E. (2014). Software Process Proceedings of the on Future of Software Engineering (pp. 1–12). New York, NY, USA: ACM. Doi:10.1145/2593882.2593883

Galvin, R. (2015). How many interviews are enough? Do qualitative interviews in building energy consumption research produce reliable knowledge? JUST SOLUTIONS CAMBRIDGE WORKING PAPER 2014B: Published in The Journal of Building Engineering, 2015.

Gandomani, T.J., Wei, K.T., & Binhamid, A.K. (2014). A Case Study Research on Software Cost Estimation Using Experts' Estimates, Wideband Delphi, and Planning Poker Technique. International Journal of Software Engineering and its Applications, Vol. 8, pp. 173-182. DOI: http://dx.doi.org/10.14257/ijseia.2014.8.11.16

Gelling ,L.(2015).Applying for ethical approval for research: the main issues.Nursing Standard, 30(20), 40-44.doi:10.7748/ns.30.20.40.s46.

Grapenthin S, Poggel S, Book M, Gruhn V (2014) Facilitating task breakdown in sprint planning meeting 2 with an interaction room: an experience report. In: Software engineering and advanced applications (SEAA), 2014 40th EUROMICRO Conference on,IEEE, pp 1–8

Grimstad, S., and Jorgensen, M. (2006).A Framework for the Analysis of Software Cost Estimating Accuracy. ACM ISESE'06, pages (58-65).doi. 1-59593-218-6/06/0009

Green, M. D. (2014) http://www.sitepoint.com/3-powerful-estimation-techniques-for-agile-teams/ [Accessed on 15 Nov 2014]

Gregory, P., Barroca, L., Taylor, K., Salah, D., & Sharp, H. (2015). Agile Challenges in Practice: A Thematic Analysis. Proceedings of the 16th International Conference on Agile Software Development, Helsinki. Forthcoming in LNCS.

Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. MIS Quarterly, 37(2), 337-355.

Grenning,J.(2002). Planning Poker or How to avoid analysis paralysis while release planning. Hawthorn Woods: Renaissance Software Consulting, vol. 3.

Gruschke, T.M., Jørgensen, M.: The role of outcome feedback in improving the uncertainty assessment of software development effort estimates. ACM TOSEM 17(4), 201–2035 (2008)

Hamdan, A., Alwadan, T. and Ababneh, J. (2013). Agile Software Methodologies: Strength and Weakness. International Journal of Engineering Science and Technology (IJEST). Vol. 5 No.03

Hamade, R., Artail, H., Jaber, M., 2007. Evaluating the learning process of mechanical CAD students. Computers & Education 49 (3), 640-661.

Hamouda, A.E.D. (2014). Using Agile Story Points as an Estimation Technique in CMMI Organizations. Agile Conference. IEEE. P.16-23.

Hanakawa N., Morisaki S, Matsumoto K. (1998). A Learning Curve Based Simulation Model for Software Development. Proceedings of 20th International Conference on Software Engineering, 350-359.

Haines, T., Idemudia, E.C., and Raisinghani, M. S. (2017). The Conceptual Model for Agile Tools and Techniques. American Journal of Management Vol. 17(3).

Halkjelsvik, T., and Jørgensen, M. (2018). Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life. Springer International Publishing, pp. 9–10.

Hevner, A.R., March, S.T., Park, J., and Ram, S. 2004. Design Science in Information Systems Research. MIS Quarterly
28, 1, 75-105.

Hevner, A., & Chatterjee, S. (2010). Design science research in information systems: Springer

Hevner, A. R. (2007). A three cycle view of design science research. Scandinavian journal of information systems, 19(2), 4.

Highsmith, J. (2004). Agile Project Management: Creating Innovative Products. Boston, MA: Pearson Education, Inc.
http://www.recruitit.co.nz/auckland-it-salaries-update-confirmation-of-download[Accessed on 15 Jun 2015]

Highsmith, J. A. (2000). Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. New York, NY, Dorset House Publishing.

Holt, Amanda (2010). Using the telephone for narrative interviewing: A research note. *Qualitative Research*, *10*(1), 113-121.

Holmes, A., Illowsky, B., & Dean, S. (2015, March 31). Retrieved January 13, 2020, from https://opentextbc.ca/introbusinessstatopenstax/chapter/skewness-and-the-mean-median-and-mode/

Huang, J.S., Chiu, N.H., and Chen, L.W. (2008). Integration of the grey relational analysis with genetic algorithm for software effort estimation. European Journal of Operational Research.188(3): p. 898-909.

Hussain, I., Kosseim, L., and Ormandjieva, O. (2013). Approximation of COSMIC functional size to support early effort estimation in Agile. Data & Knowledge Engineering, Elsevier. p. 2-14.

Hu, Q., T. R., and Hertz, B. (1998). Software Cost Estimation Using Economic Production Models. Journal of Management Information Systems / summer. Vol.15, No. 1, pp. 143-163.

Huang, S. J., and Chiu, N.H. (2007). Applying fuzzy neural network to estimate software development effort", Springer Science, 2007.

Idri, A., Amazal, F., & Abran, A. (2015). Analogy-based software development effort estimation: a systematic mapping and review, J. Inf. Softw. Technol. 58 (2015) 206–230, doi:10.1016/j.infsof.2014.07.013

Ideri, A., Abran, A., & Mbarki, S. (2006). An Experiment on the Design of Radial Basis Function Neural Networks for Software Cost Estimation. In Information and Communication Technologies, ICTTA (pp. 1612 -1617). IEEE.

Idri, A., Amazal, F., & Abran, A. (2016). Accuracy comparison of analogy-based software development effort estimation techniques, Int. J. Intell. Syst. 31 (2) 128–152, doi: 10.1002/int.21748

Ionescu, V. (2017). An approach to software development effort estimation using machine learning. 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). pp. 197-203. doi: 10.1109/ICCP.2017.8117004.

Ionescu, V., Demian, H., Czibula,I.G. (2017). Natural Language Processing and Machine Learning Methods for Software Development Effort Estimation. Studies in Informatics and Control, ISSN 1220-1766, vol. 26(2), pp. 219-228.

Inayat, E., Salim, S. S., Marczak, S., Daneva, M., and Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. Computers in Human Behavior. 51, 915 – 929. DOI: http://dx.doi.org/10.1016/j.chb.2014.10.046

Intaver Institute Inc. (2006) Quantitative Risk Analysis with Microsoft Project. Intaver Institute Inc. Retrieved on 11 June 2017 from URL http://www.intaver.com/

Islam.S., et al. (2009) Goal and Risk Factors in Offshore Outsourced Software Development From Vendor's Viewpoint. Fourth IEEE International Conference on Global Software Engineering.

Ishak, R. (2016, July 8). 19 Signs Your Work Environment Is Toxic & Affecting You Negatively. Bustle. Retrieved February 28, 2021, from https://www.bustle.com/articles/170826-19-signs-your-work-environment-is-toxic-affecting-you-negatively

Ivan, I., & Despa, M. L. (2016). Estimating maintenance cost for web applications. Informatica Economica, 20(4/2016), 62–75. https://doi.org/10.12948/issn14531305/20.4.2016.06

Iwata, K., Anan, Y., Nakashima, T., & Ishii, N. (2009). Using an artificial neural network for predicting embedded software development effort. In International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing (pp. 275-280). IEEE Computer Society. doi:10.1109/ SNPD.2009.49

Janes, A. and Succi, G. (2012).The dark side of agile software development. Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, 19–26 October, Association for Computing Machinery, Tucson, Arizona.

Jahr, M. (2014). A Hybrid Approach to Quantitative Software Project Scheduling Within Agile Frameworks. Project Management Journal, Vol.45, No.3, pp. 35-45. DOI: 10.1002/pmj.21411

Javdani, T., Zulzalil, H., Ghani, A.A., and Sultan, A.M.(2012). On the current measurement practices in agile sofware development, International Journal of Computer Science Issues, vol. 9 (2012) pp.127-132.

Jørgensen, M. (2013). The influence of selection bias on effort overruns in software development projects. http://dx.doi.org/10.1016/j.infsof.2013.03.001

Jørgensen, M., and Grimstad, S. (2011). The impact of irrelevant and misleading information on software development effort estimates: a randomized controlled field experiment. IEEE TSE 37(5), 695–707.

Jørgensen, M., & Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. IEEE Transactions on Software Engineering, Vol.33, No.1.

Jørgensen, M., Boehm, B., and Rifkin, S.(2009). Software Development Effort Estimation: Formal Models or Expert Judgement. IEEE Software, vol 26 no. 2, pp. 14-19.

Jørgensen, M., and Shepperd, M. (2007). "A systematic review of software development cost estimation studies,"IEEE Trans. Softw. Eng., vol. 33, no. 1, pp. 33–53, 2007.

Jørgensen, M. (2007). A critique of how we measure and interpret the accuracy of software development effort estimation. First International 727 Workshop on Software Productivity Analysis and Cost Estimation. pp. 15–21.

Jørgensen, M. (2007). Forecasting of software development work effort: Evidence on expert judgement and formal models. International Journal of Forecasting 23 (2007) 449-462. doi: 10.1016/j.ijforecast.2007.05.008.

Jørgensen, M., and Sjoberg, D. (2006). Expert Estimation of Software Development Work. Software Evolution and Feedback: Theory and Practice. Wiley.

Jørgensen,M. (2005). Practical Guidelines for Expert-Judgement-Based Software Effort Estimation. IEEE Software. vol. 22, pp. 57-63, 2005.

Jørgensen, M. (2005). Practical guidelines for expert-judgment-based software effort estimation". Software, IEEE, 2005. 22(3): p. 57-63.

Jørgensen, M. (2004). A Review of Studies on Expert Estimation of Software Development Effort. J. Systems and Software. vol. 70, pp. 37-60.

Jørgensen, M. (2003). Top-down and bottom-up expert estimation of software development effort. Information and Software Technology 46 (2004) 3–16. doi:10.1016/S0950-5849(03)00093-4

Jørgensen, M., and Sjøberg, D. I. K. 2002. Impact of experience on maintenance skills. Journal of Software Maintenance and Evolution: Research and Practice 142(2): 123-146

Jørgensen, M. (2002). A review of studies on expert estimation of software development effort. The Journal of Systems and Software 70 (2004) 37–60. doi:10.1016/S0164-1212(02)00156-5

Jones, C. (1996) By popular demand: software estimating rules of thumb. Computer, 29, 116-118.

Jun, LI and Jianming, LIN and Yongqin, JIN and Qingzhang, CHEN (2008) Development of the Decision Support System for Software Project Cost Estimation. International Symposium on Information Science and Engineering, (1) pp. 1-4.

Kalermo, J. and Rissanen, J. (2002), "Agile software development in theory and practice", Master's thesis, Department of Computer Science and Information Systems, University of Jyvaskyla, Jyvaskyla.

Kang S, Choi O, Baik J (2010) Model-based dynamic cost estimation and tracking method for agile software development. In: Computer and information science (ICIS), 2010 IEEE/ACIS 9th International Conference on, IEEE, pp 743–748.

Kautz, K. (2009). Customer and User Involvement in Agile Software Development. P. Abrahamsson, M. Marchesi and F. Maurer, (eds.), XP2009, Springer, Pula, Sardinia, Italy.

Keaveney, S., and Conboy, K. (2006). Cost estimation in agile development projects.

Keung, J. (2009). Software Development Cost Estimation using Analogy: A Review. Proceedings of Australian Software Engineering Conference.

Khuat, T. T., & Le, H. (2016). Optimizing parameters of software effort estimation models using directed artificial bee colony algorithm. Informatica, 40, 427–436. Retrieved from http://www.informatica.si/index.php/informatica

Kitchenham, B. A., Pickard, L. M., MacDonell, S. G., and Shepperd, M. J. (2001). What accuracy statistics really measure. IEE Proceedings-Software.725 148 (3) 81–85.

Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004). Evidence-based software engineeringIEEE Computer Society. Symposium conducted at the meeting of the Proceedings of the 26th international conference on software engineering.

Kitchenham, B. (2004a). Procedures for performing systematic reviews. Keele, UK, Keele University, 33(2004), 126. Kitchenham, B. (2004b). Procedures for performing systematic reviews. Keele, UK, Keele University, 33, 2004.

Kirmani MM, Wahid A. Impact of modification made in re-UCP on software effort estimation. J Softw Eng Appl. 2015; 8(06): 276- 289.

Korkala, M., Pikkarainen, M., and Conboy, K. Distributed Agile Development: A Case Study of Customer Communication Challenges. P. Abrahamsson, M. Marchesi and F. Maurer, (eds.), XP2009, Springer, Pula, Sardinia, Italy, 2009.

Kozlowski, S.W.J., & Bell, B.S. (2003). Work Groups and Teams in Organizations, in Handbook of Psychology (Vol. 12): Industrial and Organizational Psychology W.C. Borman, D.R. Ilgen and R.J. Klimoski (eds.). New York: Wiley, pp. 333-375.

Kocaguneli.E. (2012). A Principled Methodology: A Dozen Principles of Software Effort Estimation. Ph.D Dissertation in Computer Science, West Virginea University, USA.

KPMG (2017). https://home.kpmg.com/content/dam/kpmg/nz/pdf/01816%20PMS%20ReportCompressed.pdf

Kumar G., Bhatia P.K. (2016) Empirical Assessment and Optimization of Software Cost Estimation Using Soft Computing Techniques. https://doi.org/10.1007/978-981-10-1023-1_12

Laird, L., and Brennan, M. (2006) Software Measurement and Estimation. A Practical Approach. John Wiley & Sons, Inc., Hoboken, New Jersey.

Lefley, M., and Shepperd, M. J. (2003). Using Genetic Programming to Improve Software Effort Estimation Based on General Data Sets. LNCS, Genetic and Evolutionary Computation. ISBN: 978-3-540-40603-7, page-208, GECCO.

Levy, Z. (1990). Estimating the Effort in the early stages of Software Development. PhD Thesis. London school of Economics.

Lee, M.D., Steyvers, M., Young, M.D., and Miller, B. (2012). Inferring Expertise in knowledge and Prediction Ranking Tasks. Topics in Cognitive Science, TOPICS. DOI: 10.1111/j.1756-8765.2011.01175.x

Lee, A. S. 1999. "Rigor and Relevance in MIS Research: Beyond the Approach of Positivism Alone," MIS Quarterly (23:1), pp. 29-34.

Leung, H., and Fan, Z. (2002). Software Cost Estimation. Handbook of Software Engineering and Knowledge Engineering, pp. 307-324.https://doi.org/10.1142/9789812389701_0014.

Li, Q. (2013). A novel Likert scale based on fuzzy sets theory. The Journal of Expert Systems with Application 40(2013) 1609-1618. http://dx.doi.org/10.1016/j.eswa.2012.09.015

Licorish, A. S. (2007). Tool Support for Social Risk Mitigation in Agile Projects: Thesis MCIS submitted at AUT. (Sherlock Anthony Licorish student name)

Little, T.E., and Verhoef,C.(2016). To Estimate or #NoEstimates, that is the Question, 2016.

Liming, W. (1997). The Comparison of the Software Cost Estimating Methods. University of Calgary.

Li, J. & Wong, W.K. (2009). A semi-parametric analysis for identifying scleroderma patients responsive to an anti-fibrotic agent. Contemporary Clinical Trials 30, 105–113.

Litoriya, R., Kothari, A. (2013). An Efficient Approach for Agile Web Based Project Estimation: AgileMOW. International Journal of Computer Science and Computer applications.

Logue, K., McDaid, K., and Greer, D. (2007). Allowing for Task Uncertainties and Dependencies in Agile Release Planning. 4th Proceedings of the Software Measurement European Forum, 2007.

Lopez-Martin C., 2011. A fuzzy logic model for predicting the development effort of short scale programs based upon two independent variables. Applied Soft Computing. Pp 724-732.

López-Martín, C. (2014). (2014) Predictive accuracy comparison between neural networks and statistical regression for development effort ofsoftware projects. Applied Soft Computing. doi:10.1016/j.asoc.2014.10.033

Lubashevsky, A. (1996). Living with function points. in Network Operations and Management Symposium. IEEE.

Macdonald, J. (2005). The Impact of Project Planning Team Experience on Software Project Cost Estimates. Empirical Software Engineering, 10, 219–234

MacDonell, S.G., and Shepperd, M.J. (2001). Combining techniques to optimize effort predictions in software project management. The journal of Systems and Sofwtare 66 (2003) 91-98. doi: 10.1016/S0164-1212(02)00067-5.

Mahnic, V. (2011). A Case Study on Agile Estimating and Planning using Scrum. Electronics & Electrical Engineering. ISSN 1392-1215.No. 5(11)

Mahnic, V., and Hovelja, T. (2012). On using planning poker for estimating user stories, Journal of Systems and Software. 85 (9). 2086–2095.

Mann, C., and Maurer, F. A case study on the impact of Scrum on overtime and customer satisfaction, ADC 05, IEEE Computer Society, Denver, 2005.

Marchesi and Succi, 2002 proceeding of the 3rd International Conference on eXtreme Programming and Flexible Processes in Software Engineering (XP2002)

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. Decision support systems, 15(4), 251-266.

McCracken, D.D. and Jackson, M.A. (1982), “Life-cycle concept considered harmful”, ACM Software Engineering Notes, April, pp. 29-32.

Meli, R., and Santillo, L. (1999). Function Point Estimation Methods: A Comparative Overview. Data Processing.

Merisalo-Rantanen, H., Tuure, and T., Matti, R. (2005). Is extreme programming just old wine in new bottles: a comparison of two cases. Journal of Database Management 16 (4) 41–61.

Mendes, E., Mosley, N., and Counsell, S. (2003). Early Web Size Measures and Effort Prediction for Web Cost Imation .Proc. Ninth Int’l Software Metrics Symp., pp. 18-39.

Mendes, E., and Kitchenham, B. (2004). Further Comparison of Cross Company and Within-Company Effort Estimation Models for Web Applications. Proc. 10th Int’l Software Metrics Symp. pp. 348- 357.

Moe, N.B., Dingsoyr, T., and Dyba, T. 2008. "Understanding Self-Organizing Teams in Agile Software Development." in Proceedings of the 19th Australian Conference on Software Engineering, Aswec 2008. IEEE Xplore digital library, pp. 76-85.

Moløkken, k. (2002). Expert estimation of Web-development effort: Individual biases and group processes (Master Thesis), in Department of Informatics. University of Oslo.

Mohammed, S., Ferzandi, L., and Hamilton, K. 2010. "Metaphor No More: A 15-Year Review of the Team Mental Model Construct," Journal of Management (36:4), pp. 876-910.

Moløkken, K., and Jørgensen, M. "A Review of Surveys on Software Effort Estimation," Proc. IEEE Int'l Symp. Empirical Software Eng. (ISESE 2003), pp. 223-230, 2003.

Moløkken-Østvold, K., Haugen, N. C., and Benestad, H. C. (2008). Using planning poker for combining expert estimates in software projects, Journal of 730 Systems and Software. 81 (12). 2106–2117.

Morgenshtern, O., Raz, T., and Dvir, D. (2007). Factors affecting duration and effort estimation errors in software development projects. Information and Software Technology 49, 827–837.

Mukker, A.R.,Singh, L., Mishra, A.K.(2014).Systematic Review of Metrics in Software Agile Projects. COMPUSOFT, An international journal of advanced computer technology, vol. 3(2), February -2014.

Moharreri, K., Sapre, A.V, Ramanathan, J., and Ramnath, R. (2016). Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments. IEEE 40th Annual Computer Software and Applications Conference (COMPSAC)

Morski, J., & Miler, J. (2002). Towards an integrated environment for risk management in software distributed software projects. 7th European Conference on Software Quality, Helsinki, Finland.

Mishra, P., Pandey, C. M., Singh, U., Gupta, A., Sahu, C., & Keshri, A. (2019). Descriptive statistics and normality tests for statistical data. Annals of cardiac anaesthesia, 22(1), 67–72. doi:10.4103/aca.ACA_157_18.

Misra, S., Fantazy, K., and Akhter, M. (2012). Agile software development practices: evolution, principles, and criticisms. International Journal of Quality & Reliability Management. Vol. 29 No. 9, pp. 972-980. DOI 10.1108/02656711211272863

Miller, Carol (1995). In-depth interviewing by telephone: Some practical considerations. Evaluation and Research in Education, 9(1), 29-38.

Minku, L.L., and Yao, X.(2017). Which models of the past are relevant to the present? A software effort estimation approach to exploiting useful past models. Autom Softw Eng. 24:499–542. DOI 10.1007/s10515-016-0209-7

MGS, https://www.mountaingoatsoftware.com/agile/scrum/meetings/daily-scrum [Accessed on 22 December, 2019].

Mummolo, Giovanni. (1997). Measuring uncertainty and criticality in network planning by PERT-path technique. International Journal of Project Management, Volume (15) pp.377-387.

Muzaffar Z. And Ahmed M.A. (2010). Software development effort prediction: A study on the factors impacting the accuracy of fuzzy logic systems. Information and Software Technology 52 (2010) 92–109.

Munialo, S. W., and Muketha, G. M. (2016). A Review of Agile Software Effort Estimation Methods. International Journal of Computer Applications Technology and Research. Volume 5–Issue 9, 612-618, ISSN: -2319–8656.

Myrtveit, I., and Stensrud, E. (2012). Validity and reliability of evaluation procedures in comparative studies of effort prediction models. Empirical Software Engineering 17(1-2): 2.

Nassif, A. B., Azzeh, M., Banitaan, S. (2017). Robust Rank Aggregation Method for Case-Base Effort Estimation. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE). 978-1-5090-5538-8/17

Nassif, A. B., Capretz, L. F., and Ho, D. (2010). Enhancing Use Case Points Estimation Method Using Soft

Computing Techniques. Journal of Global Research in Computer Science, 1 (4), 12-21.

Nassif, A. B., Ho, D., Capretz, L. F. (2013). Towards an early software estimation using log-linear regression and a multilayer perceptron model. Journal of Systems and Software. Volume 86, Issue 1. Pages 144-160.ISSN 0164-1212. 10.1016/j.jss.2012.07.050.

Nembhard, D.A., Osothsilp, N., 2002. Task complexity effects on between-individual learning/forgetting variability. International Journal of Industrial Ergonomics 29 (5), 297-306.

Nembhard, D.A., Uzumeri, M.V., 2000a. An individual-based description of learning within an organization. IEEE Transactions on Engineering Management 47 (3), 370e378.

Nisar M.W., Wang Y-J., Elahi M., Khan I.A., 2009.Software development effort estimation using fuzzy logic. Information Technology Journal.

Nguyen-Cong, D., Tran-Cao, De Tho, C. (2013). A Review of Effort Estimation Studies in Agile, Iterative and Incremental Software Development. IEEE RIVF International Conference on Computing & Communication Technologies – Research, Innovation, and Vision for the Future (RIVF).

Ngai, E.W.T, Sulk, F.F.C., and Lo, S.Y.Y. (2008). Development of an RFID-based sushi management system: The case of a conveyor-belt sushi restaurant. International Journal of Production Economics Volume 112, Issue 2, April 2008, Pages 630-645

Ngai, E.W.T., & Wat, F.K.T. (2004). Fuzzy decision support system for risk analysis in e-commerce development. Decision Support Systems 40 (2005) 235– 255. doi:10.1016/j.dss.2003.12.002

Nickolett, C. (2007). Business Savvy, IT Smart Earned Value Accuracy through Estimating. Comprehensive Consulting Solutions, Inc USA. Retrieved on 11 June 2017 from URL http://www.comp-soln.com/chipn.html.

Nunamaker, J.,Twyman, W.T., & Giboney, S.J.(2013). Breaking out of the Design Science Box: High-Value Impact through Multidisciplinary Design Science Programs of Research. Proceedings of the Nineteenth Americas Conference on Information Systems, Chicago, Illinois, August 15-17, 2013.

Nunamaker, J.F., and Briggs, R.O. (2011). Toward a Broader Vision for Information Systems. ACM Transactions on Management Information Systems (2:4), pp 20: 01-12.

Nunamaker, J., Chen, M., & Purdin, T. D. M. (1991).Systems development in information systems Research. Journal of Management Information Systems (7:3), winter 1991, pp.89-106.

Nunamaker, J., Chen,J. F., Purdin, T. D. (1990). Systems development in information systems research. Journal of Management Information Systems, 7(3), 89-106.

Ochodek, Mirosław and Alchimowicz, Bartosz and Jurkiewicz, Jakub and Nawrocki, Jerzy (2010) Reliability of Transaction Identification in Use Cases, Pozna´n University of Technology. Volume (2) pp.1-8.

Ochodek, M., Nawrocki, J., & Kwarciak, K. (2011). Simplifying effort estimation based on Use Case Points. Information and Software Technology, 53(3) p. 200-213.

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009). Outline of a design science research processACM. Symposium conducted at the meeting of the Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology.

Opdenakker, Raymond (2006). Advantages and disadvantages of four interview techniques in qualitative research. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, *7*(4), Art. 11, http://nbn-resolving.de/urn:nbn:de:0114-fqs0604118 [Accessed: 27 March 2021].

Palmer, S. R. and J. M. Felsing (2002). A practical guide to feature-driven development. Upper Saddle River, NJ, Prentice-Hall Inc.

Panda, A., Satapathy, S.M., Rath. S.K. (2015). Empirical Validation of Neural Network Models for Agile

Software Effort Estimation based on Story Points. Procedia Computer Science 57 (2015) 772 – 781. 3rd International Conference on Recent Trends in Computing (ICRTC-2015). doi: 10.1016/j.procs.2015.07.474.

Patton, Michael Q. (2002). *Qualitative research and evaluation methods* (3rd ed.). Thousand Oaks, CA: Sage.

Pananiswami, S., Bishop, R.C., 1991. Behavioral implications of the learning curve for production capacity analysis. International Journal of Production Economics 24 (1-2), 157-163.

Pendharkar, P. C., Subramanian, G. H., and Rodger, J. A. (2005). A probabilistic model for predicting software development effort, IEEE Transactions on Software Engineering 31 (7) 615–624.

Pengelly, A. (1995). Performance of effort estimating techniques in current development environments. Software Engineering Journal, (9) pp. 1-9.

Peffers, K., and Tuunanen, Rothenberger, M.A., & Chatterjeer S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems / Winter 2007–8, Vol. 24, No. 3, pp. 45–77. DOI 10.2753/MIS0742-1222240302

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J. (2006). The design science research process: a model for producing and presenting information systems research Symposium conducted at the meeting of the Proceedings of the first international conference on design science research in information systems and technology (DESRIST 2006)

Pereira, D.V., Corrêa, G.M., Silva, F.Q.B, & Ribeiro, D.M. (2017). Team Maturity in Software Engineering Teams: a Work in Progress. IEEE/ACM 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE).DOI 10.1109/CHASE.2017.2

Popli, R., Chauhan, N., and Sharma, H. (2014). Prioritizing user stories in agile environment. Issues and Challenges in Intelligent Computing Techniques (ICICT) International Conference on. IEEE, 2014.

PMI-Pulse, Project Management Institute. https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/pulse-of-the-profession-2017.pdf [Access on 12-09-2017]

Popli, R., Chauhan, N. (2014). Cost and Effort Estimation in Agile Software Development. International Conference on Reliability, Optimization and Information Technology -ICROIT 2014.

Popli, R., Chauhan, N. (2014). Agile Estimation Using People & Project Related Factors. International Conference on Computing for sustainable Global Development (INDIACom).

Prikladnicki, R., Dittrich, Y., Sharp, H., De Souza, C., Cataldo, M., & Hoda, R. (2013). Cooperative and Human Aspects of Software Engineering: CHASE 2013. SIGSOFT Softw. Eng. Notes, 38(5), 34–37. doi:10.1145/2507288.2507321

Pressman, R. S. and Maxim, B. (2014). Software engineering: a practitioner's approach. McGraw-Hill Science/Engineering/Math.

Raccoon, L. B. S. (1996). A Learning Curve Primer for Software Engineers. Software Engineering Notes vol 2! no 1. ACM SIGSOFT.

Raslan, A.T., Ramadan, N., and Hefny, H.A. (2015).Effort Estimation in Agile Software Projects Using Fuzzy logic and Story Points. The 50thAnnual Conference on statistics, computer sciences, and operation research, 27-30 Dec 2015. Retrieved from https://www.researchgate.net/publication/288839279

Raslan, A. T., Darwish, N. R., & Hefny, H. A. (2015). Towards a fuzzy based framework for effort estimation in agile software development. International Journal of Computer Science and Information Security,13(1),37.Retrieved from http://search.proquest.com/openview/0ffd04c753ca702dbd632daf9f7ffb81/1?pqorigsite=gscholar&cbl=616671

Ragin, C. C (2000). Fuzzy-Set Social Science. London: The University of Chicago Press, Ltd

Rago, R. (2020). VersionOne 2016 State of Agile Report. Agile Velocity. https://agilevelocity.com/versionone-2016-state-agile-report-agile-hits-30s/ [Accessed on 11 April 2021]

Reddy, C. S., & Raju, K. V. S. V. N. (2009). A Concise Neural Network Model for Estimating Software Effort. International Journal of Recent Trends in Engineering, 1(1), 188–193.

Ren, A., and Yun, C.(2013). Research of Software Size Estimation Method. International Conference on Cloud and Service Computing. DOI 10.1109/CSC.2013.32

Reifer, D. J. (2000). Web Development: Estimating Quick-to Market Software. IEEE Software, Vol. 17, No. 6, 2000, pp. 57-64.

Reifer, D. J. (1993). SoftCost-OO Reference Manual. Reifer Consultants, Inc., Torrance, Calif.

Ricardo, D.A., Araujo, Adriano, L.I, Oliveira, Meira, S.(2017). A class of hybrid multilayer perceptions for software development effort estimation problems. Expert Systems with Applications, Volume 90, Pages 1-12. doi.org/10.1016/j.eswa.2017.07.050.

Richardson, I., Casey, V., Mccaffery, F., Burton, J., and Beecham, S. (2012). A Process Framework for Global Software Engineering Teams. Information and Soft- ware Technology, 54(11):1175–1191.

Recruit IT. http://www.recruitit.co.nz/auckland-it-salaries-update-confirmation-of-download/ [Accessed on 15 Jun 2015]

Ribu K., Estimating object-oriented software projects with use cases; 2008.

Roark, K.R. (2020). Strategies in Software Development Effort Estimation. PhD Thesis. Walden university. https://scholarworks.waldenu.edu/dissertations

Rosemann, M., & Vessey, I. (2008). Toward improving the relevance of information systems research to practice: the role of applicability checks. MIS Quarterly, 1-22.

Robey, D. 1996. "Diversity in Information Systems Research: Threat, Promise, and Responsibility," Information Systems Research (7:4), pp. 400-408

Robinson, H., and Sharp, H. (2004). The Characteristics of XP Teams.

Robinson, H., and Sharp, H. (2005a). Organisational culture and XP: three case studies Symposium conducted at the meeting of the Agile Conference, 2005. Proceedings Retrieved from 10.1109/ADC.2005.36

Robinson, H., and Sharp, H. (2010). Collaboration, Communication and Co-ordination in Agile Software Development Practice. In I. Mistrík, J. Grundy, A. Hoek, & J. Whitehead (Eds.), Collaborative Software Engineering (pp. 93-108): Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-10294-3_5. doi:10.1007/978-3-642-10294-3_5

Royce, W.W. (1987), "Managing the development of large software systems: concepts and techniques", Proceedings of Wescon (also available in Proceedings of ICSE9, IEEE Computer Society Press, 1987), August.

Runeson, P. and Höst, M. Guidelines for conducting and reporting case study research in software engineering (2009) 14: 131. https://doi.org/10.1007/s10664-008-9102-8

Sabrjoo, S., Khalili, M., & Nazari, M. (2015). Comparison of the accuracy of effort estimation methods. 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI). Doi: 978-1-4673-6506-2/15.

Sadollah, A. (2018). Introductory Chapter: Which Membership Function is Appropriate in Fuzzy System?, Fuzzy Logic Based in Optimization Methods and Control Systems and Its Applications, Ali Sadollah,IntechOpen, DOI: 10.5772/intechopen.79552. Available from: https://www.intechopen.com/books/fuzzy-logic-based-in-optimization-methods-and-control-systems-and-its-applications/introductory-chapter-which-membership-function-is-appropriate-in-fuzzy-system-

Salas, E., Stagl, K.C., Burke, C.S., and Goodwin, G.F. 2007. "Fostering Team Effectiveness in Organizations: Towards and Integrative Theoretical Framework," in Modeling Complex Systems: Volume 52 of the Nebraska Symposium on Motivation, B.

