DATA AND PROCESS MODELLING: INVESTIGATING THE GAP BETWEEN EDUCATION AND INDUSTRY EXPECTATIONS IN NEW ZEALAND

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF PHILOSOPHY

Supervisors

Dr. Mali Senapathi

Shoba Tegginmath & Dr. Stephen Thorpe

June 26, 2017

By

Vandana Taneja

School of Engineering, Computer and Mathematical Sciences

Copyright

Declaration

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 nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

Acknowledgements

When it's time to thank people it becomes a challenge. This is because there are so many people who give their enormous support, and a researcher does not want to lose an opportunity to show his/her gratitude. Well, it always starts with family, without whose support there is no thought of taking the initial step, and then there are the supervisors who nurture a researcher:

- Vipul and Simmi, thank you truly for your stupendous support. You two are incredible and I am very proud of you both. Thank you heaps for the little things that made this journey easier, your humour, and your unconditional love and support and co-operation and so much, without which I would not have written this important page of my thesis. Thank you, my children, this is the outcome of your sacrifices and unending support.
- My primary supervisor, Mali Senapathi - Mali-it has been an amazing journey. Like any naive researcher, at times I was in high spirits and the next moment I was so dejected. However, something kept me motivated and that was you. You encouraged me, and helped me to improve many of the skills which were necessary for a quality research. Thank you for your critical input, and most importantly, for not spoon feeding my learning.

  Thank you to my co-supervisors:

- Shoba, my co-supervisor- thank you for your feedback, proof reading (I will never forget your comments that 'statements do not talk').
- Stephen, my co-supervisor and my friend - thank you for supporting me and those encouraging sessions over coffees, breakfasts and quoting experiences and sharing memories from your own journey during the beginning of my journey. Now when I look back I see how the role of supervisor of undergraduate projects helped me to critically analyse student inside me and understanding the role of a supervisor at the same time.

- Kirstie Thorpe, thank you for taking care of all the 'a/an/the' by proof reading my thesis. Any one can write but never knew writing academic English could be such a challenge. If it were not you, this thesis would not have shaped up so well. Many thanks to you.

My special thanks go to all the participants for their input and all those people who have introduced me to them - If it weren't for you, this research, unique in its kind, would not have shaped up so well. I am also thankful to all who helped me in running a pilot questionnaire. Without your input, questions would have been ambiguous.

- Thank you Prof. Steve MacDonell for your guidance and welcoming me as a SERL member.
- Thank you Jim Buchen, for providing me an environment for being a researcher
- Karishma Bhatt, thank you for all the assistance and help with Post graduate regulations and finding information which was not otherwise available.
- BJ, senior technician at school, for technical assistance.
- Barry Dowdeswell, Thank you for encouraging me to use Latex (and thank you for sharing the AUT thesis template designed by Dr. Alan Litchfield)and thank you for many "Learn Latex in one hour" sessions.
- Dr. Roopak Sinha, thank you for helping me with tricky situations using the software and coming forward as a saviour.
- Rivindu Perera, thank you for helping me solving the issues I used to encounter while using Latex.
- Dr. Waqar Hussain (your input for reference checks and your queries about the research questions that helped me think like a researcher), Bilal Raza thank you both for sharing many articles that contributed to my 'learning research', and for your critical comments in the initial stages that was showing a path to a newbie.
- Aftab Ahmed, for keeping the SERL environment very light and entertaining with your humour and the evening instrumental music you used to play.
- To all members of SERL family (Nadia, Awais, Nurul, Nidhi, Tarik, Ammar,and Weihua), thank you as well,

On a personal note

I could not have reached this far without the support and encouragement and overwhelming confidence of my friend, Ambuj Uppal. who has helped me not only financially, but also helped me to keep a positive outlook throughout my journey. Thank you for your spoken and unspoken words helping me to accomplish this journey. Without your support and encouragement, I could not have found the fortitude to complete this research.

I will like to acknowledge the unwavering support of my Kiwi family (my mum Judith Ogilvy- for reading my thesis chapters and my dad Earle Ogilvy); my adopted daughter Vaishnavi (Vaish, looks like we both have adopted each other); and my friend Jyoti Sharma. Thanks for always being there for my family.

Finally, I would like to express my sincere gratitude and thanks to my brothers, sisters, my extended family for their love and affection, trust and belief.

To my parents, brother and my father-in-law in heaven..... This is, indeed a wonderful day ....

Abstract

System Analysis and Design is a core course that is normally taught in the second year of the undergraduate Information System/Information Technology/Computing curriculum. It is an essential course that prepares students for the workplace by incorporating key concepts related to requirements elicitation, use of relevant modelling tools and techniques, and skills such as communication, team building, and time management.
While research exists on understanding the teaching and learning aspects of Systems Analysis and Design, and also on the skills and development practices required in the workplace, there is limited research on investigating the gap between the two (i.e. academia and industry). In particular, to the best of this researcher's knowledge, there is insufficient information on this gap in New Zealand. Therefore, the main objective of this research is to investigate the gap between what content is being currently taught in a typical System Analysis and Design undergraduate course in New Zealand, and the needs and expectations of the industry.
Fifteen semi-structured interviews were conducted with ten industry practitioners (business analysts, system analysts, and project managers) and five academic professionals involved in teaching and delivering an undergraduate System Analysis and Design course.

Interviewing university teachers gave the researcher an insight into the content taught at different universities, and participants from industry provided an insight into the prevailing practices in the software industry of New Zealand and their expectations

of graduates.

Data was analysed using content analysis. The findings highlight some key differences between what is taught in universities and what is currently done in practice, such as methodologies, tools and techniques of System Analysis and Design. This may provide valuable insights into the expectations of relevant stakeholders (teaching teams and