Sehra, S. K., Brar, Y. S., Kaur, N., & Sehra, S. S. (2017). Research patterns and trends in software effort estimation. Information and Software Technology, 0, 1–21. https://doi.org/10.1016/j.infsof.2017.06.002

Sehra, S. K., Brar, B. Y., & Kaur, N. (2016). Predominant factors influencing software effort estimation. International Journal of Computer Science and Information Security, 14(7), 107–110. Retrieved from https://sites.google.com/site/ijcsis/

Salas, E., Sims, D. E., & Burke, C. S. (2005). Is there a "Big Five" in Teamwork? Small Group Research, 36(5), 555–599. https://doi.org/10.1177/1046496405277134

Salmanoglu, M., Demirors, O., and Demirord, O. (2017). Effort Estimation for Agile Software Development: Comparative Case Studies Using COSMIC Functional Size Measurement and Story Points. IWSM/Mensura'17. Gothenburg, Sweden.

Sauser, B.J., Reilly, R.R. and Shenhar, A.J. (2009), "Why projects fail? How contingency theory can provide new insights – a comparative analysis of NASA's mars climate orbiter loss", International Journal of Project Management, Vol. 27 No. 7, pp. 665-679.

Samson, B., Ellison, D., & Dugard, P. (1997). Software cost estimation using an Albus perceptron (CMAC). Information and Software Technology, 39(1), 55–60. doi:10.1016/0950-5849(96)01124-X

Saljoughinejad, R., and Khatibi, V. (2018). A New Optimized Hybrid Model Based on COCOMO to Increase the Accuracy of Software Cost Estimation. Journal of Advances in Computer Engineering and Technology, 4(1).

Schein, E.H. (2009). The Corporate Culture Survival Guide, New & Revised Edition. Jossey-Bass.

Scrum, I. (2019, November 19). The Scrum Product Backlog - International Scrum Institute. Retrieved July 07, 2020, from https://www.scruminstitute.org / The_Scrum_Product_Backlog.php

Schwalbe, K. (2014).Information Technology Project Management. Revised 7th edition. Cengaga Learning, USA.

Schwaber, K., & Sutherland, J. (2016). The Scrum Guide TM. https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-US.pdf [Accessed on 14-09-2017]

Schwaber, K. (1995). Scrum Development Process. OOPSLA'95 Workshop on Business Object Design and Implementation, Springer-Verlag.

Schwaber, K. and M. Beedle (2002). Agile Software Development With Scrum. Upper Saddle River, NJ, Prentice-Hall.

Schultze, Ulrike & Avital, Michel (2011). Designing interviews to generate rich data for information systems research. *Information and Organization*, *21*(1), 1-16.

Sedelmaier, Y., and Landes, D. (2014). Software Engineering Body of Skills (SWEBOS). IEEE Global Engineering Education Conference (EDUCON), 395-400. doi: 978-1-4799-3190-3/14

Schmietendorf, A., Kunz, M., and Dumke, R. (2008). Effort estimation for Agile Software Development Projects. 5th Software Measurement European Forum.

Shepperd, M. J., & MacDonell, S. G. (2012). Evaluating prediction systems in software project estimation. Information and Software Technology, 54(8), 820–827. doi:10.1016/j.infsof.2011.12.008

Shepperd, M., and Schofield, M. (1997). Estimating Software Project Effort using Analogies. IEEE Trans. On

Software Engineering, vol. 23, no. 12.

Shepperd, M. J., Schofield, C. and Kitchenham, B. (1996). Effort Estimation Using Analogy. Proc. 18th Int'l Conf. Software Eng.

Shepperd, M., Cartwright, M., and Kadoda, G. (2000). On building prediction systems for software engineers. Empirical Software Engineering. 5 (3). 723 175–182.

Shepperd, M., & MacDonell, S.G. (2012) Evaluating prediction systems in software project estimation, Information and Software Technology 54(8), pp.820-827. doi: 10.1016/j.infsof.2011.12.008
Sharna, A., and Gupta, A. (2012). Impact of organizational climate and demographics on project specific risks: in context to Indian Software industry. International Journal of Project Management 30 (2012) 176–187. doi: 10.1016/j.ijproman.2011.05.003

Shekhar, S., & Kumar, U. (2016).Review of various Software Cost Estimation Techniques. International Journal of Computer Applications (0975 – 8887), volume 141(31-34).

Shekhar, S., & Kumar, U. (2016). Review of various software cost estimation techniques. International Journal of Computer Applications, 141(11), 31–34. Retrieved from http://www.ijcaonline.org/

Shams, A., Bőhm, S., Winzer, P., and Dőrner, R. (2019). App Cost Estimation: Evaluating Agile Environments. IEEE 21st Conference on Business Informatics (CBI). DOI 10.1109/CBI.2019.00050.

Shenhar, A. and Dvir, D. (2007), Reinventing Project Management: The Diamond Approach to Successful Growth and Innovation, Harvard Business School Press, Boston, MA.

Shuart, W. Spaulding and J. Poland (Eds.). Nebraska: University of Nebraska Press, pp. 185-244
"Team Spirit." Definition of Team Spirit - Feeling (British and World English). Oxford Dictionaries, https://en.oxforddictionaries.com/definition/team_spirit [Accessed on 16-August-2017]

Sliger, M. (2012). Agile estimation techniques. Paper presented at PMI® Global Congress 2012—North America, Vancouver, British Columbia, Canada. Newtown Square, PA: Project Management Institute.

Sommerville, I. (2007). Software engineering. 8th: Harlow, England.

Song, Q., and Shepperd, M. (2011). Predicting software project effort: A grey relational analysis based method doi:10.1016/j.eswa.2010.12.005

Souza, L.S.D., and Aquino, G.S.D. (2014). Estimating the effort of mobile application development. Proceedings on 2nd International Conference on Computational Science and Information Technology. CS & IT-CSCP, pp. 45–63. DOI: 10.5121/csit.2014.4405.

Singh, S.P., & Kumar, A. (2017). Software Cost Estimation Using Homeostasis Mutation Based Differential Evolution. IEEE, 11th International Conference on Intelligent Systems and Control (ISCO).

Simon, H.A. 1996. The Sciences of the Artificial. MIT Press.

Smaczny, T. (2001). Is an alignment between business and information technology the appropriate paradigm to manage IT in today's organisations? Management Decision, 39(10), 797-802.
Sidorova, A., Evangelopoulos, N., Valacich, J. S., and Ramakrish- nan, T. 2008. "Uncovering the Intellectual Core of the Informa- tion Systems Discipline," MIS Quarterly (32:3), pp.46

Stapleton, J. (1997). Dynamic systems development method - The method in practice, Addison Wesley.

Statstestcalc - check assumptions, interpret results. (n.d.). Statskingdom.Com. Retrieved March 14, 2021, from https://www.statskingdom.com/index.html

Stutzke, R.D.: Software Project Estimation: an overview. In: Software Management. Wiley, Hoboken (2006).

Strode, D. (2015). Applying Adapted Big Five Teamwork Theory to Agile Software Development. Australasian Conference on Information Systems

Strode, D., Huff, S., Tretiakov, A. (2009). The impact of organizational culture on agile method use. Proc. of HICSS'09, Waikoloa, Big Island,Hawaii,

Stavru, S. 2014. "A Critical Examination of Recent Industrial Surveys on Agile Method Usage," The Journal of Systems and Software (94), pp. 87-97

Stober, T., and Hansmann, U. (2010). Overview of Agile Software Development, Agile Software Development: Best Practices for Large Software Development Projects. Springer.

Stefan, Burcea (2007) THE USE OF PROGRAM EVALUATION AND REVIEW TECHNIQUES (PERT) IN THE MANAGEMENT OF HEALTH ORGANIZATIONS. Academy of Economic Studies Faculty of Management Bucharest Romania.

Stutzke, R.D. Software Estimating Technology: A Survey. Science Applications International Corp.

Suharjito, Nanda, S., and Soewito, B. (2016). Modeling Software Effort Estimation Using Hybrid PSO-ANFIS: International Seminar on Intelligent Technology & its Application. IEEE.

Suri, P.K., and Ranjan, P. (2012). Comparative Analysis of Software Effort Estimation Techniques. International Journal of Computer Applications (0975–8887), 48(21).

Sutherland, J. & Schwaber, K. (2016). The Scrum Guide. The Definitive Guide to Scrum: The Rules of the Game. scrum.org.

Sturges, Judith E. & Hanrahan, Kathleen J. (2004). Comparing telephone and face-to-face qualitative interviewing: A research note. *Qualitative Research*, *4*(1), 107-118.

Tanveer, B., Guzmán, L., and Engel, U.M. (2017). Effort estimation in agile software development: Case study and improvement framework. Journal of software: Evolution and Process. DOI: 10.1002/smr.1862

Tanveer, B., Guzmán, L., and Engel, U. M. (2016). Understanding and improving effort estimation in Agile software development- an industrial case study. Proc. Int. Work. Softw. Syst. Process - ICSSP '16, pp. 41–50.

Tadayon, N. (2005). Neural Network Approach for Software Cost Estimation. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05).

Tausworthe, R. C. (1980). The work breakdown structure in software project management. The journal of systems and software, 1, 181-186.

Tariq, M., & Shobha, B.C. (2011) Quasi-Experiment Evaluating deviations in effort estimations based on Use Case Point (UCP) and UCP extended with Program Evaluation Review Technique (PERT). Master Dissertation for Software Engineering & Management. Goteborg University, Sweden.

Teddlie, C., & Tashakkori, A. (2009). Foundations of mixed methods research: Integrating quantitative and qualitative approaches in the social and behavioral sciences: Sage Publications Inc.

Teplitz, C.J. (1991). The Learning Curve Deskbook: A Reference Guide to Theory, Calculations and Applications. Quorum Books, New York.

Thamhain, H., 2013. Managing risks in complex projects. PMI. J. 44, 20–35. http://dx.doi.org/10.1002/pmj.21325.

Tolfo, C., Wazlawick, R. 2008. The influence of organizational culture on the adoption of extreme programming. Journal of Systems and Software, IST 11(81): 1955-1967.

Tolvanen, J. P. (1998). Incremental method engineering with modeling tools. PhD dissertation. Jyväskylä University Printing House and ER-Paino Ky, University of Jyväskylä, Finland.

Torrecilla-Salinas, C. J., Sede ˜ no, J., Escalona, M. J., and Mej´ıas, M. (2015). Estimating, planning and managing Agile Web development projects under a value-based perspective. Information and Software Technology. 61,124 – 144. DOI: http://dx.doi.org/10.1016/j.infsof.2015.01.006

Tripp, J.F., Riemenschneider, C., & Thatcher, J.B. (2016). Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign. Journal pf the Association for Information Systems, Volume 17, issue 4, pp. 267-307.

Tronto, I. F. B., Silva, J. D. S., & Sant'Anna, N. (2008). An investigation of artificial neural networks based prediction systems in software project management. Journal of Systems and Software, 81(3), 356–367. doi:10.1016/j.jss.2007.05.011

Ungan, E., Cizmeli, N., Demirörs, O. (2014). Comparison of Functional Size Based Estimation and Story Points, Based on Effort Estimation Effectiveness in SCRUM Projects. 40th Euro micro Conference on Software Engineering and Advanced Applications

Usman, M., Mendes, E., Weidt, F. and Britto,R. (2014). Effort estimation in agile software development: A systematic literature review, in Proc. 10th Int. Conf. Predictive Models in Software Engineering, 2014, pp. 82–91.

Usman, M., Emilia, M., Francila, W., and Ricardo, B. (2014). Effort Estimation in Agile Software Development: A Systematic Literature Review, PROMISE '14. Torino, Italy.

Usman, M., Mendes, E., and Börstler, J. (2015). Effort estimation in agile software development: A survey on the state of the practice, in Proc. 19th Int. Conf. Evaluation and Assessment in Software Engineering, 2015, pp. 12:1–12:10.

Usman, M., and Britto, R. (2016). Effort Estimation in Co-located & Globally Distributed Agile Software Development: A Comparative Study. Joint Conference of the International Workshop on Software Measurement & the International Conference on Software Process & Product Measurement, 2016. DOI 10.1109/IWSM-Mensura.2016.14

Usman, M., Börstler, J. and Petersen, K. (2017). An Effort Estimation Taxonomy for Agile Software Development. International Journal of Software Engineering and Knowledge Engineering. Vol. 27, No. 4 (2017) 641–674
DOI: 10.1142/S0218194017500243

Usman, M., Britto, R., Börstler, J., Damm, L., and Börstler, J. (2018). Effort estimation in large-scale software development: An industrial case study: An industrial case study. Information and Software Technology. Volume 99, Pages 21-40. doi.org/10.1016/j.infsof.2018.02.009.

Vachik S. Dave and Kamlesh Dutta, Neural network based models for 181 software effort estimation: a review, Artif Intell Rev (2014) 42, pp. 295-307, 2012.

Vaishnavi, V., Kuechler, W., and Petter, S. (Eds.) (2004/19). "Design Science Research in Information Systems" January 20, 2004 (created in 2004 and updated until 2015 by Vaishnavi, V. and Kuechler, W.); last updated (by Vaishnavi, V. and Petter, S.), June 30, 2019. URL: http://www.desrist.org/design-research-in-information-systems/.

Vaishnavi, V. and Kuechler, W. (2004/5). Design Research in Information Systems.

Vaishnavi, V. K., & Kuechler, W. (2015). Design science research methods and patterns: innovating information and communication technology: Crc Press.

Venable, J. (2006). The role of theory and theorising in design science researchCiteseer. Symposium conducted at the meeting of the Proceedings of the 1st International Conference on Design Science in Information Systems and Technology (DESRIST 2006)

Vetter, T. R. (2017). Descriptive Statistics: Reporting the Answers to the 5 Basic Questions of Who, What, Why, When, Where, and a Sixth, So What? International Anesthesia Research Society. Vol 125.

Vijayasarathy, Leo R. Agile Software Development: A survey of early adopters. Journal of Information Technology Management Volume XIX, Number 2. 2008.

Viswanath Venkatesh, S. A. B., and Hillol Bala. (2013). Bridging the Qualitative-Quantitative Divide: Guidelines for Conducting Mixed Methods Research in Information Systems. 37(1), 21-54.

Vits, J., Gelders, L. (2002). Performance improvement theory. International Journal of Production Economics 77 (3), 285-298.

Vogl, Susanne (2013). Telephone versus face-to-face interviews: Mode effect on semistructured interviews with children. *Sociological Methodology*, *43*(1), 133-177.

Vrhovec, S.L.R., Hovelia, T., Vavpotic, D., & Krisper, M. (2015). Diagnosing organizational risks in software projects: Stakeholder resistance. International Journal of Project Management 33 (2015) 1262–1273. http://dx.doi.org/10.1016/j.ijproman.2015.03.007

Walls, J., Widmeyer, G.R. and El Sawy, O.A. (1992). Building an Information System Design Theory for Vigilant EIS, Information Systems Research, Vol. 3, No. 1, pp. 36-59.

Wanderley, E.G., Vasconcelos, A., & Avila, B.T. (2018). Using Function Points in Agile Projects: A Comparative Analysis Between Existing Approaches. WBMA 2017, CCIS 802, pp. 47-59. https://doi.org/10.1007/978-3-319-73673-0_4

Wegner.S, and Ruhe.M. A Systematic Review of Productivity Factors in Software Development. in Proc. 2nd International Workshop on Software Productivity Analysis and Cost Estimation. (SPACE 2008), Beijing, China, 2008

Weber, C., Fayed, A. (2010). Optimizing your position on the operating curve: how can a fab truly maximize its performance? IEEE Transactions on Semiconductor Manufacturing 23 (1), 21-29.

Wen, J., Li, S., Lin, Z., Hu, Y., & Huang, C. (2012). Systematic literature review of machine learning based software development effort estimation models, Inf. Softw. Technol. 54 (1) 41–59, doi: 10.1016/j.infsof.2011.09.002.

William, T. (2005). Assessing & moving on from the dominant project management discourse in the light of project overruns. Engineering Management: IEEE Journal, 52(4), 497 -508.

Williams, E. (2017, November 21). How Do Co-Workers Affect Employee Behavior? Work - Chron.Com. Retrieved February 28, 2021, from https://work.chron.com/coworkers-affect-employee-behavior-4245.html

Wikipedia. Retrieved on 11 June 2017 https://en.wikipedia.org/wiki/Program_evaluation_and_review_technique

Wohlin, C., Runeson, P., Host, M., Regnell, B., & Wesslen, A. (2000). Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Boston/Dordrecht/London.

Wright, T. (1936). Factors affecting the cost of airplanes, Journal of Aeronautical Science 3 122–128.

Wright, G., & Bolderm, F. (1992). Expertise and decision support. Plenum Press.

Wroblewski, M. T. (2020, October 12). *Top 10 Characteristics of a Great Workplace*. Small Business – Chron.Com. Retrieved February 28, 2021, from https://smallbusiness.chron.com/top-10-characteristics-great-workplace-11226.html

Wu, L. (1997) The Comparison of the Software Cost Estimating Methods. University of Calgary.

Yinhuan, Z., Beizhan, W., Yilong, Z., & Liang, S. (2009). Estimation of software projects effort based on function point. in Computer Science & Education. ICCSE '09. 4th International Conference.

Yilmaz, M., O'Connor, R. V., and Clarke, P. (2016). Effective Social Productivity Measurements during Software Development - An Empirical Study. International Journal of Software Engineering and Knowledge Engineering, 26(3):457–490.

Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 6, 338–353.

Zahraoui, H., Idrissi, M.A.J. (2015). Adjusting Story Points Calculation in Scrum Effort & Time Estimation.

Zhao, Yuan and Tan, Hee Beng Kuan and Zhang, Wei (2003) Software Cost Estimation through Conceptual Requirement. Proceedings of the Third International Conference on Quality Software, (1) pp. 1-4.

Ziauddin, Tipu, S.K., & Zia, S. An Effort Estimation Model for Agile Software Development. Advances in Computer Science and its Applications (ACSA) Vol. 2, No. 1, 2012, ISSN 2166-2924.

Appendices

Appendix-A (COEEMO Prototype's Screenshots)

A prototype of the model was developed by using VBA in Microsoft Excel. Soft copy of this application was provided to the team who applied COEEMO during the experimentation part of this study. This application helped the team in collecting data regarding their team and sprints of their projects. For ease of use, one short-duration video was also provided to the team.

In the COEEMO prototype one individual workbook of an excel sheet is dedicated to a single significant factor to perform various tasks like data entry/selection of categories / subcategories along with relevant scaling / ratings of the factors from the given options. Figure A.1 shows an active workbook of the COEEMO prototype through which the user may choose different categories/subcategories of team experience and expertise along with their respective ratings. For ease of use, different combo boxes are used in the prototype so that the team may easily choose the required category/subcategory of the experience/expertise of the individual team members and then the application will compute the overall team experience and expertise.

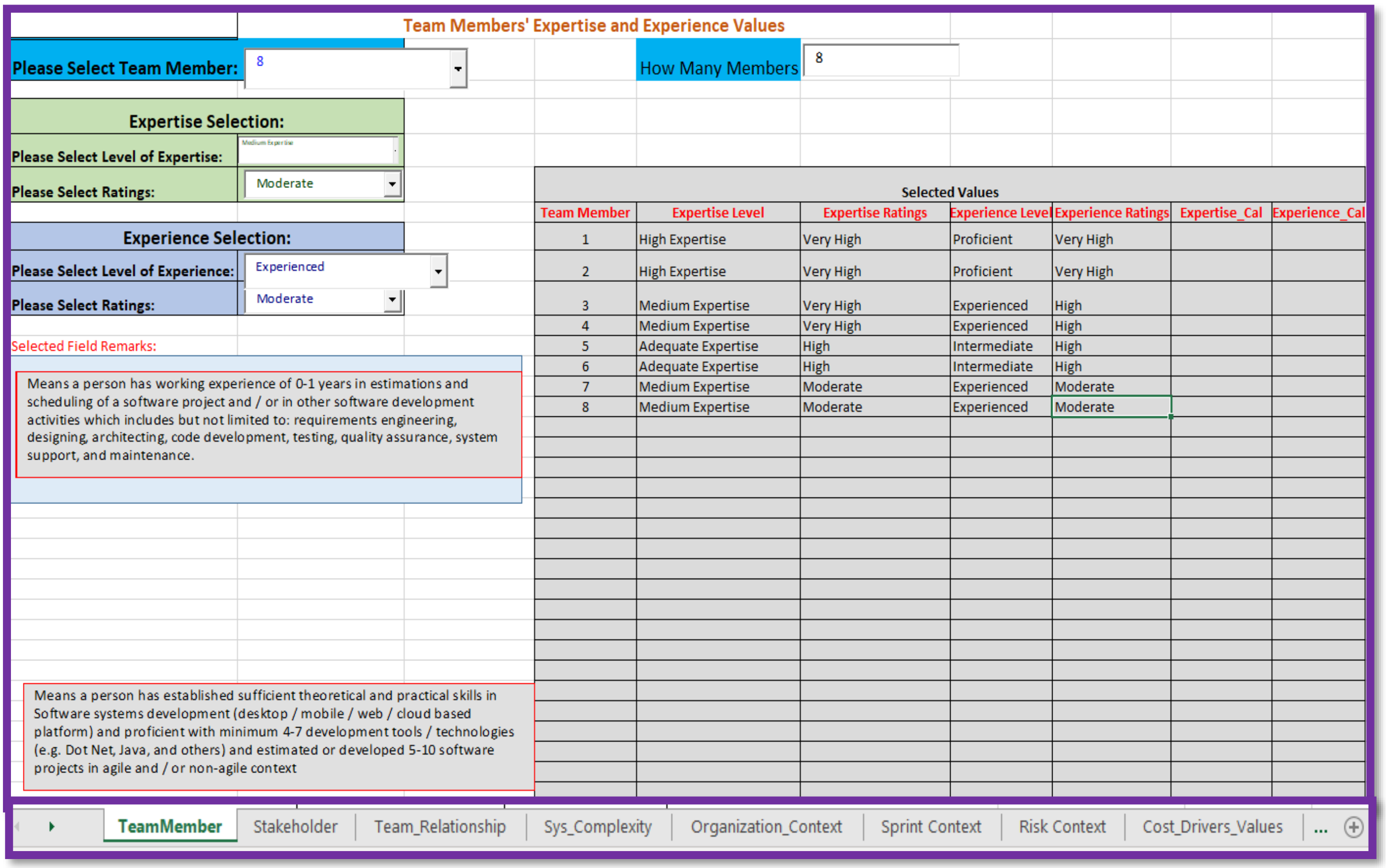


Figure A.1. COEEMO's Prototype Interface for levels' entries of Team Members' expertise and experience

Figure A.2 shows the interface of the COEEMO prototype where the team may select the number of stakeholders of their project along with their level of cooperation. The application will calculate the overall cooperation level of the stakeholders.

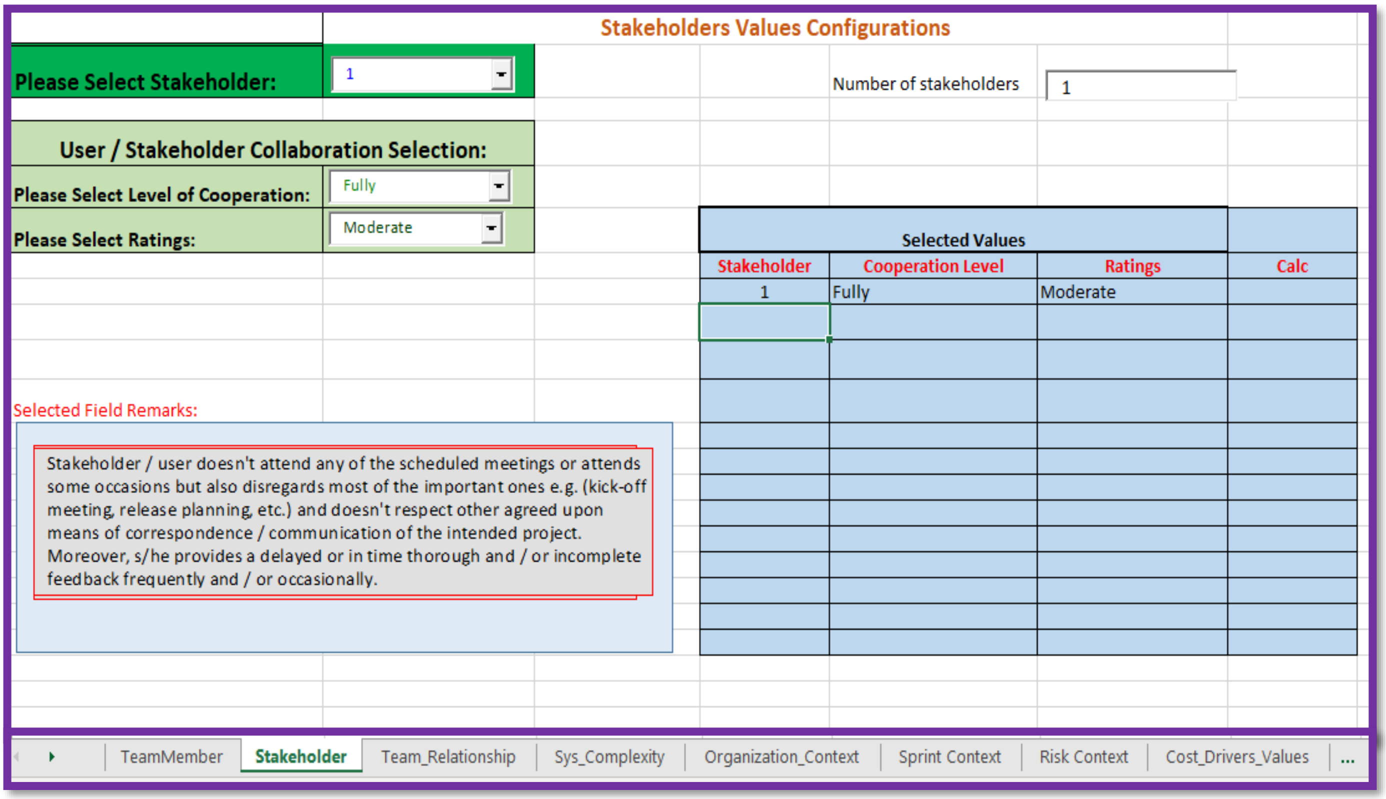


Figure A.2. COEEMO's Prototype Interface for levels' entries of Project Stakeholder cooperation

The team may choose their level of team effectiveness and relevant ratings based on the provided parameters from the interface of COEEMO, as shown in the figure A.3.

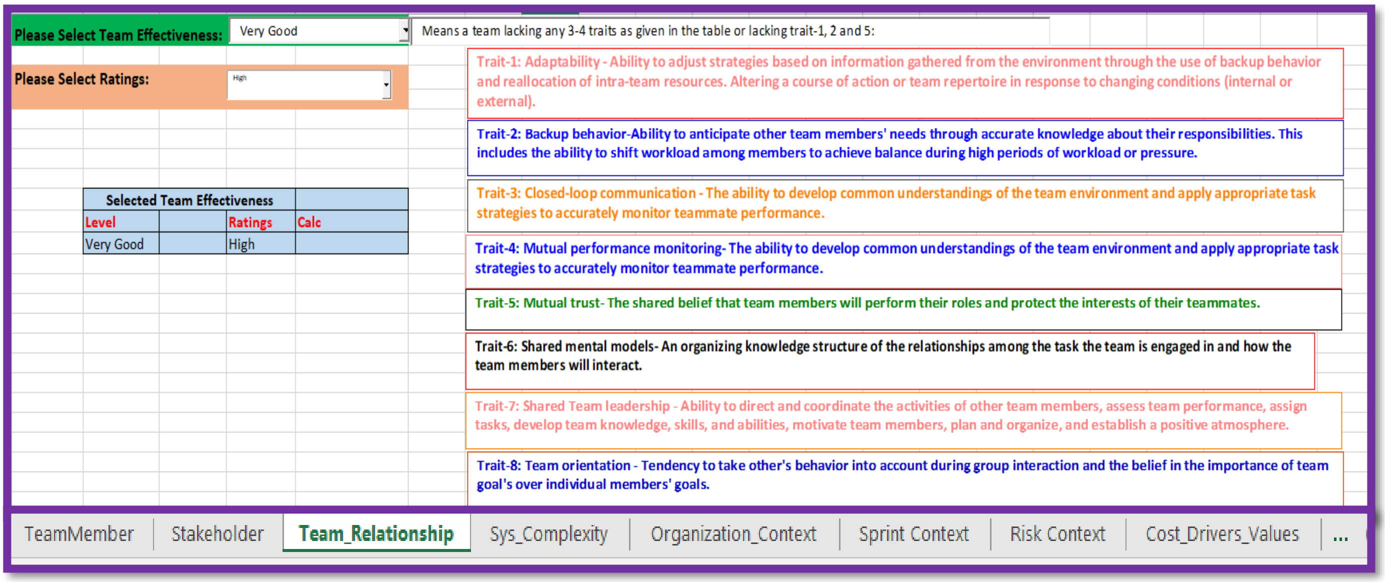


Figure A.3. COEEMO’s Prototype Interface for value entry of Team effectiveness

The team may also select the type of the system that they want to implement and then the expected level of complexity for their envisioned system with respect to its implementation. Further, entry of the total expected sprints for the whole intended project/product can be given through the same interface as shown in the figure A.4.

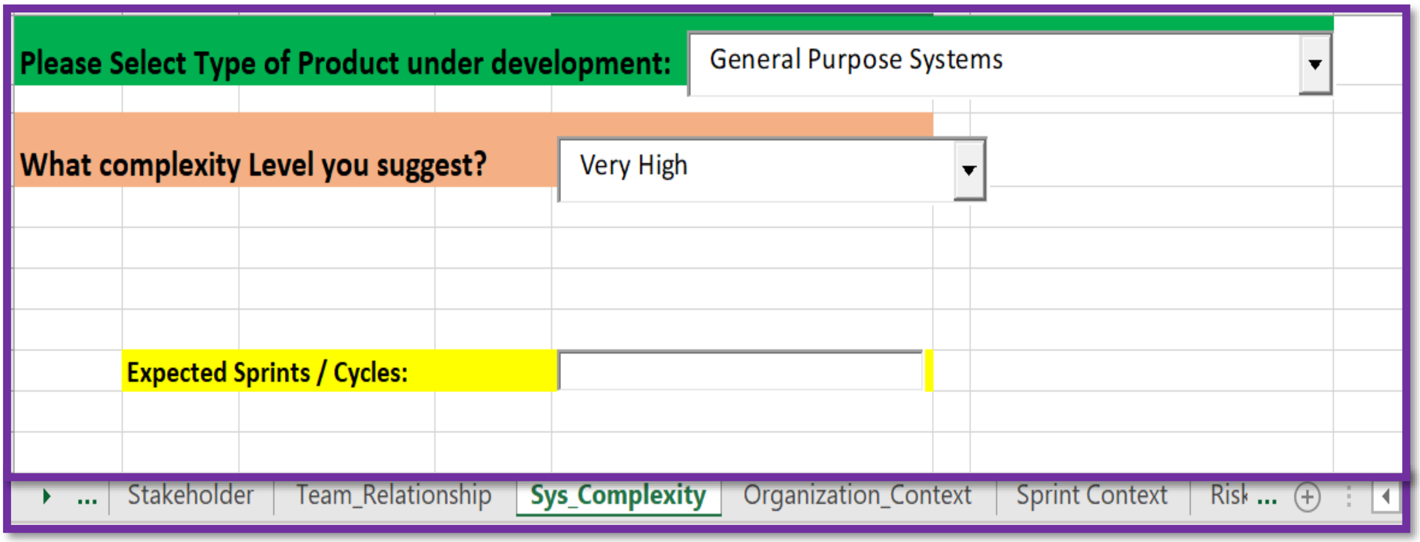


Figure A.4. COEEMO’s Prototype Interface for level entry of System’s complexity

For the entries of the type of the organization to which the development team belongs, the culture which they are bound to follow along with their relevant scaling/ratings, and the working environment where the team will sit together to work for their envisioned project, the screen as given in the figure A.5 will be used.

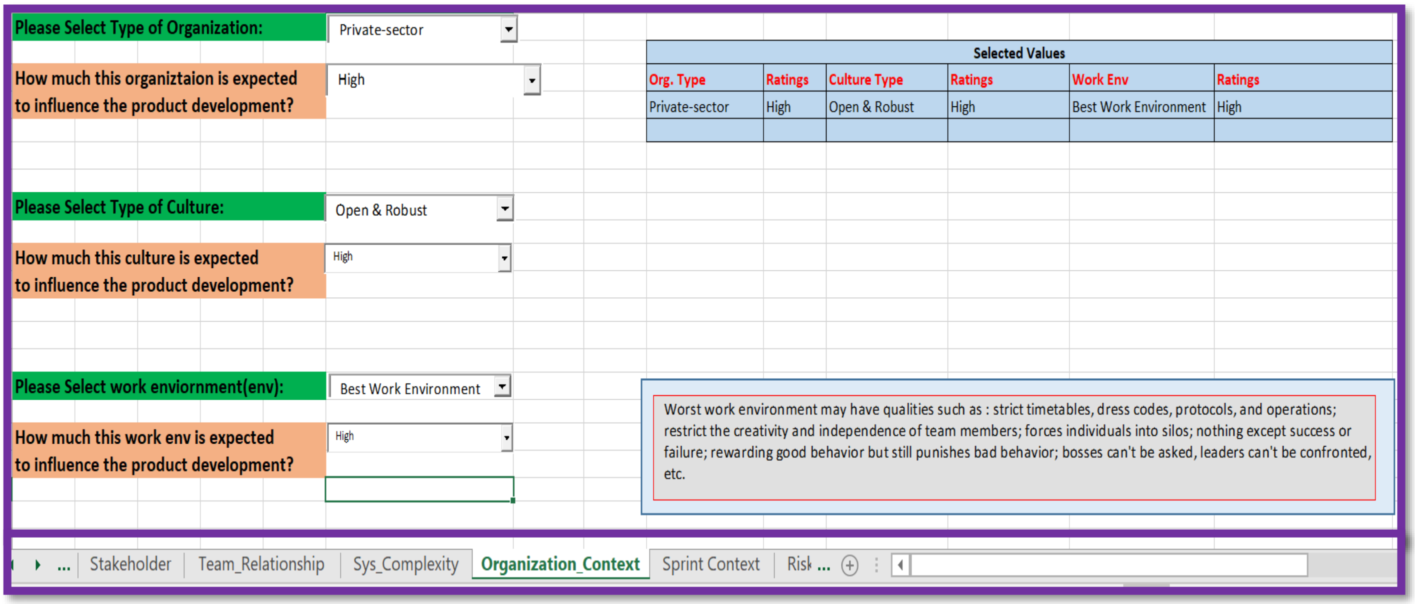


Figure A.5. COEEMO's Prototype Interface for impacts entries of Org. type, culture & work environment

The number of sprints and their expected complexity can be entered through the interface screen of the COEEMO prototype as shown in figure A.6.

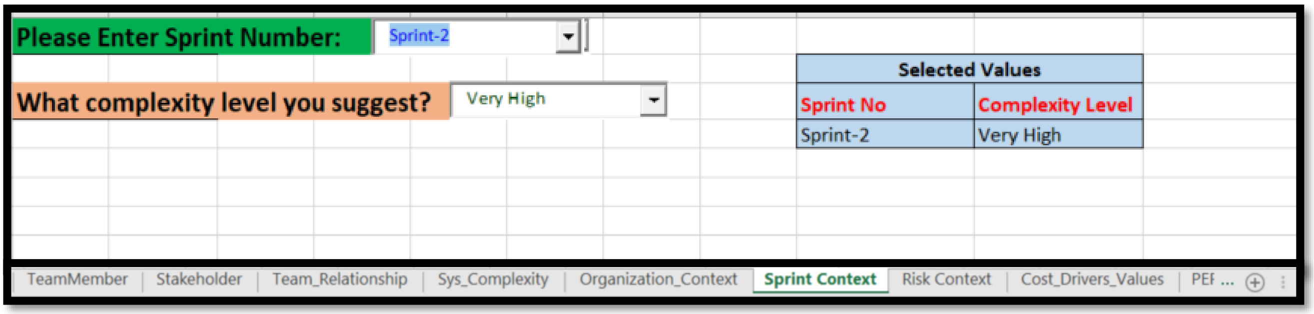


Figure A.6. COEEMO's Prototype Interface for values entries for the Spt No. and their expected complexity

To select the project's risks and their expected influences on the intended project it is possible to enter through the interface screen of COEEMO as exhibited in the figure A.7.

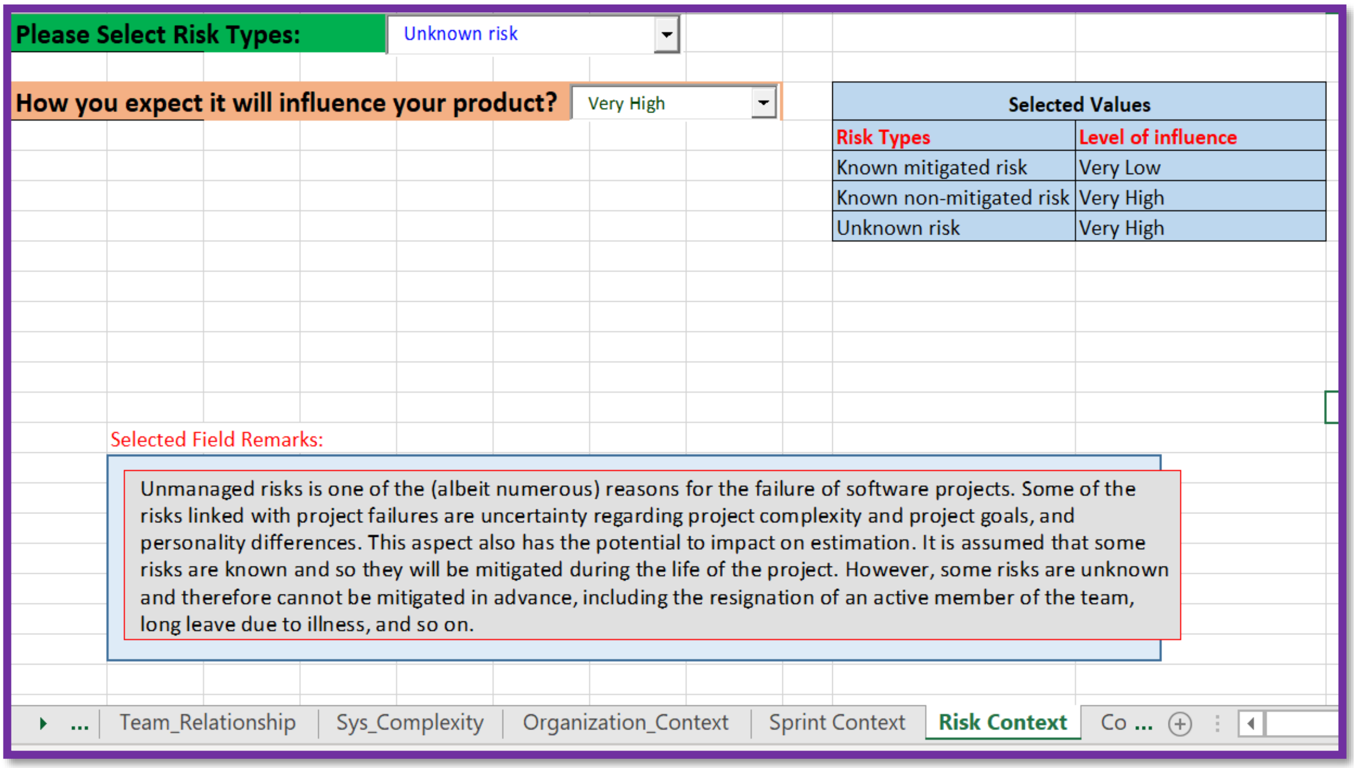


Figure A.7. COEEMO's Prototype Interface for values entries for type of Risks and its expected complexity

After the essential entries of all the factors through the various interface screens of the COEEMO prototype, as shown above from figure A.1 to A.7, the final calculations are possible through the interface screen as shown in figure A.8. This is the final output screen of the COEEMO prototype which will display the technical and environmental impacts of the envisioned project as suggested by the model.

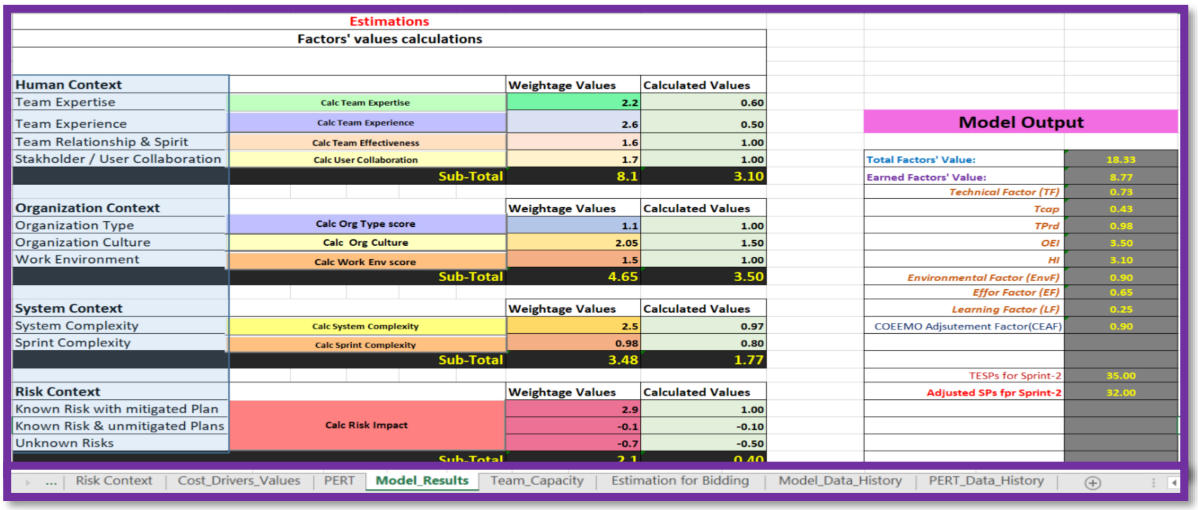


Figure A.8. COEEMO's Prototype Interface to show TF, EF, EAF and planned SPs for the Planned Sprint

This screen as shown in figure A.9 will also be utilized for the data entry of the team's velocity and the team's estimated story points for the planned sprint. After these important entries, this screen will also present the effort adjustment factor (EAF) and then finally the estimated story points as indicated by the Model. If a team is interested to follow the PERT technique in their effort estimation, then this interface will also help them to calculate the EAF based on the PERT values. Prior to using the PERT option, however, the team needs to enter the estimated story points of all their individual team members against each story of the planned sprint. The interface screen of COEEMO for the PERT entries is shown in figure A.9.

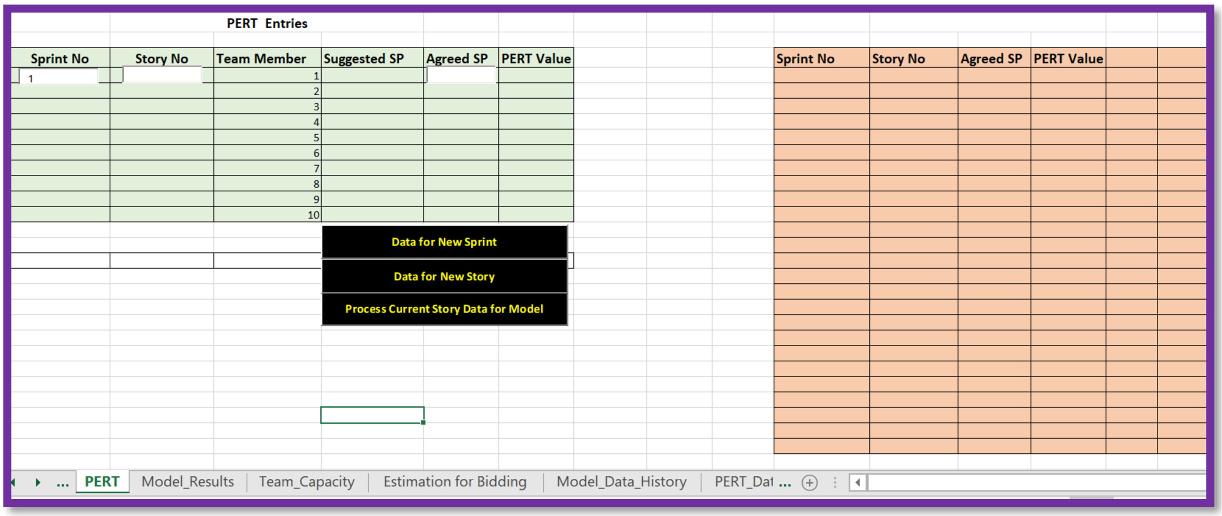


Figure A.9. COEEMO's Prototype Interface for Data entry of PERT values

Appendix-B (Actual Team Capacity Calculations' Instructions)

Activity / Items Name	Actions	Misc Instructions and Formulas
Sprint Start Date	7-Jan-19	Complete the table which is given below. Sample data is filled on the left side of this table to give you an idea. As you can see
Sprint End Date	20-Jan-19	in the sample data that if we subtract the various activity hours from the "Work Hours" in a sprint then we can easily calculate
Workdays in Sprint	10	the Net Work Hours as shown in the sample data.
Work Hours in Sprint	80	
Daily Scrum Hours	1	Formulas
Prepartion Hours (Sprint planning, etc.)	2	Hours = Availability * (NetWorkHours in a Sprint - Hoursoff)
WBH (Work / Task Break Down Hours)	2	TeamHours = Sum of Hours of all team members
Team Presentation / Demo to client	2	NetTeamResources = TeamHours / WorkHours in Sprint
Retrospective Hours	1	Points Available = TeamHours / 8*
Any other commitments in Hours	1	*These hours may vary from team to team
Net Work Hours in a Sprint	71	

Appendix-C (Treatments' Instructions to Apply)

Instructions to apply Expert Judgement by using Bottom-up Approach during Story-Sizing or Estimation:

a. **Existing Method of Estimation (EME)**: Agile teams working in SRCUM using Planning poker or T-shirt sizing or Relative Mass valuation and expert judgement based on gut feelings, analogy, Top-Down and others excluding Bottom-Up approach as their existing methods of estimation.
b. **Proposed Method of Estimation (PME)**: Agile teams working in SRCUM using Planning poker or T-shirt sizing or Relative Mass valuation need to apply expert judgement based strictly on Bottom-Up approach by using Work Break Down Structure (WBS) of a story into tasks/activities to anticipate the size of the story while considering all these tasks/activities in mind.

   Example: Team is sitting in their sprint's planning meeting to decide the number of stories along with their sizes in points or T-shirt sizing or Relative Mass Valuation to plan the work for their upcoming sprint. In the proposed method of estimation (PME), each member of the team needs to consider work breakdown of the candidate story (for which size is sought) into activities / tasks (e.g., analysis, design, coding, testing, data base work, Integration, and others) required to implement the candidate story prior to finalize/share its size.

Note: You can choose anyone option as stated below to apply above stated EME and PME to plan all sprints of your ongoing/new project in agile development.

Option-1:

1. Agile team is suggested to be divided into two subsets of small teams e.g., if there are 6 members in a team, they may be divided in two subsets (subset-A and B) of equal numbers. In case of 05 members, subset-A or B may have 03 or 02 members each.
2. One subset of team may apply EME and other PME to estimate sizes of stories simultaneously during the same sprint.
3. Both techniques are applied precisely as per above given assumptions (Under EME and PME).
4. Discussion / consensus regarding the estimation / sizing of a story should be made within own subset of the team (individual as well as team feedback regarding the sizes for all stories should be recorded within own subsets of the team).
5. If the team is using Planning Poker method, then the final size of the story should be the average of the two values as decided by the two subsets of the team. While for T-shirt sizing or Relative Mass Valuation, the middle category of the size in between the concluded sizes of both subsets should be selected as the final size of the story (e.g., in T-shirt sizing one subset says "Small" and other says "Large", then "Medium" could be selected as final size). Moreover, if no size exists in between the two estimated categories then the highest size should be considered as

the final size of the story (e.g., in the above example of T-shirt sizing if one subset says “Small” and other says “Medium”, then “Medium” could be selected as final size)

6. Unfinished story-points should be recorded for each of the sprint.
7. Feedback / salient points of retrospective meeting for each sprint should be recorded especially with respect to any improvements in communication, tool, technology, team relationship, stakeholder coordination, etc.

Option-2:

1. Agile team is suggested to apply EME and other PME approaches in alternate sprints of the project (e.g., if team apply EME in sprint-1 then PME should be followed in sprint-2, and so on).
2. Both techniques are applied precisely as per above given assumptions (Under EME and PME).
3. Discussion / consensus regarding the estimation / sizing of a story should be made (individual as well as team feedback regarding the sizes for all stories should be recorded for each sprint).
4. Unfinished story points should be recorded for each of the sprint.
5. Feedback / salient points of retrospective meeting of each sprint should be recorded especially with respect to any improvements in communication, tool, technology, team relationship, stakeholder coordination, etc.

Note: Excel sheet is provided to record the required data.

Appendix-D (Experimentation Data Collection Sheets)

SPRINT/CYCLE DATA COLLECTION SHEET FOR EXPERIMENTATION

Date	Approch Type	Meeting Number	Sprint Number	No-of-Team Members	User Story#	Individual SPs	Agreed SPs	TSPs of Sprint	Unfinished SPs	Remarks if any

SPs: Story-Points TSP: Total Story-Points

Data Collection Sheet for Retrospective Meeting

Date	Meeting Number	Sprint #	Feedback on Team Work	Feedback Regarding Organiztaion / Culture	Feedback Regarding User/Stakeholder	Feedback on other specific area
			Here feedback can be given with respect to team exeprince, expertise, communication, collaboration, complexity of Tasks, Risk, etc.	If delay is due to organization's culture, environment, policy, or else	If anything due to stakeholder / user behaviour or response, etc.	If anything not covered yet

Data Collection Sheet for Sprint Review Meeting

Date	Meeting Number	Sprint No	Accomplishments	Pendings / Remarks

Appendix-E (Data Collection Sheets for on-going/completed ASD Projects)

Data Collection Main Sheet for Completed Projects

	Project Name/ID	Total Sprints	TMs	No of Stakeholders	Project status
Sample Data	xyz	6	8	2	Completed

This is sample data. In Project status you may write completed / In-progress, etc

Note: After completeing all Projects main information here then plz select the given sheets to enter further details of the projects.

TMs: Team Members

Data Collection Sheet for Sprint/Cycle

	Project Name/ID	Sprint No	Estimated SPs	Finished / completed SPs
Sample Data	xyz	1	35	30
		2	45	38
		3	42	39
		4	40	40
		5	38	35
		6	42	37

Note: If you don’t estimate your work in story points then you can give here your own measuring units but please note that you have to give here delays in days / hours for sprints' delivery

Data Collection Sheet for TMs Expertise & Experience

	Project Name/ID	Project / Sprint level	Team Member	Expertise Level	Expertise Ratings	Experience Level	Experience ratings
Sample Data	xyz	Project level	1	High expertise	Very High	Proficient	Moderate
	xyz	Project level	2	Adequate expertise	High	Experienced	High
	xyz	Project level	3	Inadequate expertise	very High	Intermediate	Moderate
	xyz	Project level	4	Medium expertise	High	Proficient	Very High
	xyz	Project level	5	Medium expertise	Moderate	Experienced	High

I have entered data for team of five members as a sample

Ratings/Gradings:

Very High
High
Moderate
Low
Very low

Experience Level	
Proficient:	Means a person has working experience of 10 years or more in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.
Experienced:	Means a person has working experience of 6-9 years in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.
Intermediate:	Means a person has working experience of 2-5 years in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.
Beginner:	Means a person has working experience of 0-1 years in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.

Expertise Level	
High expertise:	Means a person has established sufficient theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform) and proficient with minimum 8-10 development tools / technologies (e.g. Dot Net, Java, and others) and estimated or developed more than 10 number of software projects in agile and / or non-agile context (s).
Medium expertise:	Means a person has established sufficient theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform) and proficient with minimum 4-7 development tools / technologies (e.g. Dot Net, Java, and others) and estimated or developed 5-10 software projects in agile and /
Adequate expertise:	Means a person has established necessary theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform) and proficient with minimum 1-3 development tools / technologies (e.g. Dot Net, Java, and others) and estimated or developed 1-4 software projects in agile and /
Inadequate expertise:	Means a person has established necessary theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform)

Data Collection Sheet for Stakeholder

	Project Name/I	Project/sprint le	Stakeholder	Level of Collaboratior	Ratings
Sample Data	xyz	Project level	1	Fully	Very High
		Sprint-2 only	2	Partially	High
		Project level	3	Poorly	Moderate

Entered data of 3 stakeholders

Ratings/Gradings:
Very High
High
Moderate
Low
Very low

Level of Collaborations	
Fully:	Stakeholder / user collaborates fully with the development team by attending all scheduled meetings and respects completely other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides a thorough feedback as and when required.
Partially:	Stakeholder / user collaborates partially with the development team by attending some of the scheduled meetings without missing the important ones e.g. (kick start, release planning, etc.) and respects completely other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides a thorough feedback as and when required.
Occasionally:	Stakeholder / user collaborates occasionally with the development team by attending some of the scheduled meetings but also disregards some of the important ones e.g. (kick-off meeting, release planning, etc.) and respects at times other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides delayed or in time thorough or incomplete feedback frequently or occasionally.
Poorly:	Stakeholder / user doesn't attend any of the scheduled meetings or attends some occasions but also disregards most of the important ones e.g. (kick-off meeting, release planning, etc.) and doesn't respect other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides a delayed or in time thorough and / or incomplete feedback frequently and / or occasionally.

Data Collection Sheet for Team Relationship & Spirit

	Project Name/ID	Project / Sprint level	Team Effectiveness level	Ratings
Sample Data	xyz	Project level	Very good	High
		Sprint-3 only	Good	Very high

I have entered sample data for single team

Ratings/Gradings:
Very High
High
Moderate
Low
Very low

Team Effectiveness Level

Excellent	Means team has all traits (from 1-8) as given in the Traits table
Very Good	Means team has all given traits (from 1-8) but with little deviations in trait 6 or 7
Good	Means team has any 06 traints (from 1-8) but without any deviations in trait 1,2,4,5 and 8
Conflicting	Means team is lacking any 3-4 traits (from 1-8) or lacking specifically trait 1, 2 and 5

Traits Table

Trait-1: Adaptability - Ability to adjust strategies based on information gathered from the environment through the use of backup behavior and reallocation of intra-team resources. Altering a course of action or team repertoire in response to changing conditions (internal or external).

Trait-2: Backup behavior-Ability to anticipate other team members' needs through accurate knowledge about their responsibilities. This includes the ability to shift workload among members to achieve balance during high periods of workload or pressure.

Trait-3: Closed-loop communication - The ability to develop common understandings of the team environment and apply appropriate task strategies to accurately monitor teammate performance.

Trait-4: Mutual performance monitoring- The ability to develop common understandings of the team environment and apply appropriate task strategies to accurately monitor teammate performance.

Trait-5: Mutual trust- The shared belief that team members will perform their roles and protect the interests of their teammates.

Trait-6: Shared mental models- An organizing knowledge structure of the relationships among the task the team is engaged in and how the team members will interact.

Trait-7: Shared Team leadership - Ability to direct and coordinate the activities of other team members, assess team performance, assign tasks, develop team knowledge, skills, and abilities, motivate team members, plan and organize, and establish a positive atmosphere.

Trait-8: Team orientation - Tendency to take other's behavior into account during group interaction and the belief in the importance of team goal's over individual members' goals.

Data Collection Sheet for Types' of System Developed

	Project Name/ID	Type of System	Complexity level / rating	
Sample Data	xyz	Banking system	Very High	Sample data

Types of systems
Defence/Military/Mission Critical Systems
DSS/EIS/Enterprise/Embedded Systems
Financial/Banking Systems (Mobile or Desktop)
General Purpose Systems
Other

Ratings/Gradings:	
	Very High
	High
	Moderate
	Low
	Very low

Note:

Plz select type of system which you want to develop or developed

Plz state other as type of system if your relevant system is not in the list

Plz choose level of complexity from the rating box

Data Collection Sheet for Org Type	
Type of IT Organization	Rating of influnce on Product development
(Where software developed or in-progress)	
Govt/ Public Sector	Very High
Type of organization given at the bottom of this sheet	Rating is given at bottom of this sheet

Organization Type
Government-sector
Private-sector
Joint-sector

Data Collection Sheet for Org Culture	
Type of Organization culture	Ratings of culture to influnce Product
Strong	Moderate
Type of culture along description given at bottom	Given below

Organization Culture
Strong
Semi-Strong
Fragile
Challenging/Toxic

Data Collection Sheet for Work Environment	
Work Environment	Ratings of environment to influnce product
Best	High
Work environment given	Ratings are given

Working Environment
Best
Semi-Best
Good
Poor

Data Collection Sheet for Project/Sprint Risks

	Project Name/ID	Project / Sprint level	Type of Risks	Influence Level
Sample Data	xyz	Project level	Known mitigated	Very Low
	xyz	Sprint-2 only	Known mitigated	Very Low
	xyz	Project level	Known non-mitigated	Modearte
	xyz	Project level	Unknown	Very High

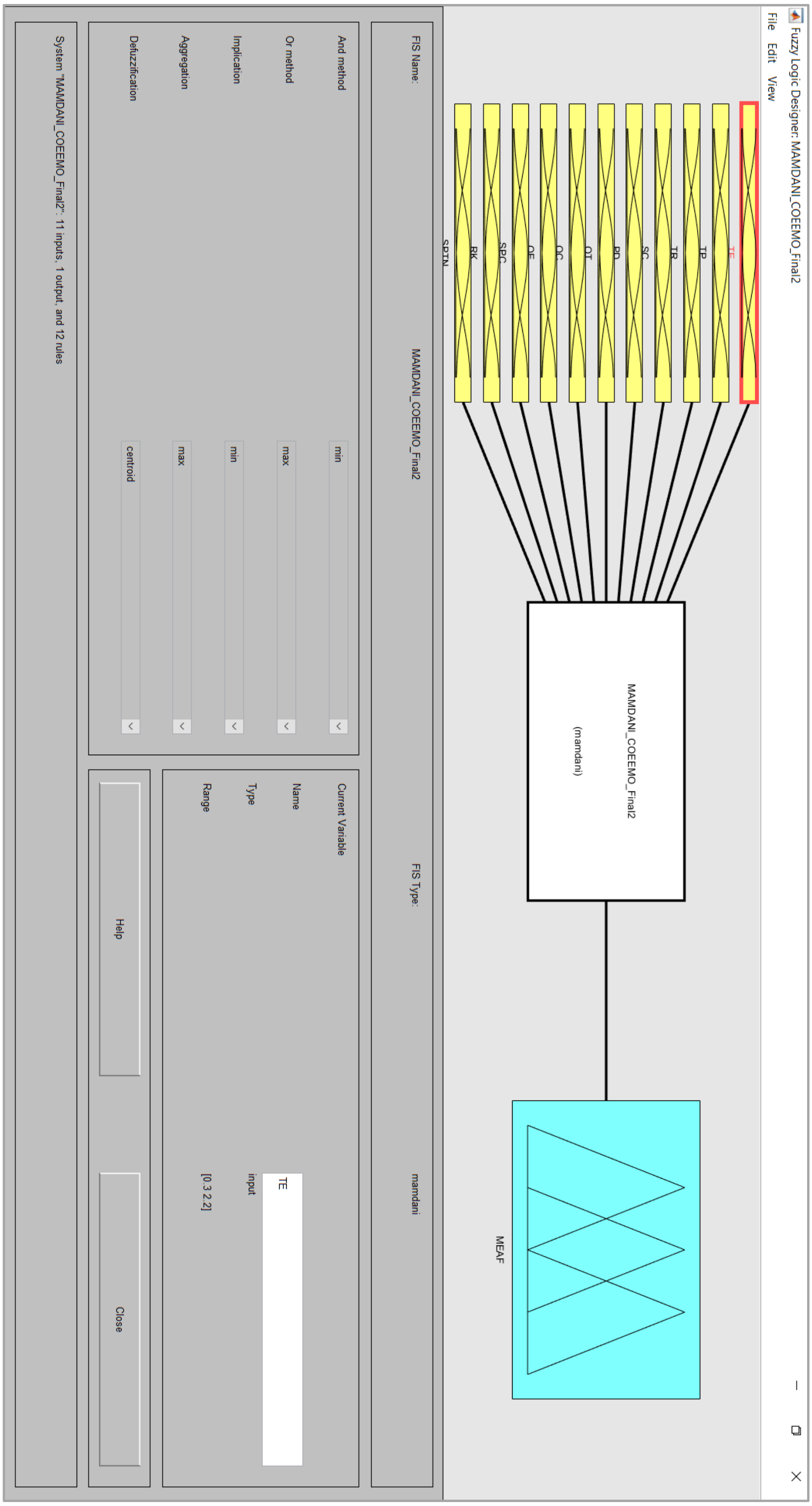
Risks' Description and how they influence project/sprint

Unmanaged risks is one of the (albeit numerous) reasons for the failure of software projects. Some of the risks linked with project failures are uncertainty regarding project complexity and project goals, and personality differences. This aspect also has the potential to impact on estimation. It is assumed that some risks are known and so they will be mitigated during the life of the project. However, some risks are unknown and therefore cannot be mitigated in advance, including the resignation of an active member of the team, long leave due to illness, and so on.

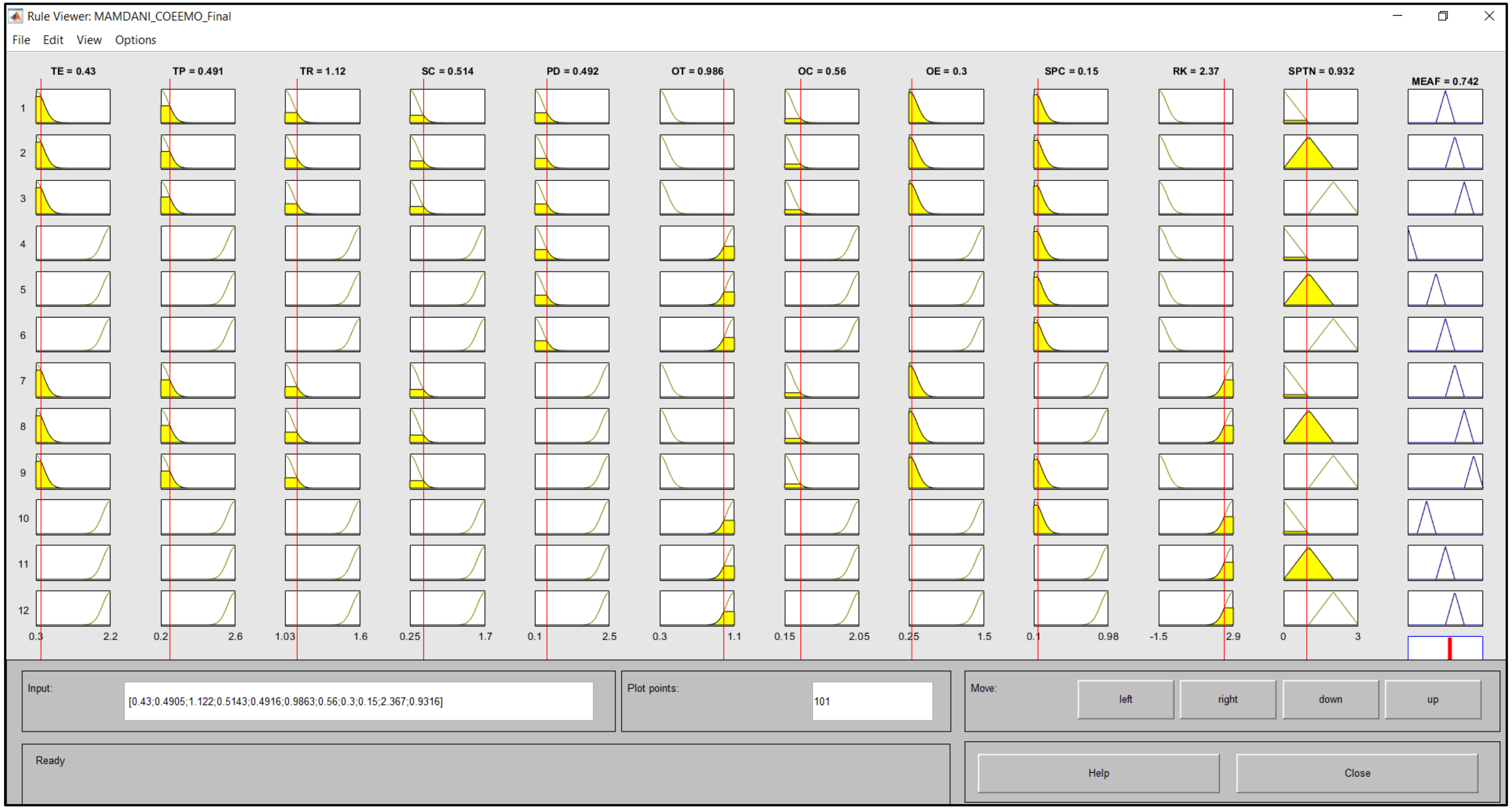
Risk Types
Known mitigated risk
Known non-mitigated risk
Unknown risk

Ratings
Very High
High
Moderate
Low
Very Low

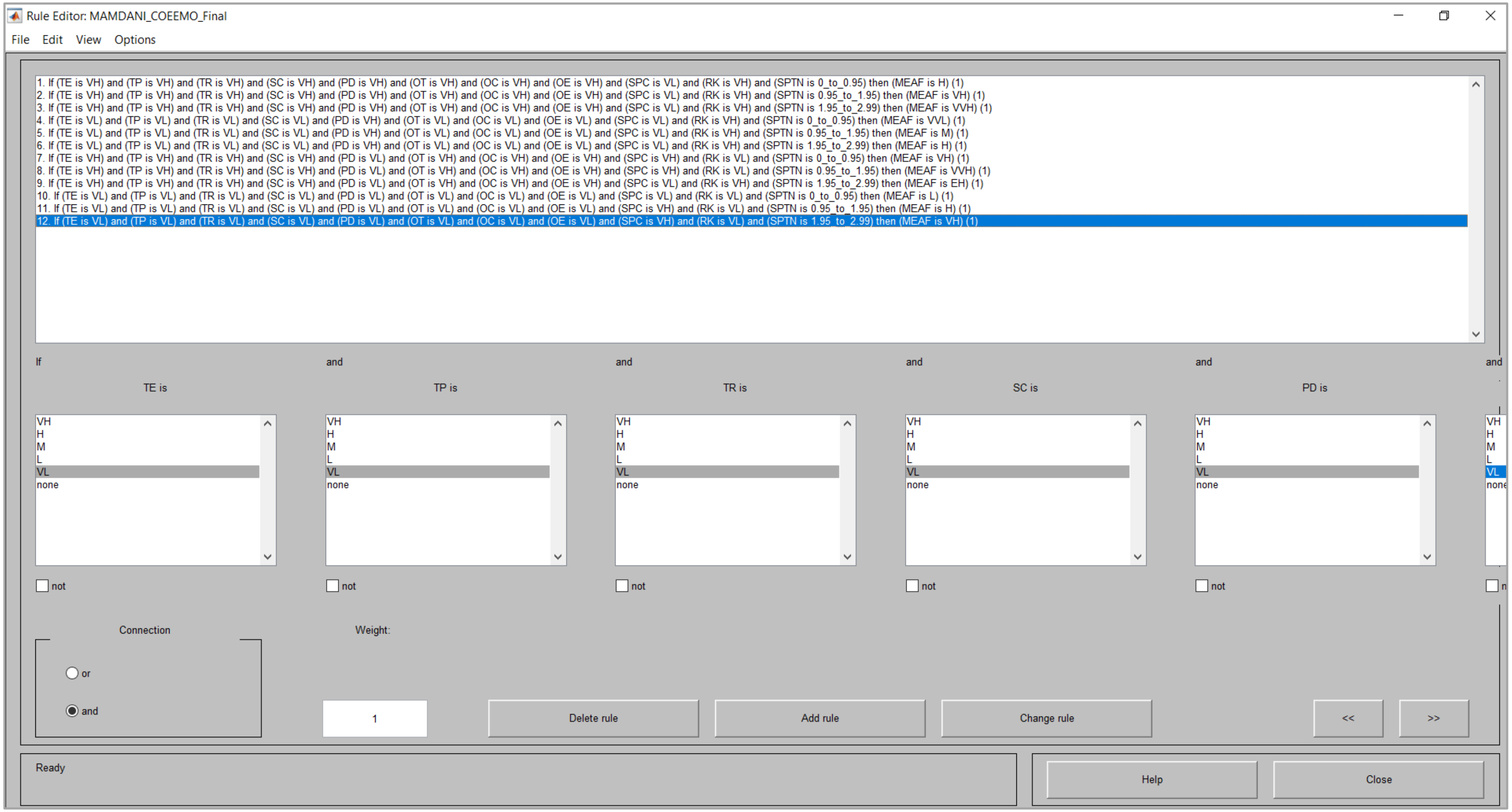
Appendix-F (COEEM-Mamdani Fuzzy Inference System Screenshots)



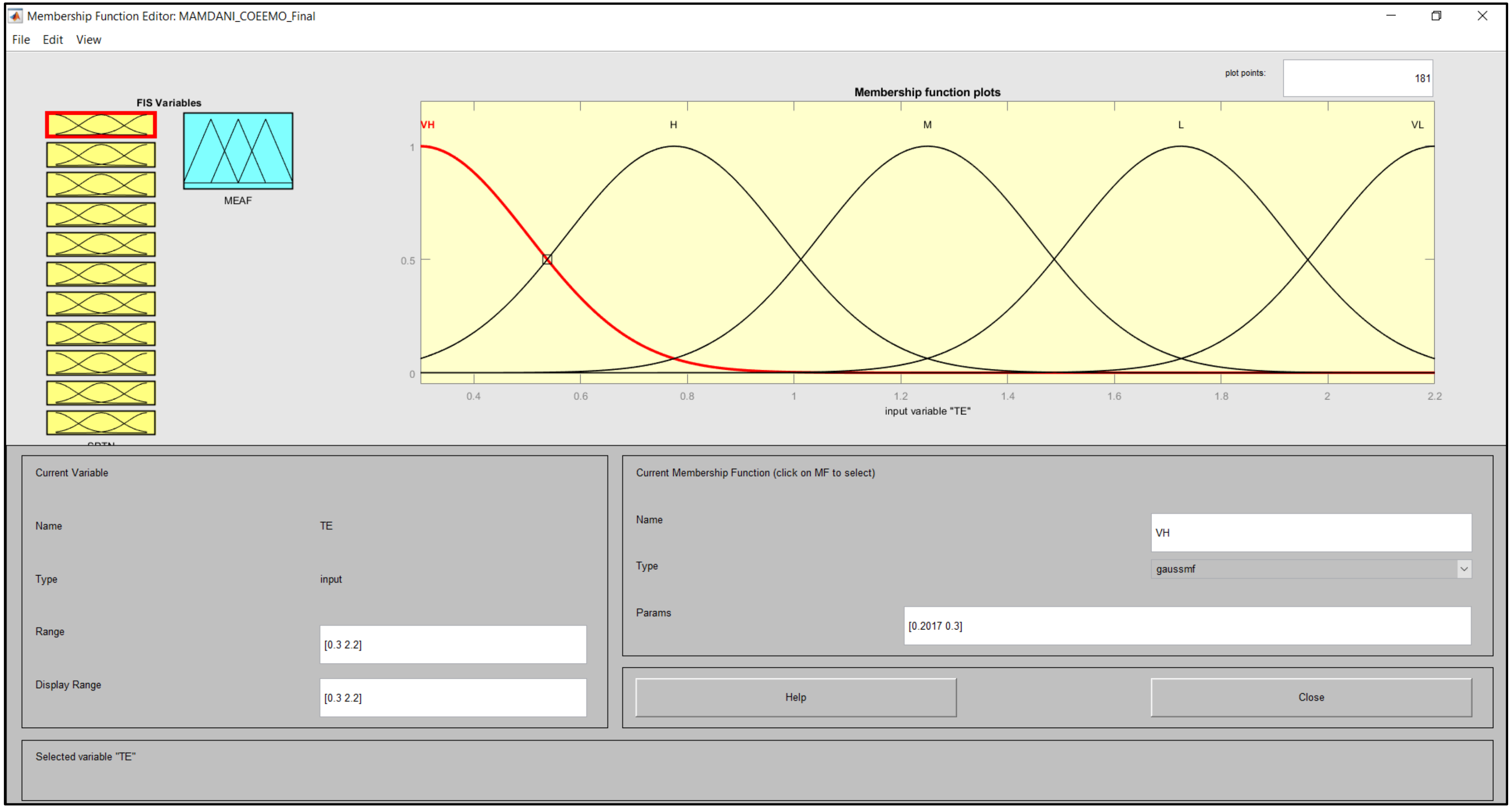
MFIS-COEEMO Rules Interface



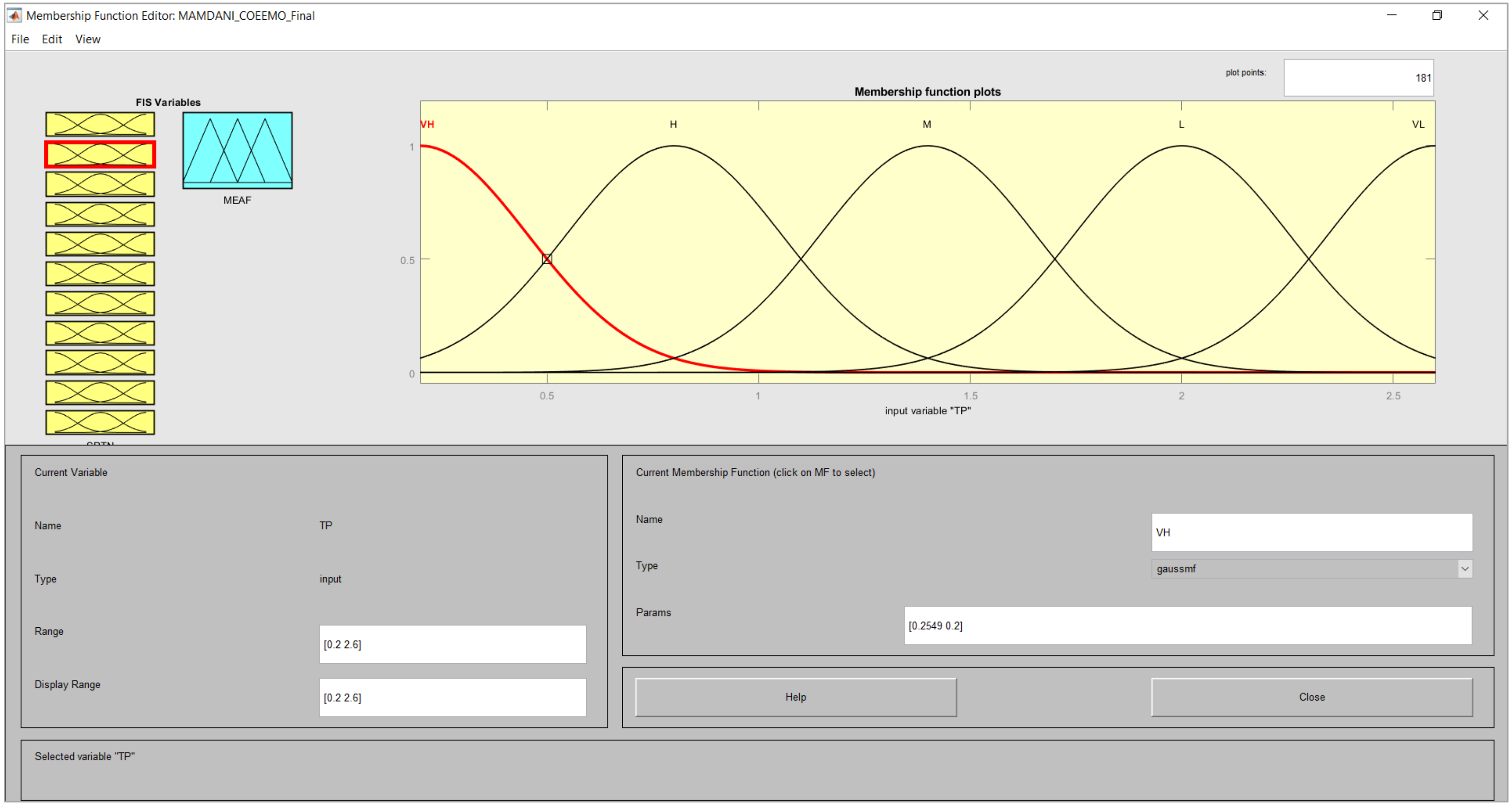
MFIS-COEEMO Rules Viewer



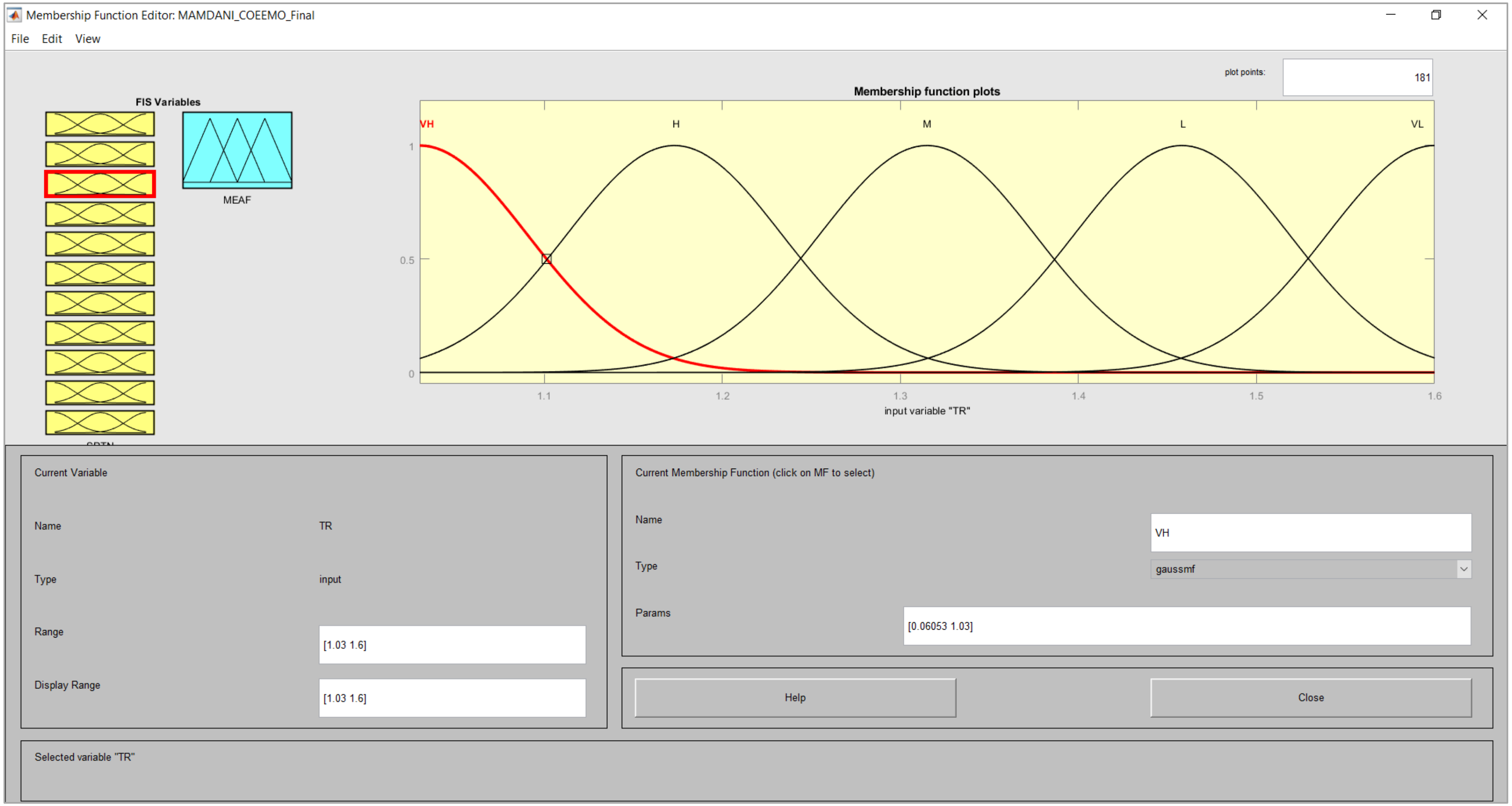
MFIS-COEEMO Rules



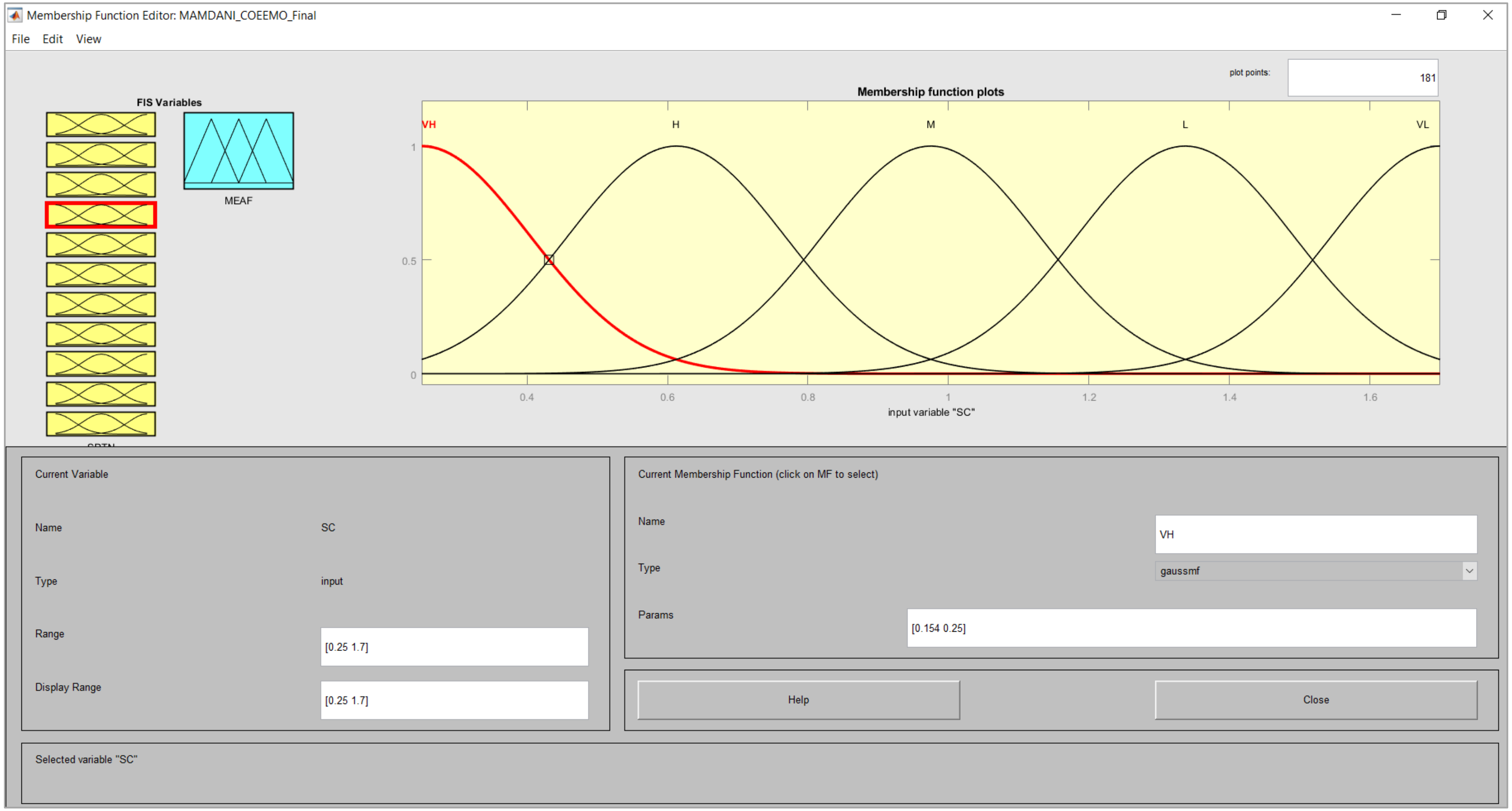
MFIS-COEEMO MF for Team Member



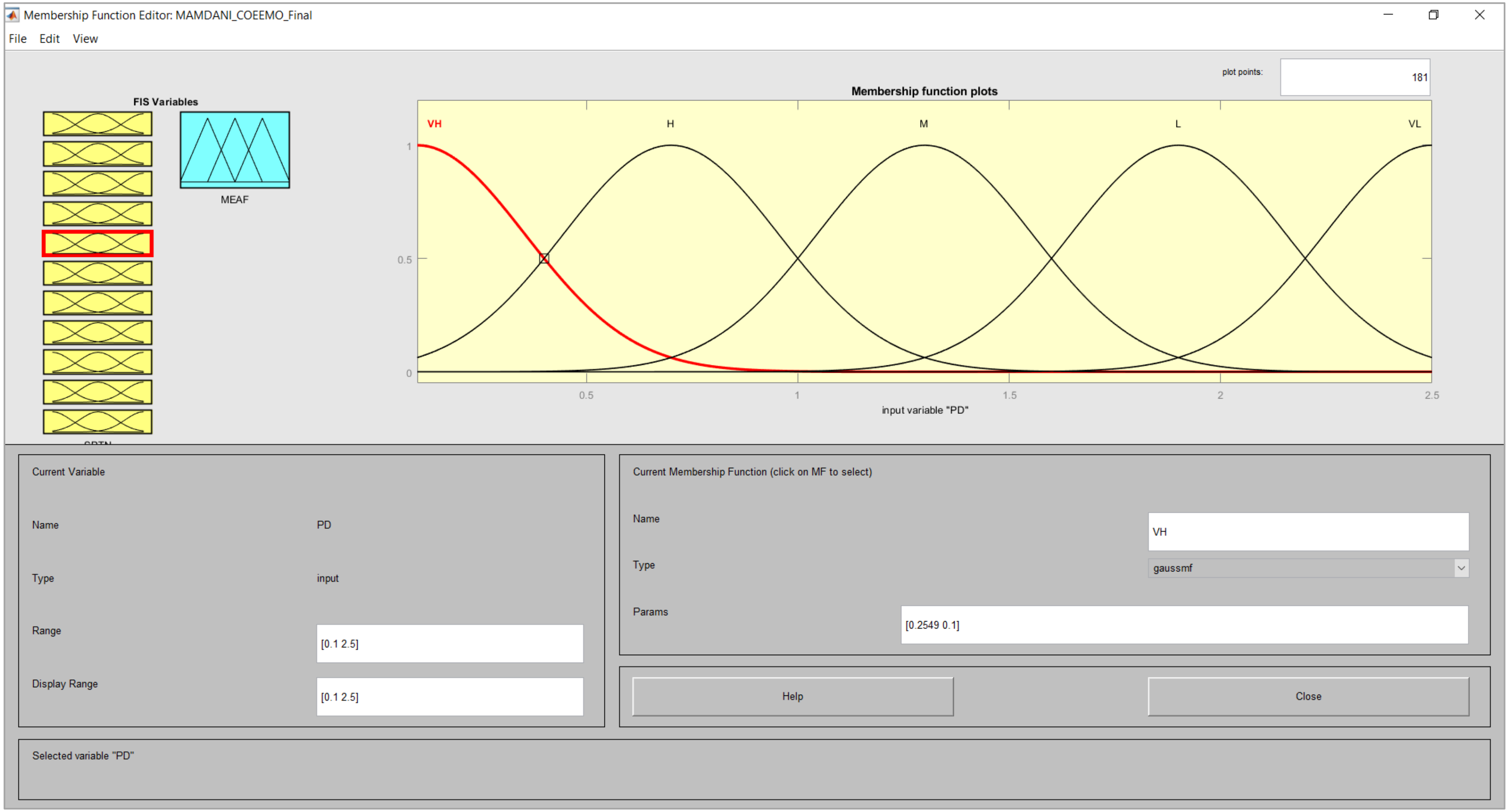
MFIS-COEEMO MF for Team Member



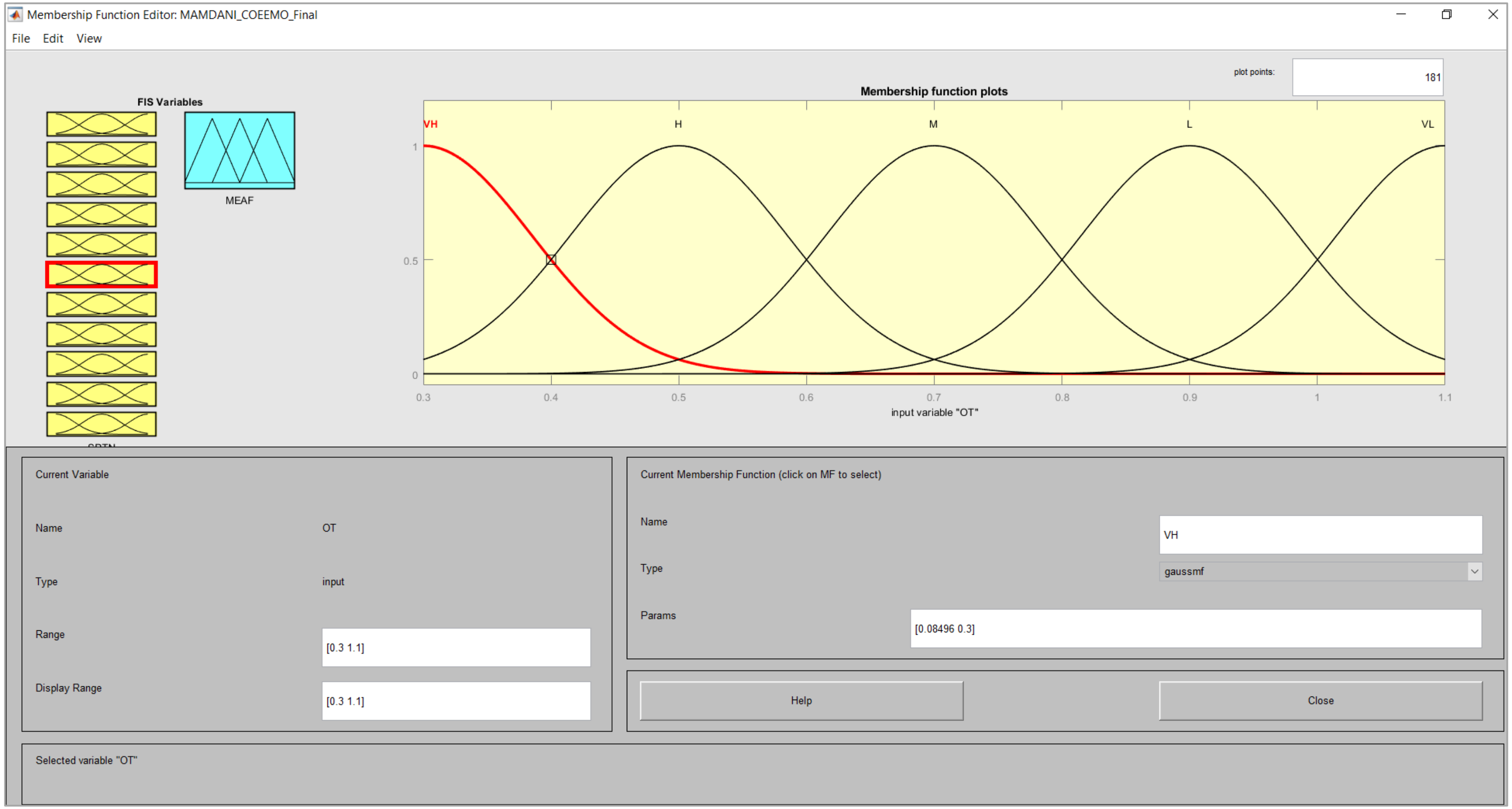
MFIS-COEEMO MF for Team Relationship



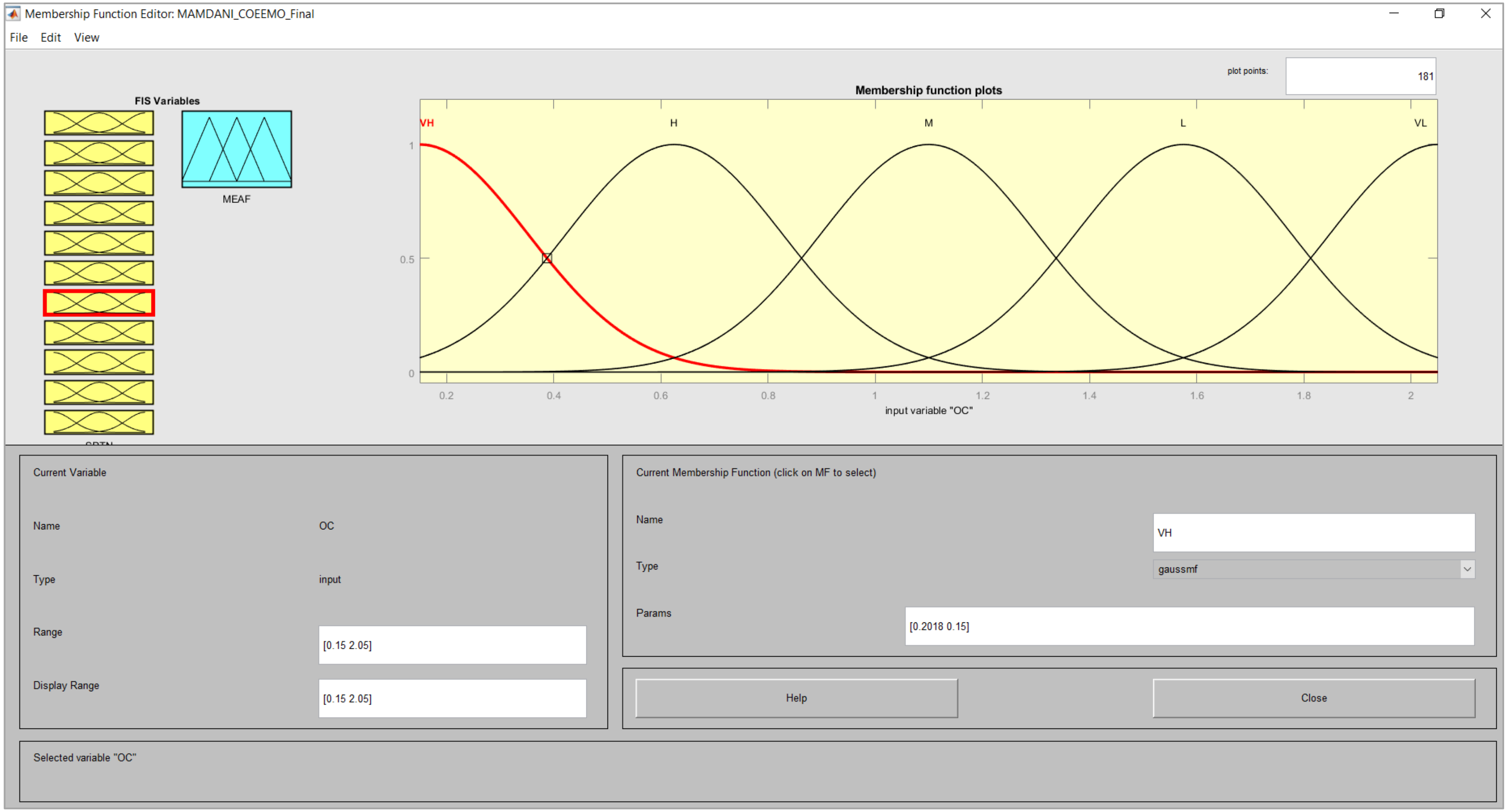
MFIS-COEEMO MF for Stakeholder



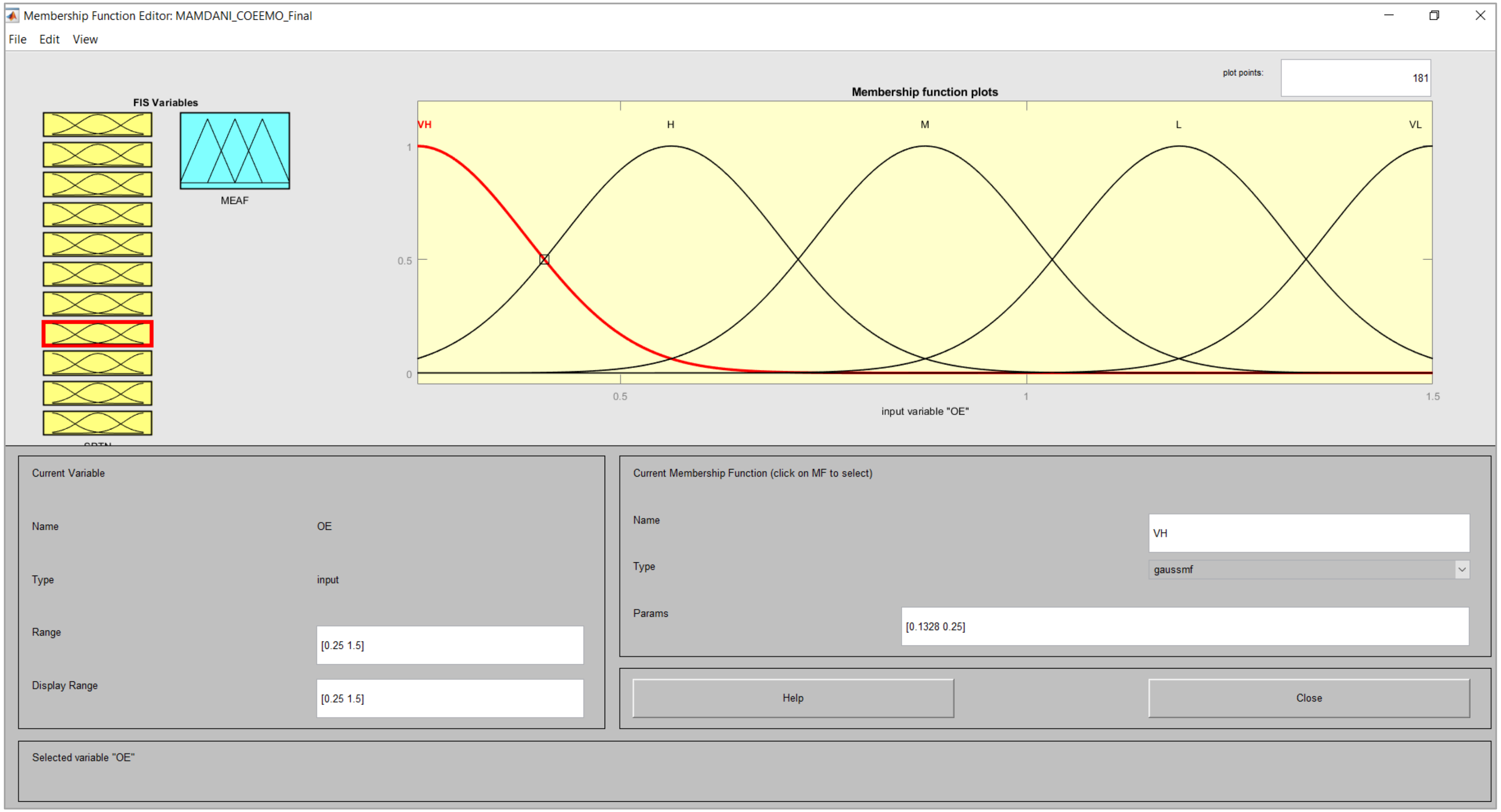
MFIS-COEEMO MF for System Complexity



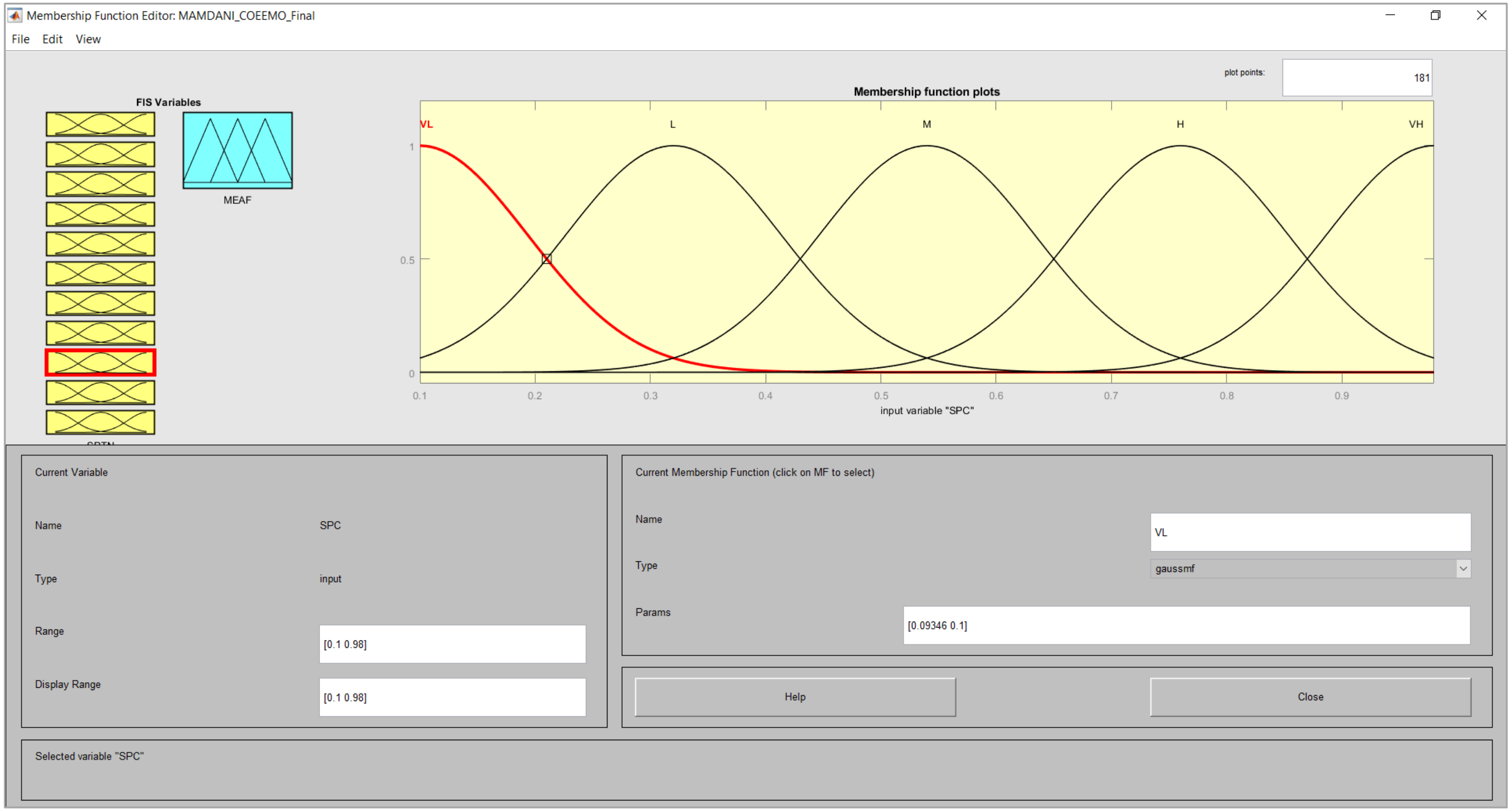
MFIS-COEEMO MF for Organization Type



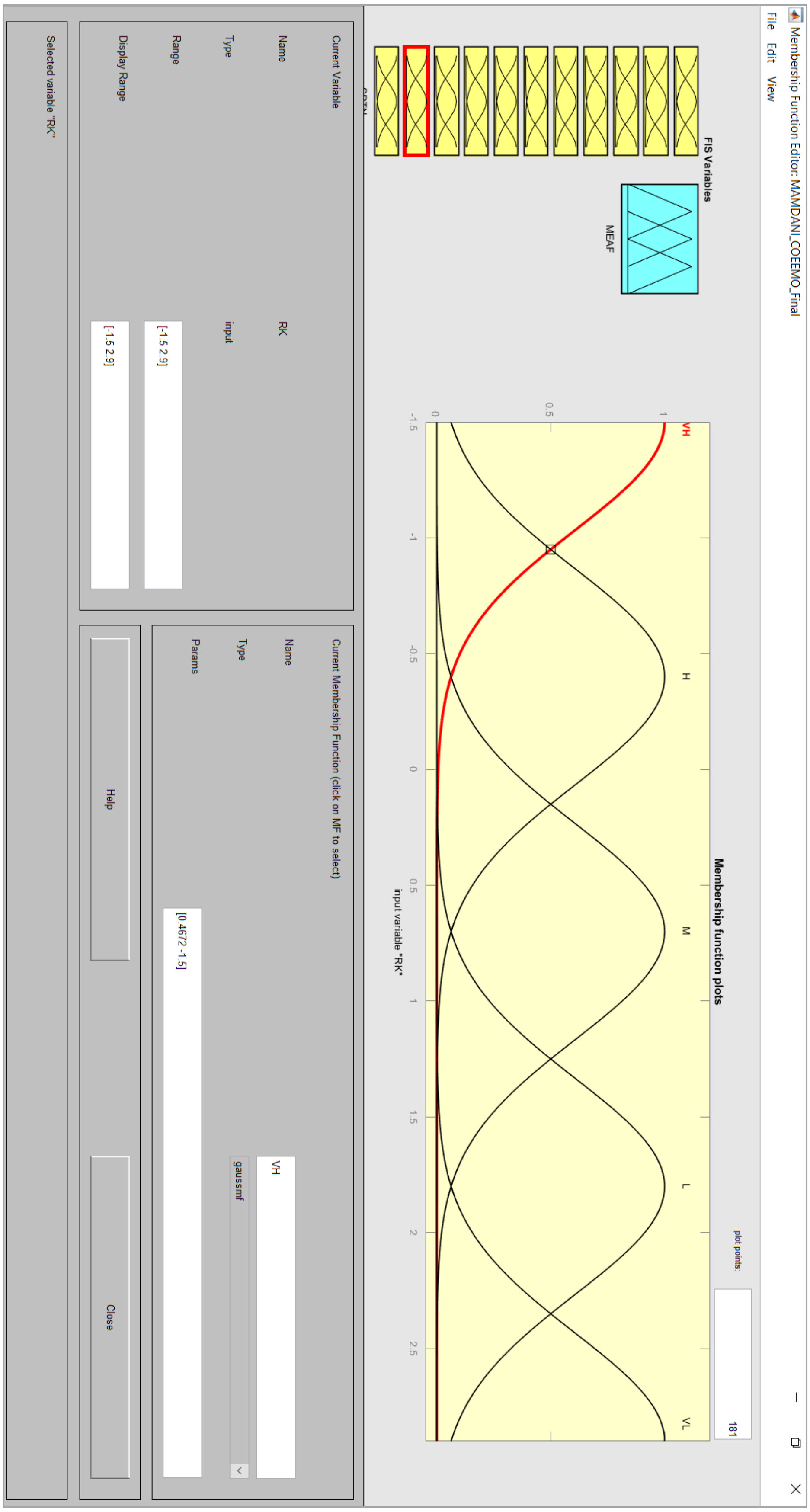
MFIS-COEEMO MF for Organization Culture



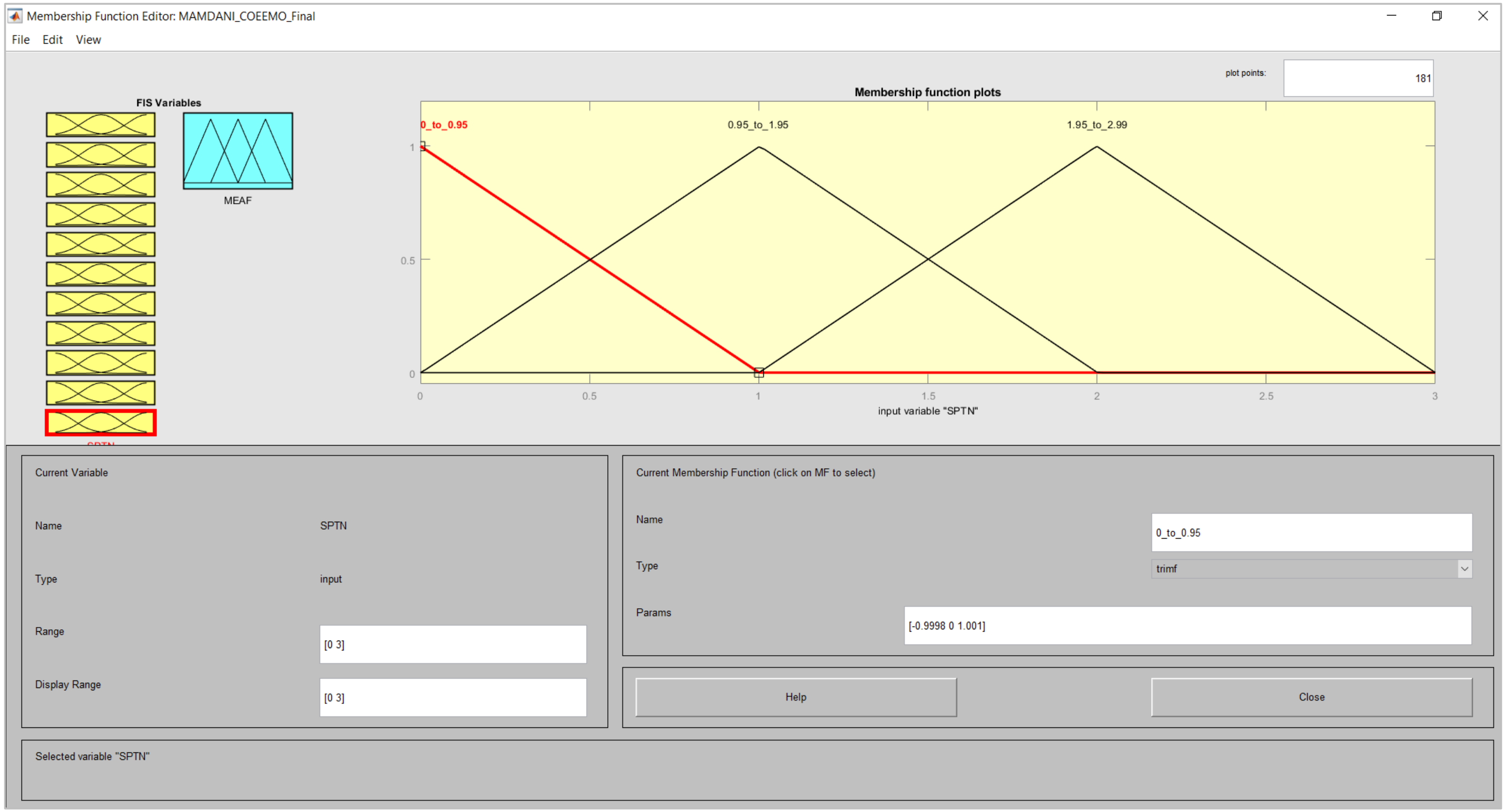
MFIS-COEEMO MF for Work Environment



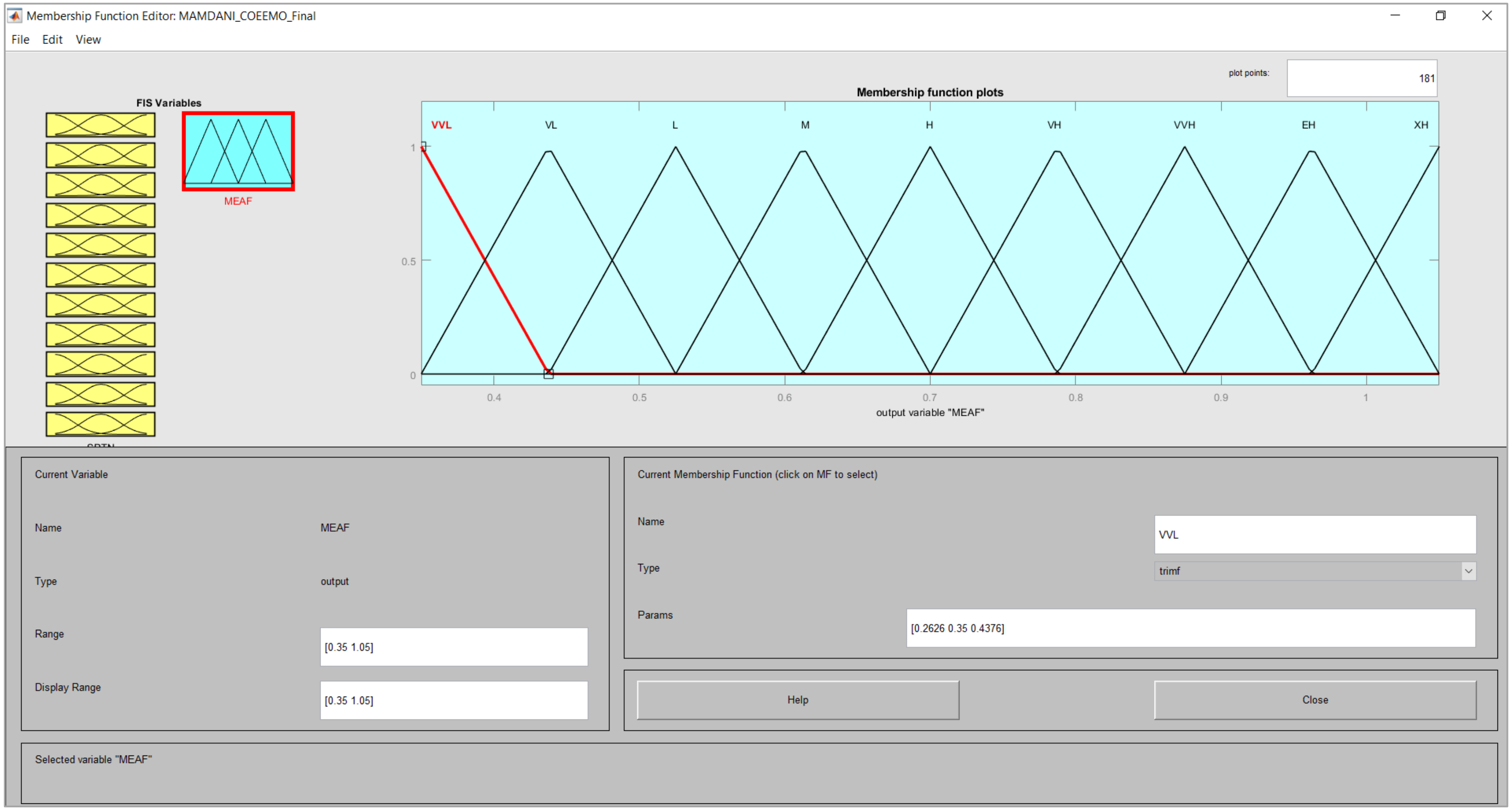
MFIS-COEEMO MF for Sprint Complexity



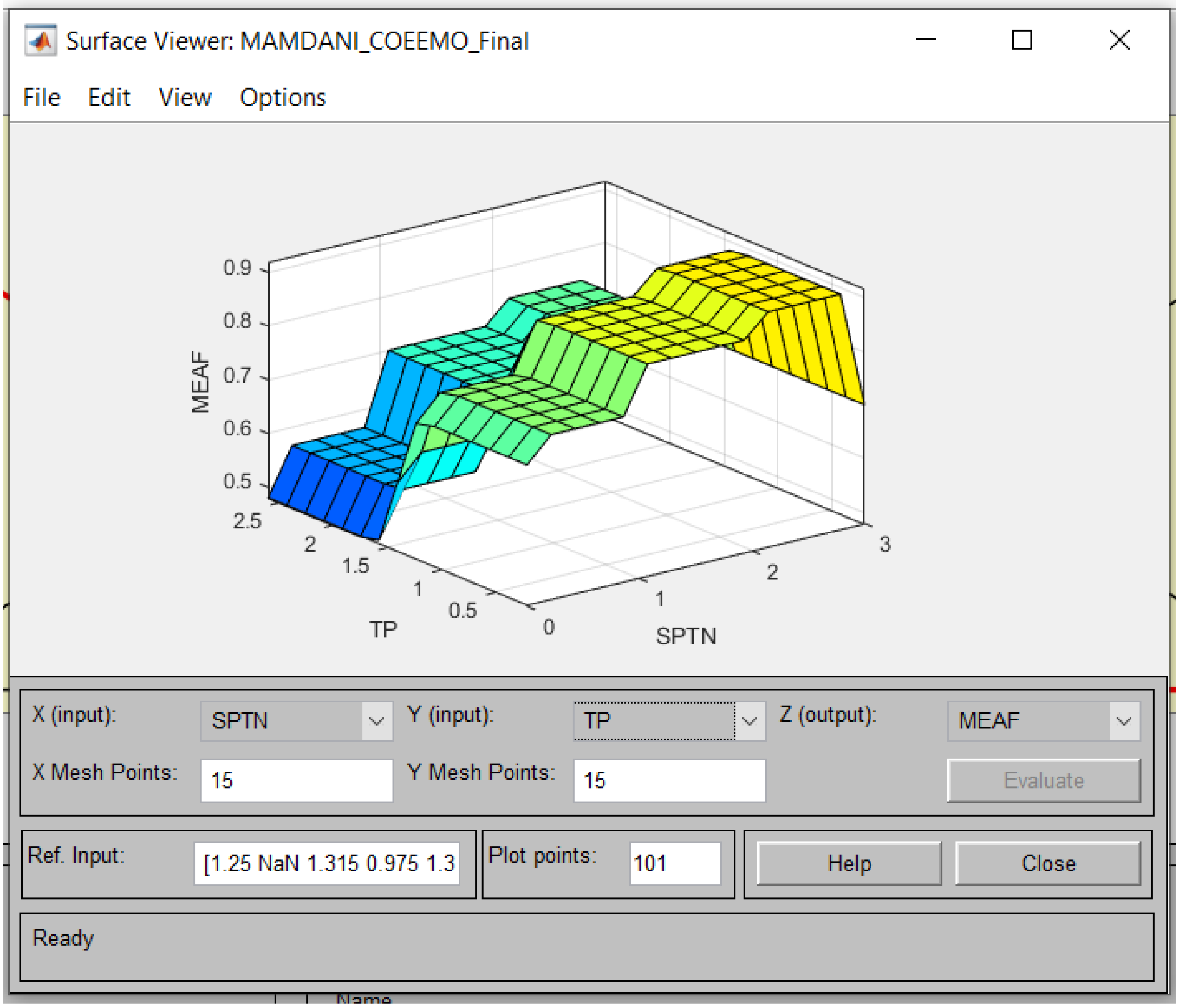
MFIS-COEEMO MF for Product Risks



MFIS-COEEMO MF to Control Sprint / Cycle

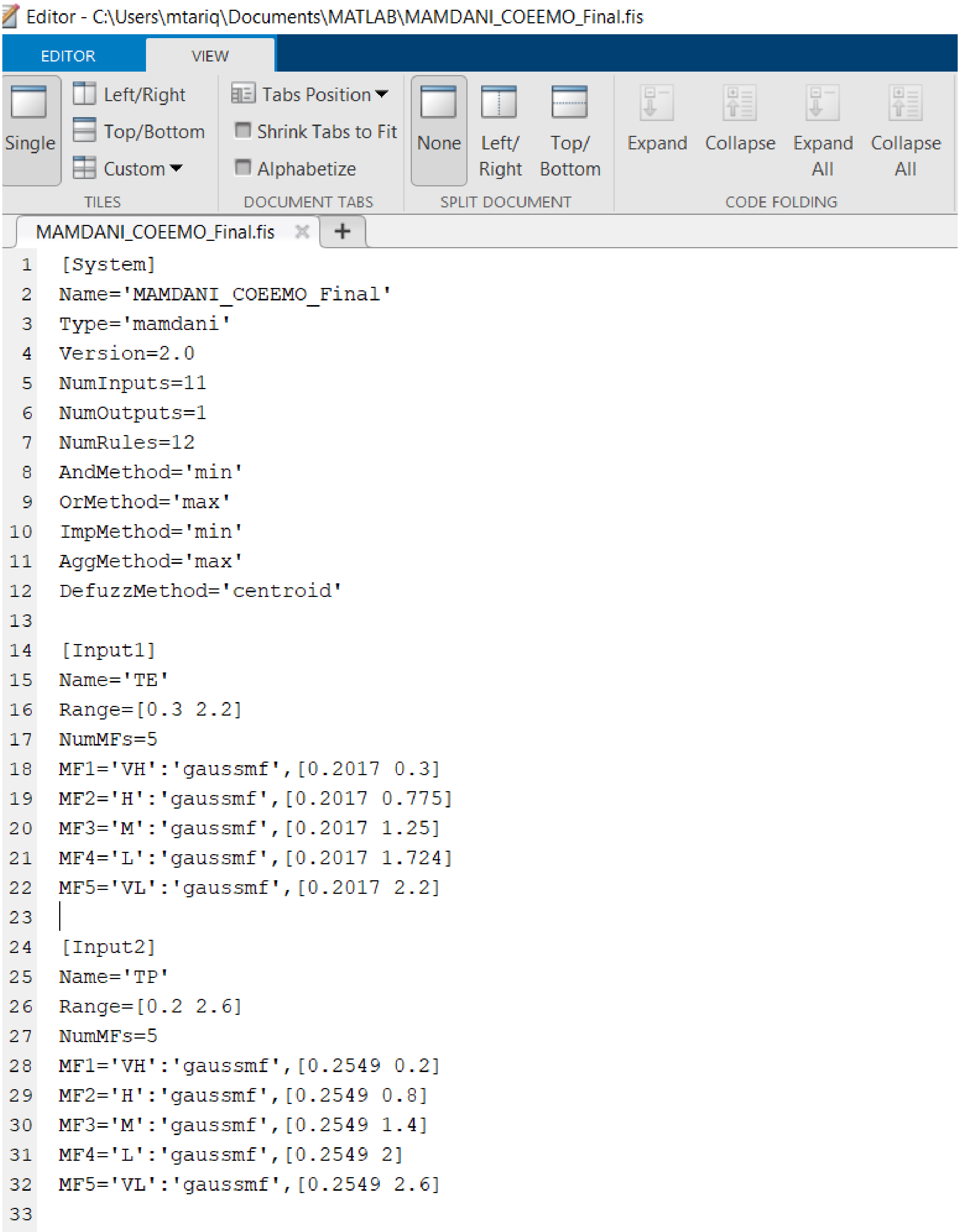


MFIS-COEEMO MF To Calculate MEAF Levels



<u>MFIS Surface Viewer</u>

MFIS-COEEMO Script Screenshots




```
[System]
Name='MAMDANI_COEEMO_Final'
Type='mamdani'
Version=2.0
NumInputs=11
NumOutputs=1
NumRules=12
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='centroid'

[Input1]
Name='TE'
Range=[0.3 2.2]
NumMFs=5
MF1='VH':'gaussmf',[0.2017 0.3]
MF2='H':'gaussmf',[0.2017 0.775]
MF3='M':'gaussmf',[0.2017 1.25]
MF4='L':'gaussmf',[0.2017 1.724]
MF5='VL':'gaussmf',[0.2017 2.2]

[Input2]
Name='TP'
Range=[0.2 2.6]
NumMFs=5
MF1='VH':'gaussmf',[0.2549 0.2]
MF2='H':'gaussmf',[0.2549 0.8]
MF3='M':'gaussmf',[0.2549 1.4]
MF4='L':'gaussmf',[0.2549 2]
MF5='VL':'gaussmf',[0.2549 2.6]

```

```
[Input3]
Name='TR'
Range=[1.03 1.6]
NumMFs=5
MF1='VH':'gaussmf',[0.06053 1.03]
MF2='H':'gaussmf',[0.06053 1.173]
MF3='M':'gaussmf',[0.06053 1.315]
MF4='L':'gaussmf',[0.06053 1.458]
MF5='VL':'gaussmf',[0.06053 1.6]

[Input4]
Name='SC'
Range=[0.25 1.7]
NumMFs=5
MF1='VH':'gaussmf',[0.154 0.25]
MF2='H':'gaussmf',[0.154 0.6125]
MF3='M':'gaussmf',[0.154 0.975]
MF4='L':'gaussmf',[0.154 1.337]
MF5='VL':'gaussmf',[0.154 1.7]

[Input5]
Name='PD'
Range=[0.1 2.5]
NumMFs=5
MF1='VH':'gaussmf',[0.2549 0.1]
MF2='H':'gaussmf',[0.2549 0.7]
MF3='M':'gaussmf',[0.2549 1.3]
MF4='L':'gaussmf',[0.2549 1.9]
MF5='VL':'gaussmf',[0.2549 2.5]

[Input6]
Name='OT'
Range=[0.3 1.1]
```

```
[Input6]
Name='OT'
Range=[0.3 1.1]
NumMFs=5
MF1='VH':'gaussmf',[0.08496 0.3]
MF2='H':'gaussmf',[0.08496 0.5]
MF3='M':'gaussmf',[0.08496 0.7]
MF4='L':'gaussmf',[0.08496 0.9]
MF5='VL':'gaussmf',[0.08496 1.1]

[Input7]
Name='OC'
Range=[0.15 2.05]
NumMFs=5
MF1='VH':'gaussmf',[0.2018 0.15]
MF2='H':'gaussmf',[0.2018 0.625]
MF3='M':'gaussmf',[0.2018 1.1]
MF4='L':'gaussmf',[0.2018 1.575]
MF5='VL':'gaussmf',[0.2018 2.05]

[Input8]
Name='OE'
Range=[0.25 1.5]
NumMFs=5
MF1='VH':'gaussmf',[0.1328 0.25]
MF2='H':'gaussmf',[0.1328 0.5625]
MF3='M':'gaussmf',[0.1328 0.875]
MF4='L':'gaussmf',[0.1328 1.188]
MF5='VL':'gaussmf',[0.1328 1.5]

[Input9]
Name='SPC'
Range=[0.1 0.98]
```

```
NumMFs=5
MF1='VL':'gaussmf',[0.09346 0.1]
MF2='L':'gaussmf',[0.09346 0.32]
MF3='M':'gaussmf',[0.09346 0.54]
MF4='H':'gaussmf',[0.09346 0.76]
MF5='VH':'gaussmf',[0.09346 0.98]

[Input10]
Name='RK'
Range=[-1.5 2.9]
NumMFs=5
MF1='VH':'gaussmf',[0.4672 -1.5]
MF2='H':'gaussmf',[0.4672 -0.4]
MF3='M':'gaussmf',[0.4672 0.7]
MF4='L':'gaussmf',[0.4672 1.8]
MF5='VL':'gaussmf',[0.4672 2.9]

[Input11]
Name='SPTN'
Range=[0 3]
NumMFs=3
MF1='0_to_0.95':'trimf',[-0.9998 0 1.001]
MF2='0.95_to_1.95':'trimf',[0.00231303006939099 1.00331303006939 2.00131303006939]
MF3='1.95_to_2.99':'trimf',[1.001 1.999 3]

[Output1]
Name='MEAF'
Range=[0.35 1.05]
NumMFs=9
MF1='VVL':'trimf',[0.2625 0.35 0.4375]
MF2='VL':'trimf',[0.35 0.4375 0.525]
MF3='L':'trimf',[0.4375 0.525 0.6125]
MF4='M':'trimf',[0.525 0.6125 0.7]
```

```
MF4='M':'trimf',[0.525 0.6125 0.7]
MF5='H':'trimf',[0.6125 0.7 0.7875]
MF6='VH':'trimf',[0.7 0.7875 0.875]
MF7='VVH':'trimf',[0.7875 0.875 0.9625]
MF8='EH':'trimf',[0.875 0.9625 1.05]
MF9='XH':'trimf',[0.9625 1.05 1.138]

[Rules]
1 1 1 1 1 1 1 1 1 1 1, 5 (1) : 1
1 1 1 1 1 1 1 1 1 1 2, 6 (1) : 1
1 1 1 1 1 1 1 1 1 1 3, 7 (1) : 1
5 5 5 5 1 5 5 5 1 1 1, 1 (1) : 1
5 5 5 5 1 5 5 5 1 1 2, 4 (1) : 1
5 5 5 5 1 5 5 5 1 1 3, 5 (1) : 1
1 1 1 1 5 1 1 1 5 5 1, 6 (1) : 1
1 1 1 1 5 1 1 1 5 5 2, 7 (1) : 1
1 1 1 1 5 1 1 1 1 1 3, 8 (1) : 1
5 5 5 5 5 5 5 5 1 5 1, 3 (1) : 1
5 5 5 5 5 5 5 5 5 5 2, 5 (1) : 1
5 5 5 5 5 5 5 5 5 5 3, 6 (1) : 1
```

Appendix-G (Consent Form to Participate in Experiment)

Consent Form to Participate in Experiments

CONSENT FORM

For use when laboratory or field experiments are involved.

Project title: Effort Estimation in Agile Software Development

Project Supervisor: *Professor Stephen MacDonell*

Researcher: *Muhammad Tariq*

- ○ I have read and understood the information provided about this research project in the Information Sheet dated 20th April 2018.
- ○ I have had an opportunity to ask questions and to have them answered.
- ○ I understand that taking part in this study is voluntary (my choice) and that I may withdraw from the study at any time without being disadvantaged in any way.
- ○ I understand that if I withdraw from the study then I will be offered the choice between having any data or tissue that is identifiable as belonging to me removed or allowing it to continue to be used. However, once the findings have been produced, removal of my data may not be possible.
- ○ I understand that my feedback / data may also be used for conducting subsequent research in the aforementioned area of research.
- ○ I am not suffering from any illness or injury that impairs my physical performance in the intended research
- ○ I agree to take part in this research.
- ○ I wish to receive the research findings (please tick one): Yes○ No○

Participant's signature: ....……………………………………………………

Participant's name: ....……………………………………………………

Participant's Contact Details (if appropriate):

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

……………………………………………………………………………………..

Date:

Approved by the Auckland University of Technology Ethics Committee on 24-April-2018 AUTEC Reference number 18/153

Note: The Participant should retain a copy of this form.

Appendix-H (Consent Form to Participate in Interview)

Consent Form to Participate in Interview

CONSENT FORM

For use when interviews are involved.

Project title: Effort Estimation in Agile Software Development
Project Supervisor: *Professor Stephen MacDonell*
Researcher: *Muhammad Tariq*

- ○ I have read and understood the information provided about this research project in the Information Sheet dated 20th April 2018.
- ○ I have had an opportunity to ask questions and to have them answered.
- ○ I understand that notes will be taken during the interviews and that they will also be audio-taped and transcribed.
- ○ I understand that taking part in this study is voluntary (my choice) and that I may withdraw from the study at any time without being disadvantaged in any way.
- ○ I understand that if I withdraw from the study then I will be offered the choice between having any data that is identifiable as belonging to me removed or allowing it to continue to be used. However, once the findings have been produced, removal of my data may not be possible.
- ○ I understand that my feedback / data may also be used for conducting subsequent research in the aforementioned area of research.
- ○ I agree to take part in this research
- ○ I wish to receive a copy of the research findings (please tick one): Yes○ No○

Participant's signature: ....……………………………………………………………

Participant's name: ....……………………………………………………………

Participant's Contact Details (if appropriate):

………………………………………………………………………………………..
………………………………………………………………………………………..
………………………………………………………………………………………..
………………………………………………………………………………………..

Date:

Approved by the Auckland University of Technology Ethics Committee on 24-April-2018 AUTEC Reference number 18/153.

Note: The Participant should retain a copy of this form.

Appendix-I (Information Sheet for Participant)

Information Sheet for Participants

Participant Information Sheet

Date Information Sheet Produced:

20 April 2018

Project Title

Effort Estimation and Project Scheduling in Agile Software Development

An Invitation

My name is **Muhammad Tariq**, I am a PhD Researcher in the School of Engineering, Computer and Mathematical Sciences, AUT, Auckland. I am working in the area of effort estimation and scheduling in agile software development. My intended study requires the participation of personnel involved in estimation and scheduling activities, including Product owners, Scrum masters, Project Managers, Quality Assurance Managers, Team leads, and agile team members, in Auckland and around the globe. Therefore, I would like to invite you to participate in my research.

There are two phases of participation in my study. The first is an initial validation phase, where I need your participation as an interviewee, while the second is a field experimentation phase, where you may participate as an implementer. However, you are open to choose to participate in either one or both of the aforementioned phases depending upon your own choice and interest. Involvement in either will be highly appreciated. Your knowledge, experience and expertise will help us to evaluate and improve the model that I am building for effort estimation and scheduling in agile contexts. Please note that your participation is entirely voluntary and you may withdraw at any time without any adverse consequences.

What is the purpose of this research?

This research is seeking to better understand and improve the utility of forecasts of size, effort, cost and schedule for Agile Software Development (ASD) projects through various approaches that take account of a range of relevant objective and subjective indicators. Some of these indicators include but are not limited to the size of stories, the velocity/capacity of the team, the numbers of sprints, the domain knowledge of the team/experts, and the organizational context. Final selection of the relevant indicators is dependent upon stakeholder feedback, from relevant individuals such as you, as well as the outcomes of statistical analyses. The results of the research will be included in my PhD thesis and may also be published in scholarly outlets such as international peer-reviewed academic journals.

How was I chosen for this invitation?

You were sent this information sheet through our contact with your CIO. They have nominated you because they consider you to have relevant experience in software effort estimation and/or in one or more of the following software development processes: requirements engineering, design and architecture, development and coding, quality assurance and testing, implementation, maintenance and management.

What will happen in this research?

You may choose to be involved in interviews or field experiments, or both. If you opt for the interview then you will be interviewed by the researcher at your suggested place and time. An interview schedule will be sent to you in advance. During the interview you will be given an opportunity to consider the factors you consider to be relevant or significant in relation to the estimation of effort for ASD projects, from the perspective of your own expertise, team relationship, work environment and others. Interviews will be recorded using a digital recorder and later transcribed by me; these notes will be shared with you so that you can check that they are correct.

If you opt for the experimentation phase then you will be given an opportunity to contribute to a short sequence of experiments to systematically evaluate the impact of a range of forecasting techniques and their variants (Top-Down, Bottom-Up and PERT) along with the proposed model of estimation so that these can be compared with your organisation's existing method of estimation. These experiments will be conducted at your own work place. It is expected that your team will apply the techniques that I suggest in your ongoing or new projects, with me there to lend support. During the experiments I will collect data on how the techniques are used and I will guide the team as needed regarding the suggested techniques. The duration of data collection may be 8-12 weeks depending upon the number and duration of development and release cycles.

What are the discomforts and risks?

No risks and discomforts are envisaged. However, in your particular organization you may need to obtain the agreement of your senior manager to participate in the research.

How will these discomforts and risks be alleviated?

For the agreement of your senior manager if it is sought, I will provide additional information if s/he may ask in order to more fully consider the invitation.

What are the benefits?

The research will contribute to the current body of knowledge in terms of novel estimation models and improved forecasts of size, effort, cost and schedule for ASD projects. From an industry perspective, the said research should help agile teams to become more effective and productive by avoiding under- and over-commitments to tasks during the ASD project development cycle. Ultimately this should improve the overall confidence of customers and management regarding a project's cost and schedule. I will also benefit in that I will receive a degree qualification.

How will my privacy be protected?

Your CIO will know that you have been suggested for this research, but they may not know whether or not you are an interview participant. Your colleagues will know if you are participating in the experiments. Your identity will be known to the researchers but the data gathered from you will not contain any personal details. Furthermore, the data will be used in such a way that your identity and those of your respective organisations or companies will be protected from disclosure. You will not know who else outside your organisation is participating in the project. The digital records and the transcriptions will be accessible only to the researchers.

What are the costs of participating in this research?

There will be no cost to participate in this research. However, the expected time for interview will take between 60 to 90 minutes overall but not more than 2 hours.

What opportunity do I have to consider this invitation?

I would be grateful if you were able to respond to my invitation within 10 working days of receiving it. Please also let me know whether further information about the research or this request is needed.

How do I agree to participate in this research?

To indicate that you agree to take part in the research please complete the attached Consent Form and email it to me as an attachment. Your participation in this research is voluntary (it is your choice) and whether or not you choose to participate will neither advantage nor disadvantage you. You are able to withdraw from the study at any time. If you choose to withdraw from the study, then you will be offered the choice between having any data that is identifiable as belonging to you removed or allowing it to continue to be used. However, once the findings have been produced, removal of your data may not be possible. The data will be kept for an indefinite period of time and may be utilized for subsequent related research.

Will I receive feedback on the results of this research?

Once the PhD is successfully completed I will send you an electronic copy of my thesis.

What do I do if I have concerns about this research?

Any concerns regarding the nature of this project should be notified in the first instance to the Project Supervisor, Professor Stephen MacDonell, smacdone@aut.ac.nz

Concerns regarding the conduct of the research should be notified to the Executive Secretary, AUTEC, Kate O'Connor, ethics@aut.ac.nz, +64 9-921-9999 extn: 6038.

Whom do I contact for further information about this research?

Researcher Contact Details:

Muhammad Tariq, PhD Researcher, School of Engineering, Computer and Mathematical Sciences, AUT University.
Email: mtariq@aut.ac.nz
Phone: +64 9-921-9999 extn: 6376.

Project Supervisor Contact Details:

Professor Stephen MacDonell, School of Engineering, Computer and Mathematical Sciences, AUT University
Email: smacdone@aut.ac.nz

Approved by the Auckland University of Technology Ethics Committee on *24-April-2018*, AUTEC Reference number *18/153*.

Appendix-J (Email Invitation)

Email

Invitation to participate in research: **EFFORT ESTIMATION AND PROJECT SCHEDULING IN AGILE SOFTWARE DEVELOPMENT**

My name is Muhammad Tariq, and I am a PhD researcher in the School of Engineering, Computer and Mathematical Sciences at AUT University in New Zealand. I am working on a project that aims to understand and improve the utility of forecasts of size, effort, cost and schedule for agile software development projects**.**

I would like to invite you to participate in my research to address issues that are relevant to effort estimation and project scheduling in agile software development. Your agreement to participate will be greatly appreciated, and should be beneficial for you, your organisation and the wider software industry. Your knowledge, experience and expertise will directly contribute to the building of a new model of estimation, addressing some of the challenges that arise specifically in agile software development contexts. Please note that your participation is entirely voluntary and you may withdraw at any time without any adverse consequence.

You can find more information about the project in the Information Sheet attached.

I hope that you will agree to take part in this research. To indicate your agreement please complete the attached Consent Form(s) and return it (them) to me at mtariq@aut.ac.nz, within 10 days of receiving this invitation. Please also let me know if you have any questions or if you need further information – I will be happy to provide clarification as needed.

Sincerely yours,

Appendix-K (Interview Guide)

Interview Guide

Topic 1 Position (10-15 min)

1. What is the title of your current position?
2. How would you describe your current occupation to others?
3. How long (years, months) have you been involved in this occupation?
    a) in this company:
    b) in other companies:
4. Please list your top three areas of professional expertise.
5. You are currently involved in the development of a software project/product development. Please describe your role in this development, and your responsibilities in it.
6. What motivated you to become involved in this development?

Topic 2 Software product/project development experience (10-15 min)

7. What kinds of projects/products does your company mainly develop e.g. Web, Desktop, mobile, built-in/embedded, games etc.?

8. Does your project/product involve global development?

9. What practices do you regularly or often employ in software development?

10. Do you have any experience estimating the size, effort, cost or schedule of a software development project? If yes, then which methodology or methodologies you have used for the purposes of estimation?

11. Please describe any problems / challenges you might have experienced in estimating the size, effort, cost or schedule of a software development project in which you were involved?

Topic 3 Software Estimation and Scheduling (40-45 min)

[To facilitate this component of the interview session a questionnaire is attached as Annex-A. This questionnaire will be provided to the interviewees and they will go through it and fill it in the presence of the researcher, and at the end of the interview it will be handed back to the researcher].

The questionnaire comprises a list of factors that are each thought to have some level of influence on software estimation. A few have been deduced from recent surveys whereas others have been included by the researcher. The intent is to analyse their relative impact with respect to estimates of size, effort, cost and scheduling of projects undertaken in agile development contexts. It may be that the given factors are not the only factors that impact estimation; therefore, suggestions from the interviewees for additional factor(s) are welcomed.

Annex-A
Individual factors descriptions along with their levels

The factors along with their categories considered significant for the proposed study are given as follows whereas their short descriptions are covered in the succeeding paragraphs:

k. Human context
   1. Team member's expertise
   2. Team member's experience
   3. Team relationship and spirit
   4. Stakeholder / user collaboration

l. System context

m. Organization context
   1. Organization type
   2. Organization culture
   3. Work place

n. Sprint context

o. Risk context
   1. Known (mitigated and non-mitigated)
   2. Unknown

Note: Interviewees may provide additional factors along with some possible justifications here:

Short Descriptions of the factors and selection of ratings:

1. **Team Member Expertise**

   Practically this is an ability to classify and use experts in a wide range of real-world scenarios (e.g., an expert may be used to predict project scheduling). For this proposed study, expertise means knowledge or skill that is acquired through practice irrespective of its duration. It is suggested to be measured across four level of expertise (High, Medium, Adequate, and Inadequate) based on the team member's domain-specific knowledge (means the demonstrable academic and/or practice and/or application knowledge in a relevant area of Information Technology, ICT, Computer Science, Software Engineering or equivalent.), proficiency with development tools (i.e. how many relevant development tools and technologies are used?), and expertise to estimate software projects (i.e., how many software projects are estimated?).

   Detailed description of expertise for each category of the team's member is given as follows:

   - **High Expertise**: Means a person has established sufficient theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform) and proficient with minimum 8-10 development tools / technologies (e.g.

Dot Net, Java, and others) and estimated or developed more than 10 number of software projects in agile and / or non-agile context (s).

- **Medium expertise** Means a person has established sufficient theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform) and proficient with minimum 4-7 development tools / technologies (e.g. Dot Net, Java, and others) and estimated or developed 5-10 software projects in agile and / or non-agile context:
- **Adequate expertise** Means a person has established necessary theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform) and proficient with minimum 1-3 development tools / technologies (e.g. Dot Net, Java, and others) and estimated or developed 1-4 software projects in agile and / or non-agile context:
- **Inadequate expertise** Means a person has established necessary theoretical and practical skills in Software systems development (desktop / mobile / web / cloud based platform):

Note: Please select an appropriate scale/rating against each level of the expertise by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-1. Ratings for Team member's expertise

Level of Expertise	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
High Expertise						
Medium Expertise						
Adequate Expertise						
Inadequate Expertise						

2. Team Member Experience

For this proposed study, experience means knowledge or skill that is acquired or practiced over a number of years. It is suggested to be measured across four levels (Proficient, Experienced, Intermediate, and Beginner) based on the team member's type and length (duration) of experience.

Type of experience includes knowledge and hands-on working in any one or more of the activities of a software project: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, maintenance, estimation and scheduling.

Detailed description of experience for each category of the team's member is given as follows:

- **Proficient:** Means a person has working experience of 10 years or more in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.

- **Experienced:** Means a person has working experience of 6-9 years in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.

- **Intermediate:** Means a person has working experience of 2-5 years in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.

- **Beginner:** Means a person has working experience of 0-1 years in estimations and scheduling of a software project and / or in other software development activities which includes but not limited to: requirements engineering, designing, architecting, code development, testing, quality assurance, system support, and maintenance.

Note: Please select an appropriate scale/rating against each level of the experience by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-2. Ratings for Team member's experience

Level of Experience	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Proficient						
Experienced						
Intermediate						
Beginner						

3. Team relationship and spirit

Agile development approaches are in use in 50% of software projects world-wide and therefore the effective teamwork of agile teams has much, and growing, significance and is of genuine interest to practitioners. For this proposed study, the slightly adapted BFTT (Big Five teamwork theory) framework as presented by Strode (2015) is suggested to be used to measure the effectiveness of teamwork. It will be measured across four levels (Excellent, Very Good, Good, and Conflicting) based on each of the concepts of BFTT as described in the following table:

According to the above given table, BFTT has given 08 concepts with their description to measure effectiveness of team. For this proposed study, these 08 concepts are treated as 08 traits of a team and they may be ordered from trait 1-8. The concept given in the first cell of the above table represents trait-1 of the team whereas second cell represents trait-2 of the team, and

Concepts	Description
Adaptability	Ability to adjust strategies based on information gathered from the environment through the use of backup behavior and reallocation of intra-team resources. Altering a course of action or team repertoire in response to changing conditions (internal or external).
Backup behavior	Ability to anticipate other team members' needs through accurate knowledge about their responsibilities. This includes the ability to shift workload among members to achieve balance during high periods of workload or pressure.
Closed-loop communication	The exchange of information between a sender and a receiver irrespective of the medium.
Mutual performance monitoring	The ability to develop common understandings of the team environment and apply appropriate task strategies to accurately monitor teammate performance.
Mutual trust	The shared belief that team members will perform their roles and protect the interests of their teammates.
Shared mental models	An organizing knowledge structure of the relationships among the task the team is engaged in and how the team members will interact.
Team leadership (relabeled Shared Team leadership in Strode (2015))	Ability to direct and coordinate the activities of other team members, assess team performance, assign tasks, develop team knowledge, skills, and abilities, motivate team members, plan and organize, and establish a positive atmosphere.
Team orientation	Tendency to take other's behavior into account during group interaction and the belief in the importance of team goal's over individual members' goals.

so on. Keeping in view these 08 traits the team effectiveness may be categorized into four levels which are explained as follows:

- **Excellent:** Means a team has all 08 traits as given in the above table without any deviations.
- **Very Good:** Means a team has all 08 traits as given in the above table with little deviations in trait 6 or 7.
- **Good:** Means a team has any 06 traits as given in the above table without any deviations in trait-1, 2, 4, 5, and 8.
- **Conflicting:** Means a team lacking any 3-4 traits as given in the above table or lacking trait-1, 2 and 5.

Note: Please select an appropriate scale/rating against each level of the team effectiveness by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-3. Ratings for Team effectiveness

Team effectiveness	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Excellent						
Very Good						
Good						
Conflicting						

4. Stakeholder / User collaboration

Agile practices lend importance to direct and regular contact between the clients and the development team. These type of contacts (e.g., Daily stand-up meetings, release & iterative planning) are said to be of help in building a positive relationship between client & team. More concretely, this relationship gives an opportunity to team to discuss user requirements and project progress with the client and to get the client's feedback and/or evaluate their success.

For this study, Stakeholder/User collaboration refers to the relationship between team & client and it can be measured across four levels (Fully, Partially, Occasionally, and Poorly) based on the type of contact (e.g. one to one, through email, Skype or other channels), frequency of visits/interactions, and the feedback level (e.g., detailed or incomplete) of the client stakeholders. Detail of each levels for stakeholder / user collaboration is given as follows:

- **Fully:** Stakeholder / user collaborates fully with the development team by attending all scheduled meetings and respects completely other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides a thorough feedback as and when required.

- **Partially:** Stakeholder / user collaborates partially with the development team by attending some of the scheduled meetings without missing the important ones e.g. (kick start, release planning, etc.) and respects completely other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides a thorough feedback as and when required.

- **Occasionally:** Stakeholder / user collaborates occasionally with the development team by attending some of the scheduled meetings but also disregards some of the important ones e.g. (kick-off meeting, release planning, etc.) and respects at times other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides delayed or in time thorough or incomplete feedback frequently or occasionally.

- **Poorly:** Stakeholder / user doesn't attend any of the scheduled meetings or attends some occasions but also disregards most of the important ones e.g. (kick-off meeting, release planning, etc.) and doesn't respect other agreed upon means of correspondence / communication of the intended project. Moreover, s/he provides a delayed or in time thorough and / or incomplete feedback frequently and / or occasionally.

Note: Please select an appropriate scale/rating against each level of the stakeholder/user collaboration by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-4. Ratings for stakeholder / user collaboration

Level of stakeholder /user collaboration	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Fully						
Partially						
Occasionally						
Poorly						

5. System context

This factor is intended to capture the complexities encountered due to the implementation of leading technologies and/or complex functionalities. For instance, evidence suggests that military and mission-critical systems are typically more complex than general purpose information-based systems, and hence require relatively more effort and cost to implement– all other things being equal. This aspect is subdivided into different types of systems (desktop / mobile / web/cloud based platform) to gauge their complexity level with respect to implementation:

a. Defence / Military / Mission Critical Systems
b. Embedded Systems
c. Decision Support / EIS / Enterprise Systems
d. Financial / Banking Systems (Mobile + Desktop)
e. General Purpose Systems other than financials (Web + Desktop + Mobile)
f. Other: if a type of system not covered in the above given types from a-e.

Note: Please select an appropriate scale/rating against each level of the complexity as anticipated by different types of systems with respect to their implementation by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-5. Ratings for complexity of types of system w.r.t. implementation

Type of Systems	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't
Defence/Military/Mission Critical Systems						
Embedded Systems						
DSS/EIS/Enterprise Systems						
Financial / Banking Systems (Mobile & Desktop)						
General Purpose Systems other than financials (Web, Desktop & Mobile)						
Other: a type of system not covered above						

6. Organization context

Organization structure and culture both affects the capability to accomplish projects success. Culture of an organization is a set of shared values, behaviours, and assumptions that describes its functional view and have 10 characteristics: member identity, group emphasis, people focus, unit integration, control, risk tolerance, reward criteria, conflict tolerance, means-ends orientation, and open-systems focus.

a) Organization Type
It seems plausible that organization type, including government, non-government, private, could influence a project directly or indirectly; e.g., in a government organization the strong bureaucracy may restrict frequent meetings with project stakeholders, or top management may be unwilling to support any specific project or team over concerns regarding probity.

Note: Please select an appropriate scale/rating to specify influence level(s) against types of the organization (e.g. government / public, private, Joint) by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-6.1. Ratings for influencing a project / team due to organization types

Type of Organization	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Government / Public sector (e.g. govt. dept. ministries, public limited company, corporations, etc.)						
Private sector (private limited companies, individual owned, etc.)						
Joint sector (public and private entrepreneur)						

b) Organization Culture

Organizations may have established values and beliefs that in turn can have positive or negative impacts on teams and their projects. Teams may seek to adhere to, or work against, strong organizational values.

Note: Please select an appropriate scale/rating against different organization's environment/culture by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-6.2. Ratings for influencing a project / team due to organization culture

Organization environment/culture	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Open & Robust (i.e. strong values and beliefs, team cohesiveness, mutual trust, freedom of expression, team autonomy, etc)						
Fragile or closed (i.e. weak values and beliefs, lack of mutual trust, no freedom of expression, etc)						

c) Work Environment

The work environment may have direct and indirect impacts on project outcomes; e.g., it can directly influence team performance, progression, and job continuity, while quality and development time can be impacted indirectly. This category is assessed in terms of the following:

- **Best Work environment**: It may have qualitative traits such as: people to operate as individuals, but still focus on succeeding as a team; favors honest feedback; keeping the team composed; offers employees flexibility to customize their own working styles; everything is questioned because questions are encouraged, etc.

- **Worst Work environment:** It may have qualities such as : strict timetables, dress codes, protocols, and operations; restrict the creativity and independence of team members; forces individuals into silos; nothing except success or failure; rewarding good behavior but still punishes bad behavior; bosses can't be asked, leaders can't be confronted, etc.

Note: Please select an appropriate scale/rating to specify influence level(s) against types of the work environment (i.e. quantitative like office desk, space, etc. and qualitative such as general feelings of discontent or enthusiasm that may be rippling through employees)) by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-6.3. Ratings for influencing a project / team due to work environment

Work environment's Types	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Best Work environment						
Worst Work environment						

7. **Sprint context**

It seems intuitive to expect that estimates and plans will be generally less accurate early in projects when information and certainty are at a premium, and that accuracy should improve as projects progress (all other things being equal). This aspect is also considered to be potentially important for the proposed model of estimation. It is assumed that the first sprint of any new project will need more effort than the succeeding sprints due to the longer time consumed by the team in initial learning and understanding of the user requirements and also in developing the first blue-print of the basic architecture of the system.

Note: Please select an appropriate scale/rating to indicate level of complexity as the project progresses sprint-wise (starting from sprint 1 to onward) by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-7. Ratings for complexity of a project progression sprint-wise (sprint 1, 2, 3,…, n)

Project progression sprint-wise	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Sprint 1						
Sprint 2						
Sprint n						

8. **Risk context**

Unmanaged risks is one of the (albeit numerous) reasons for the failure of software projects. Some of the risks linked with project failures are uncertainty regarding project complexity and project goals, and personality differences. This aspect also has the potential to impact on estimation. It is assumed that some risks are known and so they will be mitigated during the life of the project. However, some risks are unknown and therefore cannot be mitigated in advance, including the resignation of an active member of the team, long leave due to illness, and so on.

- Known (mitigated & not mitigated)
- Unknown so cannot be mitigated

Note: Please select an appropriate scale/rating to indicate influence of risks on a project success by putting in (✓) or (**x**) on the space as provided in the table below for positive or negative impact on the estimations if you agree with the views of the aforementioned factors regarding its significance:

Table-8. Ratings for influence of risks on a project success

Risk Types	**Scales / Ratings**					
	Very high	High	Moderate	Low	Very low	Don't know
Known mitigated risk						
Known non-mitigated risk						
Unknown risk						

Appendix-L (Ethics Approval from AUTEC)

Ethics Approval

Please do not staple your application

AUCKLAND UNIVERSITY OF TECHNOLOGY ETHICS COMMITTEE (AUTEC)

AUT

EA1

APPLICATION FOR ETHICS APPROVAL BY AUTEC

For AUTEC Secretariat Use only

___________/___________

Please print this application single sided in greyscale and do not staple. Once this application has been completed and signed, please read the notes at the end of the form for information about submission of the application for review.

NOTES ABOUT COMPLETION

- Ethics review is a community review of the ethical aspects of a research proposal. Responses should use clear everyday language with appropriate definitions being provided should the use of technical or academic jargon be necessary.
- The AUTEC Secretariat and your AUTEC Faculty Representative are able to provide you with assistance and guidance with the completion of this application which may help expedite the granting of ethics approval.
- The information in this application needs to be clearly stated and to contain sufficient details to enable AUTEC to make an informed decision about the ethical quality of the research. Responses that do not provide sufficient information may delay approval because further information will be sought. Overly long responses may also delay approval when unnecessary information hinders clarity. In general, each response should be around 100 words.
- AUTEC reserves the right not to consider applications that are incomplete or inadequate. Please do not alter the formatting or numbering of the form in any way or remove any of the help text.
- Comprehensive information about ethics approval and what may be required is available online at http://aut.ac.nz/researchethics
- The information provided in this application will be used for the purposes of granting ethics approval. It may also be provided to the Graduate Research School, the Research and Innovation Office, or the University's insurers for purposes relating to AUT's interests.
- The Form is focussed around AUTEC's ethical principles, which are in accordance with the *Guidelines for the approval of ethics committees* in New Zealand.

To respond to a question, please place your cursor in the space following the question and its notes and begin typing.

Project Information

What is the title of the research?

Effort Estimation and Project Scheduling in Agile Software Development

Is this application for research that is being undertaken in stages? ☒ Yes ☐ No

If the answer is 'Yes' please answer A.2.1 and the following sections, otherwise please answer A.3 and continue from there.

Does this application cover all the stages of the research? ☒ Yes ☐ No

If the answer is 'No' please provide details here of which stages are being covered by this application, otherwise please answer A.3 and continue from there.

Who is the applicant?

Professor Stephen MacDonell

Further information about the applicant.

In which faculty, directorate, or research centre is the applicant located?

School of Engineering, Computer & Mathematical Sciences, Faculty of Design & Creative Technologies

What are the applicant's qualifications?

1993, PhD, University of Cambridge, United Kingdom
1990, M.Com, University of Otago, New Zealand

1988, B.Com (Hons), University of Otago, New Zealand

What is the applicant's email address?

stephen.macdonell@aut.ac.nz

At which telephone numbers can the applicant be contacted during the day?

+64-9-921 9999 Ext.5811

Research Instruments

Which of the following does the research use:

☐ a written or electronic questionnaire or survey ☐ focus groups ☒ interviews
☒ observation ☐ participant observation ☐ ethnography ☐ photographs
☐ videos ☐ other visual recordings ☐ a creative, artistic, or design process
☐ performance tests
☐ some other research instrument (please specify)

Please attach to this application form all the relevant research protocols. These may include: Indicative questions (for interviews or focus groups); a copy of the finalised questionnaire or survey in the format that it will be presented to participants (for a written or electronic questionnaire or survey); a protocol indicating how the data will be recorded (e.g. audiotape, videotape, note-taking) for focus groups or interviews (Note: when focus groups are being recorded, you will need to make sure there is provision for explicit consent on the Consent Form and attach to this Application Form examples of indicative questions or the full focus group schedule. Please note that there are specific confidentiality issues associated with focus groups that need to be addressed); a copy of the observation protocol that will be used (for observations); full information about the use of visual recordings of any sort, including appropriate protocols and consent processes; protocols for any creative, artistic, or design process; a copy of the protocols for the instruments and the instruments that will be used to record results if you will use some other research instrument.

Who will be transcribing or recording the data?

If someone other than the applicant or primary researcher will be transcribing the interview or focus group records or taking the notes, you will need to provide a confidentiality agreement with this Application Form.

Please provide a brief plain English summary of the research (300 words maximum).

Organizations that develop software typically deploy considerable resources to drive innovation and transformation. However, enduring evidence indicates that many software projects fail to deliver within time or budget and do not achieve business objectives. One of the reasons underlying such 'failures' has been an inability to provide accurate forecasts of time and resources needed for project completion. Numerous estimation techniques have been employed in Traditional Software Development (TSD) to forecast project size, the human effort required, the likely developmental cost and the project schedule, but their effectiveness in Agile Software Development (ASD) is unknown. ASD is currently the most popular development approach and it brings with it comparatively simple forecasting methods; however these have several gaps and limitations. This research will seek to understand and improve the utility of forecasts of size, effort, cost and schedule for ASD projects through variants of approaches based on a range of indicators including but not limited to stories' size (estimates of task size in relative terms, often in unit-less story points), numbers of sprints (development cycles), teams' velocity (story points completed in a sprint by a team), experts' domain knowledge, and the organizational context. The proposed research will enable the determination of which single approach, or combination of multiple variants of approaches, could be expected to improve the utility of such forecasts for ASD projects.

Additional Research Information

Is this research an intervention study? ☒ Yes ☐ No

For research in general, what is the difference between intervention, interaction, and observation? Intervention includes both physical procedures by which data are gathered and manipulations of the participant or participant's environment that are performed for research purposes. Interaction includes communication or interpersonal contact between the investigator and participant that are performed for research purposes. Observation is neither an intervention nor an interaction. (cf https://www.gvsu.edu/hrrc/faq-definitions-35.htm).

Within health and disability research, 'intervention study' has the meaning given to it by the National Ethics Advisory Council's [Ethical Guidelines for Intervention Studies]; namely, a study in which the investigator controls and studies the intervention(s) provided to participants for the purpose of adding to knowledge of the health effects of the intervention(s). The term 'intervention study' is often used interchangeably with the terms 'experimental study' and 'clinical trial' (s.24 Standard Operating Procedures for Health and Disability Ethics Committees).

Is this Health and Disability Research? ☐ Yes ☒ No

Health and disability research is research that aims to generate knowledge for the purpose of improving health and independence outcomes (s.21 Standard Operating Procedures for Health and Disability Ethics Committees).

Does this research involve people in their capacity as consumers of health or disability support services, or in their capacity as relatives or caregivers of consumers of health or disability support services, or as volunteers in clinical trials (including, for the avoidance of doubt, bioequivalence and bioavailability studies)? ☐ Yes ☒ No

---

The Ethical Principle of Research Adequacy

AUTEC recognises that different research paradigms may inform the conception and design of projects. It adopts the following minimal criteria of adequacy: the project must have clear research goals; its design must make it possible to meet those goals; and the project should not be trivial but should potentially contribute to the advancement of knowledge to an extent that warrants any cost or risk to participants.

Is the applicant the person doing most of the research (the primary researcher)? ☐ Yes ☒ No

If the answer is 'No' please answer B.1.1 and the following sections, otherwise please answer B.2 and continue from there.

What is the name of the primary researcher if it is someone other than the applicant?

Muhammad Tariq

What are the primary researcher's completed qualifications?

2011 M.Sc. in Applied Information Technology from Goteborg University, Goteborg, Sweden

1996 M.Sc. in Computer Science from University of Agriculture, Faisalabad, Pakistan

1993 B.Sc. (Hons) Agri. Sciences from University of Agriculture, Faisalabad, Pakistan

What is the primary researcher's email address?

mtariq@aut.ac.nz

At which telephone numbers can the primary researcher be contacted during the day?

+64-9-921 9999 Ext 5852

Is the primary researcher

☐ an AUT staff member ***☒ an AUT student***

If the primary researcher is an AUT staff member, please answer B.2.1 and the following sections, otherwise please answer B.3 and continue from there.

In which faculty, directorate, or research centre is the primary researcher employed?

If the response to this section is the same as that already given to section A.4.1 above, please skip this section and go to section B.2.2.

In which school or department is the primary researcher employed?

When the primary researcher is a student:

What is their Student ID Number?

Student ID 1129307

In which faculty are they enrolled?

Faculty of Design & Creative Technologies

In which school, department, or Research Centre are they enrolled?

Department: School of Engineering, Computer and Mathematical Sciences (SECMS)

Research Centre: Software Engineering Research Lab (SERL)

What is the primary researcher's experience or expertise in this area of research?

The primary researcher has more than 17 years of professional experience in the field of Information Technology, out of which 9 years of experience relate to software project management / estimation and scheduling. Moreover, the area of research of the primary researcher's M.Sc. in applied IT is closely related to the domain of his PhD studies at AUT in which he undertook quasi-experiment to evaluate deviations in effort estimations based on Use Case Point (UCP) and UCP extended with Program Evaluation Review Technique (PERT). The objective of this research was to improve accuracy in effort estimation by applying PERT on technical & environmental factors of UCP equation/method.

Who is in charge of data collection?

Primary researcher

Who will interact with the participants?

The primary researcher and potentially the applicant

Is this research being undertaken as part of a qualification? ☒ Yes ☐ No

If the answer is 'Yes' please answer B.7.1 and the following sections, otherwise please answer B.8 and continue from there.

What is the name of the qualification?

Doctor of Philosophy (PhD) in Computer and Information Sciences

In which institution will the qualification be undertaken?

Auckland University of Technology

Details of Other Researchers or Investigators

Will any other people be involved as researchers, co- investigators, or supervisors? ☒ Yes ☐ No

If the answer is 'Yes' please answer B.8.1.1 and the following sections, otherwise please answer B.8.2 and continue from there.

What are the names of any other people involved as researchers, investigators, or supervisors?

Jim Buchan, Senior Lecturer

Where do they work?

School of Engineering, Computer and Mathematical Sciences (SECMS)

What will their roles be in the research?

Secondary Supervisor

What are their completed qualifications?

B.Sc., M.Sc. (Hons), Graduate Diploma (InfoSys), Diploma Teaching

Will any research organisation or other organisation be involved in the research? ☐ Yes ☒ No

If the answer is 'Yes' please answer B.8.2.1 and the following sections, otherwise please answer B.9 and continue from there.

What are the names of the organisations?

Not Applicable

Where are they located?

Not Applicable

What will their roles be in the research?

Not Applicable

Why are you doing this research and what is its aim and background?

Please provide the key outcomes or research questions and an academic rationale with sufficient information, including relevant references, to place the project in perspective and to allow the project's significance to be assessed.

Purpose: This research study seeks to understand and improve the utility of forecasts of size, effort, cost and schedule for ASD projects through variants of approaches based on a range of relevant objective and subjective indicators. Some of these indicators include but are not limited to: the size of stories (estimates of task size in relative terms, often in unit-less story points), the numbers of sprints (development cycles), the velocity/capacity (story points completed in a sprint) of the team, the domain knowledge of the team/experts, and the organisational context. Final selection of the relevant indicators is dependent upon stakeholder feedback and the results of our analyses.

Expected outcome: The proposed research will enable the determination of which single approach, or which combination of multiple variants of approaches, could be expected to improve the utility of forecasts of size, effort, cost and schedule for ASD projects.

Background: Organisations that develop software deploy considerable resources on projects in order to drive innovation and transformation (Sauser et al., 2009; Shenhar and Dvir, 2007). However, enduring evidence indicates that many software projects fail to deliver within time or budget and do not achieve business objectives (KPMG, 2017, PMI-Pulse, 2017).

Given its universal and pervasive nature, software is produced in various environments and under numerous situations (Prikladnicki et al., 2013). The development and use of software fall within a wider setting of 'socio-technical systems' where the human and organisational context play significant roles (Fuggetta & Di Nitto, 2014), along with other factors including budgets, infrastructure (e.g. offices, hardware, communications, travel, training) and others (Banker, 1987). The success of a software project is consequently multi-faceted – common considerations are adherence to scope, time, and cost goals and/or stakeholders' satisfaction (Schwalbe, 2014; Dragicevic et al., 2017). As such, estimates of the personnel effort likely to be required to complete a software project are central for forecasting size, cost or schedule, and these may need to be mediated by a wide range of potentially influential factors, including (Wegner & Ruhe, 2008):

f. product factors such as precedence, reliability, reuse
g. process factors such as architecture risk resolution, process maturity
h. development environment factors such as the use of particular practices or tools
i. human factors such as corporate culture, team culture, team capabilities and experience
j. physical factors such as the workplace, facilities.

Accurate estimation of effort has long been held to be important in planning a software development project and it remains an ongoing challenge in the field of software engineering (Dragicevic et al., 2017), yet in many instances software project managers fail to arrive at correct estimates (Tariq & Shobha, 2011). A review of studies has shown the disbursement of 30% to 40% more effort than estimated (Molokken-Ostvold & Jorgenson, 2003) and 20% to 30% average error on effort estimation (Grapenthin et al., 2014; Abrahamsson et al., 2011; Kang et al., 2010). Project bids and work schedules are developed based on some form of effort projection and the accuracy of that projection is significant in terms of organisations managing financial issues and monitoring a project's development and time-to-market activities (Suharjito et al., 2016). Poor estimation of software development effort can lead to detrimental outcomes such as schedule and budget overruns of the project, and could in some cases ultimately threaten organisational viability and competitiveness (Kocaguneli, 2012). A collaborative survey conducted by McKinsey and Oxford University of more than 54,000 software projects reported in 2013 that 45% of projects ran over budget, 7% fell behind schedule, and 56% provided less functionality than predicted. When these effects were considered overall, 17% of projects were said to threaten their organisations' existence (Bloch et al. 2013).

Numerous estimation techniques have been proposed for use in the software industry (Jørgenson & Shepperd, 2007; Sommerville, 2007), most of them formulated in the 1980s, and to varying degrees of success. Almost all have been founded on the prior estimation of the scale, and perhaps the complexity, of the project to be undertaken, based on some granular breakdown of the functionality to be delivered. While none has found universal applicability, some of these estimation methods have proved to be effective in specific development contexts. Particularly over the last decade, however, many software teams have moved from what we term here as traditional software development (TSD) (also known as planned or plan-driven, and typified by the waterfall method) to agile software development (ASD), due to the latter's emphasis on fast release cycles, frequent delivery of value-added features to customers, and close collaboration between the product owner and the development team (Hamdan et. al., 2013). Whether the estimation methods established prior to the advent of agile approaches have any relevance or value in ASD environments is still to be explored. The most widely known examples of sizing and estimation techniques used for projects undertaken in TSD environments are COCOMO (Briand et al., 1999; Zhao et al., 2003; Jodpimai et al., 2009), Function Point Analysis (Wu, 1997; Jun et al., 2008), Top-Down & Bottom-Up approaches (Pengelly, 1995; Wu, 1997), Use Case Points (UCP) Estimation (Diev, 2006; Ochodek et al., 2010; Collaris & Dekker, 2011), and expert estimation (Kumar et al., 2016), whereas the most common techniques used for sizing and forecasting per sprint in an ASD environment are Planning Poker, T-Shirt Sizing, and Relative Mass Valuation (Green, 2014).

Problem Statement: ASD represents a class of contemporary methods of software development. Although variants of agile methods have been used in multiple forms since the late 1950's, it is only relatively recently that agile software development has received global recognition and adoption (Hamdan et. al., 2013).

ASD has fundamentally changed the software life cycle. The emphasis of ASD is on the iterative and continuous delivery of working software features, replacing the formerly large-scale (and often document-heavy) phases of TSD including Vision, Analysis, Design, and Test. Moreover, just as ASD introduced the disruptive concept of agility in every step of the software life cycle, it has also ushered in a range of new and unique methods for sizing and forecasting. Rather than being based on a breakdown of the functionality to be delivered by the project to the customer, the breakdown in ASD typically centres on an increment of activity for the development team – in some ASD methods this increment is referred to as a sprint.

For instance, some executions of ASD include a group estimation methodology in which the team of developers sits with the product backlog manager and they estimate the relative size of each story, i.e., a feature of the software to be developed comprising a written description, conversations, and tests (Cohn, 2004), often in unit-less story points. The number of story points completed in a sprint is called the team's velocity. Over time, and assuming stability over a range of factors, the anticipated velocity of a team per sprint may be calculated based on actual development outcomes. Such data can inform those charged with the planning and delivery of features and releases in terms of management and scheduling.

As stated above, while in ASD a range of techniques are being utilized to forecast the story points that would make up a sprint, evidence suggests that the most popular are Planning Poker, T-Shirt Sizing, and Relative Mass Valuation (Green, 2014). The existing methodology of ASD to estimate story points and inform the scheduling of a software project is simple and can produce results that are close to actual values of size and effort for a project. However, the trade-off for that simplicity is that there are many gaps in the prevailing methodology, and their treatment may help to improve the estimations further:

k. Estimates of the work required/expected in a sprint are dependent on team experience in software development and/or effort estimation (Nguyen-Cong & De Tran-Cao, 2013), and on how effectively a product owner explains a story, making the process vulnerable to aspects of team composition and experience.

l. Offshore software development is increasingly common due to its access to cheaper labor costs (Islam, et al., 2009) whereas there is an assumption in ASD that a team meets frequently and relies on verbal rather than written material; this may compromise shared knowledge and hence may affect project schedules (Hamdan et. al., 2013). Geographical, temporal, and cultural differences are some of the global barriers which should be considered as additional cost and effort drivers while estimating effort for distributed ASD (Usman & Britto, 2016). In addition, extensive travel may be needed when a project is developed at different sites, and specialist tools and technologies may need to be employed, which may contribute to raise a further fraction of the project costs.

m. Existing estimation and tracking methods for agile software development cannot be easily related to time durations (relevant to scheduling) because story points are used to estimate the size and complexity of a story rather than the amount of work required, and its completion time is velocity-dependent where velocity may differ from team to team (Popli & Chuhan, 2014; Dragicevic et al., 2017).

n. In TSD senior project managers are frequently responsible for the estimation of size and effort in projects and in general they are not contributing substantially towards the overhead of the project because they are also involved in other management activities as well as project activities.

   In ASD, however, the team – which may consist of 6-8 members – is responsible for forecasts of all of the sprints in a project. While the specifics will vary it is not unreasonable to assume that a one-month sprint may require 8 time-boxed hours of the development team (Schwaber & Sutherland, 2016) to carry out the sizing and forecasting exercise. This is a non-trivial cost that, if it is to be expended on a regular basis, should be offset by the benefits of having the team involved. This can be further explained by an example: let us assume one team of eight members is working on a single agile software development project made up of four sprints. If the team utilizes 8 time-boxed hours to derive size estimates per sprint then they might need 32 hours exclusively for estimating sprints. The approximate cost over the project could be NZ$32,000 (based on the hourly rate of an experienced IT professional with multiple areas of expertise (Recruit IT, 2015).

o. The techniques used in agile forecasting of component size (e.g., user story size) are based on expert judgement which tends to be the most popular method applied (Jørgensen, 2005; Usman et al., 2014; Popli & Chuhan, 2014), probably due to its simplicity and flexibility. However, limited substantial empirical evidence is available to support this method over other methods of effort estimation (Jørgensen, 2005). Furthermore, a more recent study in a medium-sized organisation (focused on bespoke software development for domestic and international markets) showed cost estimation based on unstructured expert judgement to be unreliable during the bid phase of the project (Faria & Miranda, 2012).

p. Techniques including Function Point Analysis, UCP estimation, COCOMO and the like, which have been used successfully in TSD to estimate the size and effort of software projects, indicate that many factors can affect the cost of a project, including the desired reliability and usability of the software, team experience, team culture, corporate culture, workplace factors, use of modern tools/practices, offshore development, and others; how such factors are or could be accommodated in ASD sizing and estimation is not clear.

q. ASD embraces the concepts of self-management and self-organisation in a team. Such an approach assumes that all members are dutiful, responsible and honest in their dedication to a project. This is potentially an optimistic assumption. It is a matter of fact that some people are more capable, hardworking and dedicated compared to others, and/or more suited to particular roles and responsibilities. Furthermore, ASD supports teamwork more so than that of individuals. It is contended here that any compensation to the team members may have a good impact on the overall velocity of a team and may create a sense of positive competition among the team members. Consideration of such factors may influence the overall velocity of a team and have flow-on effects for project scheduling.

r. ASD suggests close collaboration between stakeholders and the Product Owner and user requirements are typically clarified through verbal communications (Marchesi & Succi, 2002). The same collaboration is also expected between the team and other stakeholders at the time of project estimation. Non-availability of stakeholders or their inadequate input may affect project estimates (Hamdan et. al., 2013).

By taking into consideration the issues just described it may be possible to improve the accuracy of ASD forecasts for size, effort, cost and schedule. This study is therefore designed to investigate how the above identified gaps and opportunities may be assessed and addressed in terms of their effect on ASD forecasts.

What are the potential benefits of this research to the participants, the researcher, and the wider community?

Agile software development (ASD) has quickly become a predominant development method recognised globally due to its objective of enabling fast release cycles of working software through the collaborative development of value-added features for customers. ASD management, however, is still an emergent phenomenon.

The proposed research will consider in depth the estimation approaches in ASD, the potential crossover concepts and factors from Traditional Software Development (TSD) methods, and the potential for ASD methods to be improved while retaining the flexibility and transparency that are so important in ASD. These advances will benefit the software engineering research community.

The research participants and their practitioner community will benefit through using optimal and 'accurate enough' cost estimations for contemporary software projects, one of the core objectives of this intended research.

The primary researcher will benefit by undertaking empirical work and theory development that will give him the opportunity to complete his PhD and to publish research papers. The applicant will also benefit from those publication opportunities, and from the primary researcher's PhD completion (as will the School, Faculty and University). These papers will contribute to the body of knowledge of those academics researching and teaching aspects of empirical software engineering.

What are the theoretical frameworks or methodological approaches being used?

The System Development Research Methodology, a multi-method research design framework that can accommodate (cycles of) quasi-experimentation, literature reviews, and interviews is planned. After confirming the factors relevant to estimation and scheduling in ASD via interviews the core empirical work in the study will be conducted via quasi-experimental methods conducted in the lab and in the field. The purpose of conducting a series of quasi-experiments is to systematically evaluate the impact of a range of forecasting techniques and their variants (Top-Down, Bottom-Up and PERT). This will enable the comparative analysis of existing methods of estimation along with the proposed new model of estimation. Models will be applied and calibrated on a number of projects using appropriate regression methods (linear, log-transformed, robust or trimmed), possibly along with fuzzy inference methods, and their performance assessed using best practice error measures – likely to include mean absolute error and Standardized Accuracy.

How will data be gathered and processed?

In this study multiple sources will be traversed to gather data in order to limit the effects of constrained interpretation due to the availability of a single data source. Data will be collected from various software development and management artefacts (both electronic and non-electronic), participant interviews and discussions, as well as quasi-experiments in the lab and in the field, which will in turn be subject to analysis.

The techniques used for data collection in this study fall under the three broad categories defined by (Lethbridge et al, 2005): a) Direct methods, real time data collected directly from subjects through interviews, meetings and the like; b) indirect methods, which involve collection of raw data without actually interacting with the subjects during the data collection, such as are available via software tools (company's project databank), or observations; and also by c) examining already available compiled or un-compiled data regarding projects (such as documented records of already completed ASD projects, archival data, repositories, logs).

The research methods will be employed with the support of various data collection instruments:

a. Individual in-depth interviews

Interviews will be conducted individually with industry practitioners to collect data relating to their perceptions of the factors that are thought to be relevant to estimation and scheduling. Audio-recording along with note taking techniques will be used to collect interview data. Daily Interpretive Analysis (DIA) will be utilized to assemble and interpret data as collected from the interviewees at the end of every day of interviewing. Suitably qualified and experienced participants will be recruited and they will be given pre-interview guidelines and an interview schedule one week prior to the planned meeting. The number of interviews will depend on the validation of the significant factors identified in the intended study and on the subsequent factors revealed by the respondents during the interviews.

Experimentation

Quasi-experimentation is planned. Projects will be used as objects of the experiment while teams who have completed or are working on ASD projects will be the subjects of the experiments. The estimation practices of around ten ASD teams will be observed by the primary researcher and notes taken as to how estimation is performed and the results obtained.

How will the data be analysed?

Data collected through the interviews will be transcribed, with audio data converted into written form. As per the requirements of the research, open and closed questions will be asked of the respondents. The closed questions will be evaluated for frequency of each response and median values. The open-ended questions will be coded, and depending on the volume of data recorded, this may be followed by thematic analysis, and by generalization from the themes regarding the phenomena in question.

The core empirical work in the research will be conducted via quasi-experimental methods. The purpose of conducting a series of experiments is to systematically evaluate the impact of a range of forecasting techniques and their variants (Top-Down, Bottom-Up and PERT). This will enable the comparative analysis of existing methods of estimation along with the proposed new model of estimation. Models will be applied and calibrated on a number of projects using appropriate regression methods (linear, log-transformed, robust or trimmed), and possibly fuzzy inference modelling, and their performance assessed using best practice error measures – likely to include mean absolute error and Standardized Accuracy. A block diagram depicting the experimental implementation is now shown:

Has any peer review taken place? **☒ Yes ☐ No**

If your answer is 'Yes', please specify and provide evidence e.g. a letter of confirmation.

☐ AUT Competitive Grant ***☐ External Competitive Research Grant***

☐ PGR1 ***☒ PGR2*** ***☒ PGR9*** ***☐ Independent Peer Review****

Optional exemplars for evidencing peer review are available from the Ministry of Health (HDEC) website (http://ethics.health.govt.nz/) or from the Forms section of the Research Ethics website (http://aut.ac.nz/researchethics)

General Project Details

Likely Research Output

What are the likely outputs of this research?

☒ a thesis ☐ a dissertation ☐ a research paper ☒ a journal article
☐ a book ☒ conference paper ☐ a documentary ☐ an exhibition
☐ a film ☐ some other artwork ☐ other academic publications or presentations
☐ Some other output, please specify

Research Location and Duration

In which countries and cities/localities will the data collection occur?

New Zealand: potentially Auckland, Wellington; Australia: TBC; Pakistan: Islamabad, Lahore, Karachi; Sweden: Goteborg, Stockholm; India: TBC;

Exactly where will any face to face data collection occur?

In the meeting rooms of software development companies.

In which countries and cities/localities will the data analysis occur?

New Zealand: Auckland

When is the data collection scheduled to commence?

May 2018

Research Participants

Who are the participants?

Personnel involved in the development activities of a software project, including requirements engineering, design, architecture, code development, testing, quality assurance, maintenance and management (in the roles of Product owners, Scrum masters, agile team members).

How many participants are being recruited for this research?

It is difficult to predict how many participants will be involved in interviews as this will depend on the number and diversity of personnel involved in software development and the availability of personnel to participate, as well as the diversity of their views. It is hoped that a minimum of 20 people will be involved in order to inform and validate the potential factors to be included in the proposed model. In the experimentation phase of the research up to ten agile teams (each generally comprising 6-8 members) will be interacting with the researcher via observation and interactive estimation using the methods described in B.13.

What criteria will be used to choose who to invite as participants?

For interviewing, recruitment will be conducted based on participants' overall experience, which should be not less than 3 years as a practitioner in software development. For experimentation, the recruitment will depend upon the availability of organisations' agile teams who are involved in the development of ongoing or new software products.

How will you select participants from those recruited if more people than you need for the study agree to participate?

We do not foresee this as being a problem as recruitment in this space tends to be challenging. However, if there are more people willing to participate in the interviews then preference will be given to those with more experience (in terms of years worked) in agile software development among those who agree to participate. That said, we will not explicitly exclude willing participants as long as they meet the minimum experience threshold, as they can be readily included in the interviews.

Will any people be excluded from participating in the study? ☒ Yes ☐ No

Exclusion criteria apply only to potential participants who meet the inclusion criteria. An exclusion criterion is any characteristic that ought to disqualify any potential participant from recruitment into the study. Consider

exclusion criteria when there are heightened risks due to power differences in the relationship, recent injury, or other characteristics that might place potential participants at unreasonable risk of harms.

If the answer to this question is 'Yes' please answer C.3.4.1 and the following sections, otherwise please answer C.3.5 and continue from there.

What criteria will be used to exclude people from the study?

Personnel with limited prior experience (less than 3 years) in software development will not be involved in the interviews.

Why is this exclusion necessary for this study?

We are seeking to confirm the factors that are relevant to estimation and scheduling in ASD – only those with sufficient experience (which we estimate as having been gained from a minimum of 3 years' work) will be in an informed position to contribute to the interviews.

Recruitment of participants.

Candidate organisations engaged in agile software development will be identified from public sources such as the Agile Auckland LinkedIn group and the indexNZ sector listing for Software Firms. The CIO of these organisations will be contacted by the applicant and primary researcher via email. The message will describe the nature and purpose of the research, the role of the researcher and our expectations of organisations and participants. Participants will be considered eligible to take part in interviews if they have knowledge and experience working for at least 3 years in an agile context in any of the activities of software development, such as requirements engineering, design, architecture, code development, testing, quality assurance, maintenance, and management. Further, organisations and teams sought to participate in the experimentation phase will be eligible only if they are using an agile methodology in the development of their software products and services. Organisations that respond positively to the request will be asked to confirm their participation in the research. Once confirmation is received the primary researcher will proceed. The primary researcher will ask the first contact of the targeted organisations to forward an email invitation to suitable participants who will then be asked to contact the researcher directly. In some cases the researcher may contact candidate organisations or research participants through phone calls or by face-to-face meetings.

How will the initial contact with potential participants occur?

Generally through email, with a follow-up phone call where necessary.

How will the contact details of potential participants be collected and by whom?

Contact details of software organisations that are engaged in agile software development will be identified from public sources such as the Agile Auckland LinkedIn group and the relevant indexNZ sector listing http://www.indexnz.com/Top/Computers-and-Internet/Software/Software-Firms/ by the primary researcher and applicant.

How will potential participants be invited to participate?

The applicant and primary researcher will contact the CIOs of software firms working in the area of agile development, initially through emails. The content of the initial message will describe the nature and purpose of research, the role of the researcher and our expectations from the participants. They will also contact the potential participants directly based on their social and professional links.

How much time will potential participants have to consider the invitation?

10 working days

How will potential participants respond to the invitation?

Through email or a phone call to the applicant and / or primary researcher.

How will potential participants give consent?

In writing, potentially conveyed/sent through email.

How and when will the inclusion criteria and exclusion criteria given in sections C.3.2 and C.3.3 be applied?

The email message sent by the applicant and the primary researcher to potential organisations and participants stating the purpose of the research and our expectations will also specify the associated inclusion/exclusion criteria. Moreover, these criteria will be reiterated prior to the conduct of the interviews and field experiments.

Will there be any follow up invitations for potential participants?

Yes, follow up invitations will be sent to the potential organisations / participants in case no response is received from them within 2 weeks.

Partnership, Participation and Protection

How does the design and practice of this research implement the principle of Partnership in the interaction between the researcher and other participants?

How will your research design and practice encourage a mutual respect and benefit and participant autonomy and ownership?

The interview phase of the research methodology relies on the input of suitably experienced domain experts (including product owners, scrum masters, quality managers and agile team members) for the validation of the previously identified factors (drawn from the literature review) and consideration of any additional factors considered relevant to the improvement of forecasts of size, effort, cost and schedule for ASD projects.

The experimentation phase of this research methodology depends on participants' contributions as agile teams to a series of experiments to systematically evaluate the impact of a range of forecasting techniques and their variants (Top-Down, Bottom-Up and PERT) along with the proposed model of estimation so that these can be compared with the organisations' existing method of estimation.

Participants are expected to work in teams and be willing to apply the researchers' suggested techniques on their on-going or new projects during the experimentation phase, to gain full benefit of the intended research. Any disagreement or reluctance within members of the teams regarding the use of the researchers' suggested techniques may be mitigated by replacing the participating teams with new teams.

The intent is that those involved in the above phases will mutually contribute to and benefit from this research, thus hopefully ensuring that each contributes and receives value for their involvement. The personnel play key roles in software development and so will be aware of the challenges they face in the forecasting of size, effort, cost and schedule for ASD projects. They will bring this knowledge to the interviews and experiments, to the benefit of the researchers. On the other hand, the researchers will bring a depth of understanding and prior knowledge of a range of solutions to overcome those challenges and will leverage this knowledge, and that obtained from the interview and experiments, to improve the practices of the participating organisations, teams and individuals.

How will you ensure that participants and researchers will act honourably and with good faith towards each other?

The methodology and research methods incorporate activities that should ensure that the participants are fully informed of the research process, its benefits, potential risks, and outcomes, which should help to secure mutual understanding and acceptance of the roles and responsibilities of all parties involved. The success of the data collection activities depends entirely on the good faith of those involved. Therefore, the researchers will show respect towards the participants – for instance, interviews will be scheduled at the participants' convenience. The researchers have prior experience in similar projects, so they are familiar with the challenges inherent to such research.

Are the outcomes designed to specifically benefit the participants and/or their social or cultural group?

As stated above, the design of the research will benefit the participant organisations indirectly through the support provided to the personnel involved.

How will the information and knowledge provided by the participants be acknowledged?

The thesis and research papers will acknowledge the input of the intended participants, generally in the Acknowledgements page/section and more specifically in informing the validation and empirical evaluation of the estimation factors and models.

How does the design and practice of this research implement the principle of Participation in the interaction between the researcher and other participants?

What is the actual role of participants in your research project? Will participants be asked to inform or influence the nature of the research, its aims, or its methodology? Will participants be involved in conducting the research or is their principal involvement one of sharing information or data? Do participants have a formal role as stakeholders e.g. as the funders and/or beneficiaries of the research? What role will participants have in the research outputs (e.g. will they be asked to approve transcripts or drafts)?

What is the actual role of participants in your research project?

Participants will be actively involved during both the interview and experimentation phases of the intended research. During the interviews product owners, scrum masters, and team members will be verbally encouraged by the researcher to comment on the relevance of the previously identified potential factors and to provide input regarding additional factors not yet considered in our study but which they believe are relevant to forecasts of size, effort, cost and schedule for ASD projects. During the experimentation phase the participant teams will undertake estimation as part of their normal work and will apply the suggested approaches and their variants (Top-Down, Bottom-Up and PERT) within their new and/or ongoing ASD projects, to enable the researcher to systematically evaluate their impact on forecasts of size, effort, cost and schedule for the ASD projects.

Participants are expected to work in teams and be willing to apply the researchers' suggested techniques on their on-going or new projects during the experimentation phase, to gain full benefit of the intended research. Any disagreement or reluctance within members of the teams regarding the use of the researchers' suggested techniques may be mitigated by replacing the participating teams with new teams.

Will participants be asked to inform or influence the nature of the research, its aims, or its methodology?

The participants will certainly influence the nature of the research in terms of the factors considered in the models, and they *may* impact the methodology in terms of the number of design cycles that will be needed in order to build, test and refine the proposed models and/or their variants.

Will participants be involved in conducting the research or is their principal involvement one of sharing information or data?

In the experimentation phase the participants will apply the suggested approaches to their new or on-going ASD projects, they will be asked for their views on the strengths and weaknesses of the suggested approaches, and to share their experiences on how to further refine and implement the suggested techniques in order to make them more effective.

Do participants have a formal role as stakeholders e.g. as the funders and/or beneficiaries of the research?

No – though as stated, the hope is that they will benefit indirectly from the research outcomes.

What role will participants have in the research outputs (e.g. will they be asked to approve transcripts or drafts)?

Participants will have no role in the research outputs, although they will be informed of the availability of the thesis when it is completed.

How does the design and practice of this research implement the principle of Protection in the interaction between the researcher and other participants?

How will you actively protect participants from deceit, harm and coercion through the design and practice of your research? How will the privacy of participants and researchers be protected? How will any power imbalances inherent in the relationships between the participants and researchers be managed? How will any cultural or other diversity be respected?

How will you actively protect participants from deceit, harm and coercion through the design and practice of your research?

The nature of the research will be communicated in full to the participants prior to their involvement, and will be reiterated to them at the time of the interviews and experimentation. The participants will be fully informed of their role in the research, and their consent will be sought prior to any involvement. The researchers acknowledge and respect the role of the participants as being crucial to robust and useful outcomes – deceit, harm and coercion work entirely against this acknowledgement.

How will the privacy of participants and researchers be protected?

The researcher only will know the identities of participants. In the published reports of this research (thesis and papers), participants' identities will not be revealed. No sensitive data will be published in the reports. The data and participant consent forms will be kept separately and securely for six years before being destroyed.

How will any power imbalances inherent in the relationships between the participants and researchers be managed?

Participation by the development personnel is voluntary, and the participants will determine the time and place of the interviews and experimentation. Only the primary researcher will attend the interviews and experiments. Moreover, the open-ended nature of the interview prompts should ensure that participants understand what is being asked of them but at the same time should not constrain their responses. Participants will be advised that they may decline to answer any questions or respond to any prompts if they find them inappropriate or of any concern.

How will any cultural or other diversity be respected?

Prior to the interviews and experimentation the researcher will contact each participant/organisation to ask if they have any particular preferences in regard to the conduct of the interview/experimentation e.g., that it take place in an accessible location. The researcher will respect any such requests wherever feasible.

Social and Cultural Sensitivity (including the obligations of the Treaty of Waitangi)

What familiarity does the researcher have with the social and cultural context of the participants?

The primary researcher is familiar with the social and cultural contexts of the proposed locations of this study, namely software development organisations in Pakistan, Sweden, and New Zealand. The primary researcher has lived in all three countries. Moreover, he has studied and worked in each location, in software development or research organisations, and so has social and cultural roots that are likely to coincide with those of the organisations and participants involved in this research. It is believed that this relevant professional and cultural familiarity will enable the primary researcher to conduct the research in an appropriate manner.

What consultation has occurred?

Research procedures should be appropriate to the participants. Researchers have a responsibility to inform themselves of, and take the steps necessary to respect the values, practices, and beliefs of the cultures and social groups of all participants. This usually requires consultation or discussion with appropriate people or groups to ensure that the language and research approaches being used are relevant and effective. Consultation should begin as early as possible when designing the project and should continue throughout its duration.

All researchers are encouraged to make themselves familiar with Te Ara Tika: Guidelines for Maori Research Ethics: A framework for researchers and ethics committee members which is able to be accessed through the Research Ethics website. Researchers may also find Te Kaahui Maangai a directory of Iwi and Maaori organisations to be helpful. This may be accessed via the Te Puni Kookiri website (http://www.tkm.govt.nz/). As well as these documents, the Health Research Council has published Pacific Health Research Guidelines, and Guidelines on research involving children. (see http://www.hrc.govt.nz). There are also guidelines by various organisations about researching with other populations that researchers will find helpful.

The primary researcher has conducted several cycles of discussions with his primary supervisor to seek guidance and gain clarity in the aforementioned area. Further, he also shared and discussed his draft ethics application with Dr Adam Taylor (Senior Research Lecturer in the Faculty of Design and Creative Technology's SECMS) for his valued feedback and advice. The primary researcher has also visited the above guidelines and links so as to gain an initial understanding of the aforementioned areas and issues.

With whom has the consultation occurred?

Please provide written evidence that the consultation has occurred.

Consultations have occurred with the primary supervisor and Dr Adam Taylor, the Senior Research Lecturer in SECMS, and Faculty representative on AUTEC.

How has this consultation affected the design and practice of this research?

Numerous refinements have been made to the plans and instruments as a result.

Does this research target Māori participants? ☐ Yes ☒ No

All researchers are encouraged to make themselves familiar with Te Ara Tika: Guidelines for Maori Research Ethics: A framework for researchers and ethics committee members

If your answer is 'No', please go to section E.4 and continue from there. If you answered 'Yes', please answer the next question.

Which iwi or hapu are involved?

Does this research target participants of particular cultures or social groups? ☒ Yes ☐ No

AUTEC defines the phrase 'specific cultures or social groups' broadly. In section 2.5 of Applying for Ethics Approval: Guidelines and Procedures it uses the examples of Chinese mothers and paraplegics. This is to identify their distinctiveness, the first as a cultural group, the second as a social group. Other examples of cultural groups may be Korean students, Samoan husbands, Cook Islanders etc., while other examples of social groups may be nurse aides, accountants, rugby players, rough sleepers (homeless people who sleep in public places) etc. Please refer to Section 2.5 of AUTEC's Applying for Ethics Approval: Guidelines and Procedures (accessible in the Ethics Knowledge Base online via http://www.aut.ac.nz/about/ethics) and to the relevant Frequently Asked Questions section in the Ethics Knowledge Base.

If your answer is 'No', please go to section E.5 and continue from there. If you answered 'Yes', please answer the next question.

Which cultures or social groups are involved?

Software Developers / IT Practitioners

Does this research focus on an area of research that involves Treaty obligations? ☐ Yes ☒ No

All researchers are encouraged to make themselves familiar with Te Ara Tika: Guidelines for Maori Research Ethics: A framework for researchers and ethics committee members.

If your answer is 'No', please go to section E.6 and continue from there. If you answered 'Yes', please answer the next question.

Which treaty obligations are involved?

Will the findings of this study be of particular interest to specific cultures or social groups? ☒ Yes ☐ No

If the answer is 'Yes' please answer E.6.1 and the following sections, otherwise please answer F.1 and continue from there.

To which iwi, hapū, culture or social groups will the findings be of interest?

Software Developers / IT Practitioners

How will the findings be made available to these groups?

A summary of findings will be distributed to participants and a copy will be made available for download on the SERL website.

Respect for the Vulnerability of Some Participants

"Vulnerable persons are those who are relatively (or absolutely) incapable of protecting their own interests. More formally, they may have insufficient power, intelligence, education, resources, strength, or other needed attributes to protect their own interests. Individuals whose willingness to volunteer in a research study may be unduly influenced by the expectation, whether justified or not, of benefits associated with participation, or of a retaliatory response from senior members of a hierarchy in case of refusal to participate may also be considered vulnerable." (Standards and Operational Guidance for Ethics Review of Health-Related Research with Human Participants, World Health Organisation).

Will your research involve any of the following groups of participants? ☐ Yes ☒ No

If your research involves any of these groups of participants, please clearly indicate which ones and then answer F.2 and the following section, otherwise please answer G.1 and continue from there.

☐ people unable to give informed consent? ☐ your (or your supervisor's) own students?

☐ preschool children? ☐ children aged between five and sixteen years?

☐ legal minors aged between sixteen and twenty years?

☐ People lacking the mental capacity for consent?

☐ people in a dependent situation (e.g. people with a disability, or residents of a hospital, nursing home or prison or patients highly dependent on medical care)?

☐ people who are vulnerable for some other reason (e.g. the elderly, persons who have suffered abuse, persons who are not competent in English, new immigrants)? – please specify

How is respect for the vulnerability of these participants reflected in the design and practice of your research?

What consultation has occurred to ensure that this will be effective?

Please provide evidence of the consultation that has occurred.

Informed and Voluntary Consent

How will information about the project be given to potential participants?

A copy of all information that will be given to prospective participants is to be attached to this Application Form. If written information is to be provided to participants, you are advised to use the Information Sheet exemplar. The language in which the information is provided is to be appropriate to the potential participants and translations need to be provided when necessary.

All participants will be provided with a copy of the Information Sheet associated with the research. This sets out the goals of the research as well as the intended data collection methods. Roles and expectations of those involved will also be described.

How will the consent of participants be obtained and evidenced?

AUTEC requires consent to be obtained and usually evidenced in writing. A copy of the Consent Form which will be used is to be attached to this application. If this will not be the case, please provide a justification for the alternative approach and details of the alternative consent process. Please note that consent must be obtained from any participant aged 16 years or older. Participants under 16 years of age are unable to give consent, which needs to be given by their parent or legal guardian. AUTEC requires that participants under the age of 16 assent to their participation. When the nature of the research requires it, AUTEC may also require that consent be sought from parents or legal guardians for participants aged between 16 and twenty years. For further information please refer to AUTEC's Applying for Ethics Approval: Guidelines and Procedures.

Participants will be asked to complete an AUTEC-approved Consent Form prior to the commencement of the research. These forms will be collected and stored (with the data storage privacy protocols as discussed in section H.1. and H.9. below) at AUT in a secure location in WT building level 7 at SERL or in WZ building when SERL will move from WT in future.

Will any of the participants have difficulty giving informed consent on their own behalf? ☐ Yes ☒ No

Please consider physical or mental condition, age, language, legal status, or other barriers.

If the answer is 'Yes' please answer G.3.1 and the following sections, otherwise please answer G.4 and continue from there.

If participants are not competent to give fully informed consent, who will consent on their behalf?

Researchers are advised that the circumstances in which consent is legally able to be given by a person on behalf of another are very constrained. Generally speaking, only parents or legal guardians may give consent on behalf of a legal minor and only a person with an enduring power of attorney may give consent on behalf of an adult who lacks capacity.

How will these participants be asked to provide assent to participation?

Whenever consent by another person is possible and legally acceptable, it is still necessary to take the wishes of the participant into account, taking into consideration any limitations they may have in understanding or communicating them.

Is there a need for translation or interpreting? ☐ Yes ☒ No

If your answer is 'Yes', please provide copies of any translations with this application and any Confidentiality Agreement required for translators or interpreters.

Respect for Rights of Privacy and Confidentiality

How will the privacy and confidentiality of participants be protected?

Please note that anonymity and confidentiality are different. For AUTEC's purposes, 'Anonymity' means that the researcher is unable to identify who the participant is in any given case. If the participants will be anonymous, please state how, otherwise, if the researcher will know who the participants are, please describe how the participants' privacy issues and the confidentiality of their information will be managed.

Privacy and confidentiality during the research will be respected and protected with diligence and by all available means. The following measures will be taken to protect the identity of the participants:

- Coding of data and removal of identifying material from any documentation
- Referring to participants by their roles rather than their names when collecting and storing information
- Non-sharing or discussion of acquired information/data with others apart from the primary researcher and the applicant
- Removing field notes from the field study sites and keeping them at a different place to avoid any accidental misuse of the information that may reveal participants' identities.

All information (including the identity of participants) will be kept confidential and secure from interception or appropriation by unauthorised persons, or for purposes other than the approved research. The confidentiality of information obtained incidentally during research will also be respected except where disclosure is necessary to avoid harm. The primary researcher and the applicant are aware of their responsibilities for the safekeeping and confidentiality of the signed consent forms and these will be stored separately from the data. Participants will have on-going rights to access any and all personal information related to them that is held by the researchers. Information will be used only for the purpose for which it was gathered as per the principles established in the Privacy Act 1993.

In summary, the researchers are aware of the possibility of inadvertent disclosure of identities and will remain vigilant in avoiding any such incidents, for example, when reflecting on research experiences and outcomes in any form of written or verbal discussion.

How will individuals or groups be identified in the final report?

If participants or groups will be identified, please state how this will happen, why, and how the participants will give consent.

In order to retain confidentiality, the names of research participants and their organisations will not be used or referred to in research outputs. Organisations will likely be labelled with letters (e.g., Organisation A, B etc.). Individual participants will be referred to using generic role names rather than organisation-specific role names.

What information on the participants will be obtained from third parties?

This includes use of third parties, such as employers or professional organisations, in recruitment.

Initial contact information will be sought from employers or line managers (most likely the CIO).

How will potential participants' contact details be obtained for the purposes of recruitment?

The applicant and primary researcher will contact the CIOs of software firms working in the area of agile development, initially through emails. The content of the initial message will describe the nature and purpose of research, the role of the researcher and our expectations from the participants. They will also contact potential participants directly based on their social and professional links. The candidate participants will then be contacted through email, with possible follow-up via a phone call, to finalise the recruitment process.

What identifiable information on the participants will be given to third parties?

None

Who will have access to the data during the data collection and analysis stages?

The applicant and the primary researcher.

Who will have access to the data after the findings have been produced?

The applicant and the primary researcher.

Are there any plans for the future use of the data beyond those already described? ☒ Yes ☐ No

The applicant's attention is drawn to the requirements of the Privacy Act 1993 (see Appendix I of AUTEC's Applying for Ethics Approval: Guidelines and Procedures). Information may only be used for the purpose for which it was collected so if there are plans for the future use of the data, then this needs to be explained in the Information Sheets for participants. If you have answered 'Yes' to this question, please answer section H.8.1.1 and continue from there. If you answered 'No' to this question, please go to section H.9 and proceed from there.

If data will be stored in a database, who will have access to that information, how will it be used, for what will it be used, and how have participants consented to this?

The primary researcher will have on-going access to the data. The data may be used for conducting subsequent research in the aforementioned area of research by the researcher and the applicant.

Will any contact details be stored for future use and if so, who will have access to them, how will they be used, for what will they be used, and how have participants consented to this?

The primary researcher and the applicant will have on-going access to contact details. This information may be used to recruit participants for future research which will also be communicated to participants in the current consent form. If this is done a separate application for ethical approval will be submitted to AUTEC.

Where will the data be stored once the analysis is complete?

Please provide the exact storage location. AUTEC normally requires that the data be stored securely on AUT premises in a location separate from the consent forms. Electronic data should be downloaded to an external storage device (e.g. an external hard drive, a memory stick etc.) and securely stored. If you are proposing an alternative arrangement, please explain why.

The final storage of data and consent forms will be on secure AUT servers and in the WT Building Level 7 at SERL in the specified secure storage facilities provided and maintained by the School of Engineering, Computer and Mathematical Sciences (or in a similar facility in WZ when SERL moves there in time). Paper documents will be stored at the same location in a locked cupboard along with digital material stored on CD/DVD/thumb drives. A copy of data will be kept separately in an archive CD/DVD/thumb drive in a secure location in the WT building level 7 at SERL (or in WZ as appropriate).

For how long will the data be stored after completion of analysis?

AUTEC normally requires that the data be stored securely for a minimum of six years, or ten years for health data. If you are proposing an alternative arrangement, please explain why.

For six years.

How will the data be destroyed?

If the data will not be destroyed, please explain why, identify how it will be safely maintained, and provide appropriate informed consent protocols.

Documents will be shredded and computer files will be deleted from storage to protect the confidentiality of participants. Electronic devices (hard disks, memory sticks etc.) will be re-formatted in a manner that means the data is beyond recovery.

Who will have access to the Consent Forms?

The primary researcher.

Where will the completed Consent Forms be stored?

Please provide the exact storage location. AUTEC normally requires that the Consent Forms be stored securely on AUT premises in a location separate from the data. If you are proposing an alternative arrangement, please explain why.

The Consent Forms will be stored securely in WT Building Level 7 at SERL in a locked cupboard (or in WZ as appropriate). The consent forms will be stored separately from the data.

For how long will the completed Consent Forms be stored?

AUTEC normally requires that the Consent Forms be stored securely for a minimum of six years, or ten years in the case of research involving health data. If you are proposing an alternative arrangement, please explain why.

For six years.

How will the Consent Forms be destroyed?

If the Consent Forms will not be destroyed, please explain why.

Hard copies will be shredded and electronic devices (hard disks, memory sticks etc.) will be re-formatted in a manner that means the forms are beyond recovery.

Does your research involve the collection of personally identifiable and sensitive data? ☐ Yes ☒ No

Sensitive data can be used to identify an individual, object or location and has a risk of discrimination, harm or unwanted attention. Sensitive data potentially poses a substantial threat to those who are or who have been involved in it, especially if it is shared inappropriately, or if it falls into the wrong hands. If you have answered 'Yes' please identify what data is being collected and how it is sensitive and provide a Data Safety Management Protocol (see the Forms section of the Research Ethics website for a guide to drafting one). If the answer is 'No', please answer H.13 and continue from there.

Does your project involve the use of previously collected information or biological samples for which there was no explicit consent for this research? ☐ Yes ☒ No

If the answer is 'Yes' please answer H.13.1 and the following sections, otherwise please answer H.14 and continue from there.

What previously collected data will be involved?

Who collected the data originally?

Why was the information originally collected?

For what purposes was consent originally given when the information was collected?

How will the data be accessed?

Does your project involve any research about organisational practices where information of a personal or sensitive nature may be collected and / or where participants may be identified? ☒ **Yes** ☐ **No**

If the answer is 'Yes' please answer H.14.1 and the following sections, otherwise please answer I.1 and continue from there.

How will organisational permission be obtained and recorded?

The organisations will be made fully aware of the nature of the research, its aims and objectives, and the type of information required both about the organisation itself as well as the participants. As stated above, a project Information Sheet describing the details of the research will be provided when they are asked to consider participation in the research. Each participant will be asked to sign a Consent Form if they agree to take part.

Will the organisation know who the participants are?

The organisations will share our initial emails with their employees who later may contact the primary researcher directly if they are willing to participate in the intended research. Having said that, in small organisations it may not be possible to keep participants' involvement completely private. With multiple personnel from each organisation involved, however, specifics of the information provided by an individual should not be able to be traced back to them.

How will the identity of the participants be kept confidential?

All information (including the identity of participants) will be kept confidential and secure from interception or appropriation by unauthorised persons, or for purposes other than the approved research. This will be done by the coding of data and removal of identifying material from documentation. Furthermore, the participants will be addressed by their generic roles rather than individual roles to maintain confidentiality.

Minimisation of risk

Risks to Participants

Please consider the possibility of moral, physical, psychological or emotional risks to participants, including issues of confidentiality and privacy, from the perspective of the participants, and not only from the perspective of someone familiar with the subject matter and research practices involved. Please clearly state what is likely to be an issue, how probable it is, and how this will be minimised or mitigated (e.g. participants do not need to answer a question that they find embarrassing, or they may terminate an interview, or there may be a qualified counsellor present in the interview, or the findings will be reported in a way that ensures that participants cannot be individually identified, etc.) Possible risks and their mitigation should be fully described in the Information Sheets for participants.

How much time will participants be required to give to the project?

For the purpose of the interviews participants will be asked to take part in a one-hour interview/discussion.

For the purpose of the experimentation participants will be applying the suggested techniques along with their normal estimation practices while the primary researcher takes notes of his observations and contributes to any discussion as requested by the participants. During this phase the number and length of the interactions between researcher and participants will vary, depending on the duration and status of the ASD project and participant availability. That said, we would expect that interactions occurring over approximately 2-3 hours per week, over at least a four week period, for each ASD project being analysed, will be sufficient to enable adequate data collection, analysis and feedback.

What level of discomfort or embarrassment may participants be likely to experience?

Low or none

In what ways might participants be at risk in this research?

- The participants may feel an obligation to participate.
- The participants may express views that do not meet with the approval of the organisation.
- The participating organisation may be worried about commercial sensitivity.
- The participants may be concerned that their processes, practices and methods will be criticised.

Reassuring participants regarding their voluntary participation and the provisions for confidentiality will mitigate the above stated risks, as will the researcher's reiteration that the information gathered from their views will not be discussed with others. They will also be reassured that it is 'how things are done' in practice that is of interest, and no judgement is made whether these practices are somehow 'right' or 'wrong'. The participating organisations will be guaranteed that their commercial sensitivity will be safe-guarded as rigorously as possible and that their organisational identities will not be disclosed in the publication of results from this research.

In what ways are the participants likely to experience risk or discomfort as a result of cultural, employment, financial or similar pressures?

- The participants may feel uncomfortable criticising or providing negative evaluation of their existing practices or processes.
- The participants may fear that their opinions expressed in the interviews are not confidential.
- The participants may feel that their personal performance is being monitored.

Will your project involve processes that are potentially disadvantageous to a person or group, such as the collection of information, images etc. which may expose that person/group to discrimination, criticism, or loss of privacy? ☐ Yes ☒ No

If your answer is 'Yes', please detail how these risks will be managed and how participants will be informed about them.

Will your research involve collection of information about illegal behaviour(s) which could place the participants at current or future risk of criminal or civil liability or be damaging to their financial standing, employability, professional or personal relationships? ☐ Yes ☒ No

If your answer is 'Yes', please detail how these risks will be managed and how participants will be informed about them.

If the participants are likely to experience any significant discomfort, embarrassment, incapacity, or psychological disturbance, please state what consideration you have given to the provision of counselling or post-interview support, at no cost to the participants, should it be required.

Adult research participants in Auckland are able to utilise counselling support from the AUT Counselling Team, otherwise you may have to consider local providers for participants who are located nationwide, or in some particular geographical area or who are children. You may discuss the potential for participant psychological impact or harm with the Head of AUT Counselling, if you require. Please check the relevant Frequently Asked Question on the research ethics website as well and ensure the appropriate wording in included in the Information Sheet when counselling opportunities need to be offered.

Not Applicable

Will any use of human remains, tissue or body fluids which does not require submission to a Health and Disability Ethics Committee occur in the research? ☐ Yes ☒ No

e.g. finger pricks, urine samples, etc. (please refer to section 13 of AUTEC's Applying for Ethics Approval: Guidelines and Procedures). If your answer is yes, please provide full details of all arrangements, including details of agreements for treatment, how participants will be able to request return of their samples in accordance with right 7 (9) of the Code of Health and Disability Services Consumers' Rights, etc.

Will this research involve potentially hazardous substances? ☐ Yes ☒ No

e.g. radioactive material, biological substances (please refer to section 15 of AUTEC's Applying for Ethics Approval: Guidelines and Procedures and the Hazardous Substances and New Organisms Act 1996).

If the answer is 'Yes', please provide full details, including hazardous substance management plan.

Risks to Researchers

If this project will involve interviewing participants in private homes, undertaking research overseas, in unfamiliar cultural contexts, or going into similarly vulnerable situations, then a Researcher Safety protocol

should be designed and appended to this application. This should identify simple and effective processes for keeping someone informed of the researcher's whereabouts and provide for appropriate levels of assistance.

Are the researchers likely to be at risk? ☐ Yes ☒ No

If the answer is 'Yes' please answer I.2.1.1 and then continue, otherwise please answer I.3 and continue from there.

In what ways might the researchers be at risk and how will this be managed?

Risks to AUT

Is AUT or its reputation likely to be at risk because of this research? ☐ Yes ☒ No

If the answer is 'Yes' please answer I.3.1.1 and then continue, otherwise please answer I.3.2 and continue from there.

In what ways might AUT be at risk in this research?

Please identify how and detail the processes that will be put in place to minimise any harm.

Are AUT staff and/or students likely to encounter physical hazards during this project? ☐ Yes ☒ No

If yes, please provide a hazard management protocol identifying how harm from these hazards will be eliminated or minimised.

Truthfulness and limitation of deception

How will feedback on or a summary of the research findings be disseminated to participants (individuals or groups)?

Please ensure that this information is included in the Information Sheet.

The results of the research will be offered to the participants as they are produced during the research in the form of scholarly articles (conference or journal papers) and the thesis itself.

Does your research include any deception of the participants, such as non-disclosure of aims or use of control groups, concealment, or covert observations? ☐ Yes ☒ No

Deception of participants in research may involve deception, concealment or covert observation. Deception of participants conflicts with the principle of informed consent, but in some areas of research it may sometimes be justified to withhold information about the purposes and procedures of the research. Researchers must make clear the precise nature and extent of any deception and why it is thought necessary. Emphasis on the need for consent does not mean that covert research can never be approved. Any departure from the standard of properly informed consent must be acceptable when measured against possible benefit to the participants and the importance of the knowledge to be gained as a result of the project or teaching session. This must be addressed in all applications. Please refer to Section 2.4 of AUTEC's Applying for Ethics Approval: Guidelines and Procedures when considering this question.

If the answer is 'Yes' please answer J.2.1 and the following sections, otherwise please answer J.3 and continue from there.

Is deception involved?

Why is this deception necessary?

How will disclosure and informed consent be managed?

Will this research involve use of a control group? ☐ Yes ☒ No o

If the answer is 'Yes' please answer J.3.1 and the following sections, otherwise please answer K.1 and continue from there.

How will the Control Group be managed?

What percentage of participants will be involved in the control group?

What information about the use of a control group will be given to the participants and when?

Avoidance of Conflict of Interest

Researchers have a responsibility to ensure that any conflict between their responsibilities as a researcher and other duties or responsibilities they have towards participants or others is adequately managed. For example, academic staff members who propose to involve their students as participants in research need to ensure that no conflict arises between their roles as teacher and researcher, particularly in view of the dependent relationship between student and teacher, and of the need to preserve integrity in assessment processes. Likewise researchers have a responsibility to ensure that any conflict of interest between participants is adequately managed for example, managers participating in the same research as their staff.

While none is anticipated, the researchers will continually seek to identify any potential conflicts of interest and determine suitable measures to deal with any that arise. No sponsorship, funding or commercial support of the research will be accepted so as to ensure that this does not compromise the adequacy or ethical integrity of the research.

What conflicts of interest are likely to arise as a consequence of the researchers' professional, social, financial, or cultural relationships?

No such conflicts of interest are expected to arise as a consequence of this research engagement.

What possibly coercive influences or power imbalances are there in the professional, social, financial, or cultural relationships between the researchers and the participants or between participants (e.g. dependent relationships such as teacher/student; parent/child; employer/employee; pastor/congregation etc.)?

The only power imbalance that can be foreseen is the professional relationship between employee and employer in the participating organisations. This is addressed in sub para K.3 below.

How will these conflicts of interest, coercive influences or power imbalances be managed through the research's design and practice and how will any adverse effects that may arise from them be mitigated?

- The interviews will be conducted in a meeting room. The risks will be mitigated by maintaining confidentiality over any information that may reveal a participant's identity or their views on practices and processes.
- During field experiments participants are expected to work in teams and be willing to apply the researchers' suggested techniques on their on-going or new projects during the experimentation phase, to gain full benefit of the intended research. Any disagreement or reluctance within members of the teams regarding the use of the researchers' suggested techniques may be mitigated by replacing the participating teams with new teams.
- Participants will be addressed by generic roles rather than their individual titles or roles.
- The interview data collected from participants will not be discussed with other participants.
- Field notes will be taken off site at regular intervals and kept private.
- The participants will be reassured that their performance is not being monitored. The research is about what kinds of information is important in software estimation/forecasting.
- The participants will be reassured that it is how things are done in practice that are of interest, and no judgement is being made whether they are somehow 'right' or 'wrong'.

Does your project involve payments or other financial inducements (including koha, reasonable contribution towards travel expenses or time, or entry into a modest prize draw) to participants? ☐ Yes ☒ No

If the answer is 'Yes' please answer K.4.1 and the following sections, otherwise please answer K.5 and continue from there.

What form will the payment, inducement, or koha take?

Of what value will any payment, gift or koha be?

Will potential participants be informed about any payment, gift or koha as part of the recruitment process, and if so, why and how?

Have any applications for financial support for this project been (or will be) made to a source external to ☐ Yes ☒ No

If the answer is 'Yes' please answer K.5.1 and the following sections, otherwise please answer K.6 and continue from there.

What financial support for this project is being provided (or will be provided) by a source external to AUT?

Who is the external funder?

What is the amount of financial support involved?

How is/are the funder/s involved in the design and management of the research?

Have any applications been (or will be) submitted to an AUT Faculty Research Grants Committee or other AUT funding entity? ☐ Yes ☒ No

If the answer is 'Yes' please answer K.6.1 and the following sections, otherwise please answer K.7 and continue from there.

What financial support for this project is being provided (or will be provided) by an AUT Faculty Research Grants Committee or other AUT funding entity?

What is the amount of financial support involved?

How is/are the funder/s involved in the design and management of the research?

Is funding already available, or is it awaiting decision?

Do the applicant or the researchers, investigators or research organisations mentioned in Part B of this application have any financial interests in the outcome of this project? ☐ Yes ☒ No

If the response is 'Yes', please provide full details about the financial interests and how any conflicts of interest are being managed, otherwise, please respond to section K.9 and continue from there.

Are the participants expected to pay in any way for any services associated with this research? ☐ Yes ☒ No

If the response is 'Yes', please provide full details about the charges and describe how any benefits will balance the burdens involved as well as how any conflicts of interest are being managed. Otherwise please respond to section L.1 and continue from there.

Respect for Property

Researchers must ensure that processes do not violate or infringe legal or culturally determined property rights. These may include factors such as land and goods, works of art and craft, spiritual treasures and information.

Will this research impact upon property owned by someone other than the researcher? ☐ **Yes ☒ Noo**

If the answer is 'Yes' please answer L.1.1 and the following sections, otherwise please answer L.2 and continue from there.

How will this be managed?

How do contexts to which copyright or Intellectual Property apply (e.g. research instruments, social media, virtual worlds etc.) affect this research and how will this be managed?

Particular attention should be paid to the legal and ethical dimensions of intellectual property. Care must be taken to acknowledge and reference the ideas of all contributors and others and to obtain any necessary permissions to use the intellectual property of others. Teachers and researchers are referred to AUT's Intellectual Property Policy for further guidance.

Not Applicable

Checklist

Please ensure all applicable sections of this form have been completed and all appropriate documentation is attached as incomplete applications will not be considered by AUTEC.

Have you discussed this application with your AUTEC Faculty Representative, the Executive Secretary, or the Ethics Coordinator?	☐ Yes ☒ No
Is this application related to an earlier ethics application? If yes, please provide the application number of the earlier application.	☐ Yes ☒ No
Are you seeking ethics approval from another ethics committee for this research? If yes, please identify the other committee.	☐ Yes ☒ No

Section A	***Project information provided***	☒
Section B	***Research Adequacy information provided***	☒
Section C	***Project details provided***	☒
Section D	***Three Principles information provided***	☒
Section E	***Social and Cultural Sensitivity information provided***	☒
Section F	***Vulnerability information provided***	☒
Section G	***Consent information provided***	☒
Section H	***Privacy information provided***	☒
Section I	***Risk information provided***	☒
Section J	***Truthfulness information provided***	☒
Section K	***Conflict of Interest information provided***	☒
Section L	***Respect for Property information provided***	☒
Section M	***References provided***	☒
Section N	***Checklists completed***	☒
Section O.1 and 2	***Applicant and student declarations signed and dated***	☒
Section O.3	***Authorising signature provided***	☒

Spelling and Grammar Check (please note that a high standard of spelling and grammar is required in documents that are issued with AUTEC approval)

Attached Documents (where applicable)

Participant Information Sheet(s)	☒
Consent Form(s)	☒
Questionnaire(s)	☐
Indicative Questions for Interviews or Focus Groups	☒
Observation Protocols	☐
Recording Protocols for Tests	☐
Advertisement(s)	☐
Researcher Safety Protocol	☐
Hazardous Substance Management Plan	☐
Any Confidentiality Agreement(s)	☐
Any translations that are needed	☐
Other Documentation	☒

Declarations

Declaration by Applicant

Please tick the boxes below.

☒ ***The information in this application is complete and accurate to the best of my knowledge and belief. I take full responsibility for it.***

☒ ***In conducting this study, I agree to abide by all applicable laws and regulations, and established ethical standards contained in AUTEC's Applying for Ethics Approval: Guidelines and Procedures and internationally recognised codes of ethics.***

☒ ***I will continue to comply with AUTEC's Applying for Ethics Approval: Guidelines and Procedures, including its requirements for the submission of annual progress reports, amendments to the research protocols before they are used, and completion reports.***

☒ ***I understand that brief details of this application may be made publicly available and may also be provided to the Graduate Research School, the Research and Innovation Office, or the University's insurers for purposes relating to AUT's interests.***

09/04/2018

Signature ***Date***

Declaration by Student Researcher

Please tick the boxes below.

☒ ***The information in this application is complete and accurate to the best of my knowledge and belief.***

☒ ***In conducting this study, I agree to abide by all applicable laws and regulations, and established ethical standards contained in AUTEC's Applying for Ethics Approval: Guidelines and Procedures and internationally recognised codes of ethics.***

☒ ***I will continue to comply with AUTEC's Applying for Ethics Approval: Guidelines and Procedures, including its requirements for the submission of annual progress reports, amendments to the research protocols before they are used, and completion reports.***

☒ ***I understand that brief details of this application may be made publicly available and may also be provided to the Graduate Research School, the Research and Innovation Office, or the University's insurers for purposes relating to AUT's interests.***

09/04/2018

Signature ***Date***

Authorisation by Head of Faculty/School/Programme/Centre

Please tick the boxes below.

☐ ***The information in this application is complete and accurate to the best of my knowledge and belief.***

☐ ***In authorising this study, I declare that the applicant is adequately qualified to undertake or supervise this research and that to the best of my knowledge and belief adequate resources are available for this research and all appropriate local research governance issues have been addressed.***

☐ ***I understand that brief details of this application may be made publicly available and may also be provided to the Graduate Research School, the Research and Innovation Office, or the University's insurers for purposes relating to AUT's interests.***

Signature ***Date***

Notes for submitting the completed application for review by AUTEC

- Please ensure that you are using the current version of this form before submitting your application.
- Please ensure that all questions on the form have been answered and that no part of the form has been deleted.
- Please provide ***one*** printed, single sided, A4, and signed copy of the application and all related documents.

- Please deliver or post to the AUTEC Secretariat, room WU406, fourth floor, WU Building, City Campus. The internal mail code is D-88. The courier address is 46 Wakefield Street, Auckland 1010. Alternatively, please hand the application to the Research Ethics Advisor in person at one of the Drop In sessions at any of the four campuses (http://www.aut.ac.nz/researchethics/resources/workshops-and-drop-inns).
- Applications should be submitted once they have been finalised. For a particular meeting it needs to have been received in the AUTEC Secretariat by midday on the relevant agenda closing day [AUTEC's meeting dates are listed in the website at http://www.aut.ac.nz/researchethics]
- If sending applications by internal mail, please post them at least two days earlier to allow for any delay that may occur.
- Late applications will be placed on the agenda for the following meeting.

MINIMAL RISK CHECKLIST

Your application may be appropriate for an expedited review if it poses no more than minimal risk of harm to participants. To assist AUTEC's Secretariat to screen the application for assignment to the correct review pathway, please complete the following checklist:

Does the research involve any of the following?

ANONYMOUS SURVEY ASSESSMENT

		Yes	No
1	The collection of anonymous and non-sensitive survey/questionnaire data only. *(If YES is checked, the application may receive an expedited review if the data is from adults and poses no foreseeable risks to participants OR where any foreseeable risk is no more than inconvenience – no further questions on this checklist need be answered.)*	☐	☒

MINIMAL RISK ASSESSMENT[1]

		Yes	No
2	Participants who are unable to give informed consent (including children under 16 years old), or who are particularly vulnerable or in a dependent situation, (e.g. people with learning difficulties, over-researched groups, people in care facilities, or patients highly dependent on medical care)?	☐	☒
3	A reasonable expectation of causing participants physical pain beyond mild discomfort, or that experienced by the participants on an every-day basis, or any emotional discomfort, embarrassment, or psychological or spiritual harm, (e.g. asking participants to recall upsetting events)?	☐	☒
4	Research processes which may elicit information about any participant's involvement in illegal activities, or activities that represent a risk to themselves or others, (e.g. drug use or professional misconduct)?	☐	☒
5	Collection of any human tissue, blood or other samples, or invasive or intrusive physical examination or testing?	☐	☒
6	The administration of any drugs, medicines, supplements, placebo or non-food substances?	☐	☒
7	An intervention of any form of exercise, or other physical regime that is different to the participants' normal activities (e.g. dietary, sleep)?	☐	☒
8	Participants who are being asked to give information of a personal nature about their colleagues, employers, teachers, or coaches (or any other person who is in a power relationship with them), and where the identity of participants or their organisation may be inferred?	☐	☒
9	Any situation which may put the researcher at risk of harm? (E.g. gathering data in private homes)?	☐	☒
10	The use of previously collected biological samples or identifiable personal information for which there was no explicit consent for this research?	☐	☒
11	Any matters of commercially sensitive information?	☒	☐
12	Any financial interest in the outcome of the research by any member(s) of the research team?	☐	☒
13	People who are not giving consent to be part of the study, or the use of any deception, concealment or covert observations in non-public places, including social media?	☐	☒
14	Participants who are in a dependent or unequal relationship with any member(s) of the research team (e.g. where the researcher is a lecturer/ teacher/ health care provider/ coach/ employer/ manager/ or relative etc.) of any of the participants?	☐	☒

[1] If "No" is checked to all items 2-14, the application's status as Minimal Risk will be checked by the Secretariat, and may be forwarded to expedited review. Applications with more than Minimal Risk (any one "yes" to questions 2-14 above), and applications where the checklist is not completed will appear on AUTEC's next agenda.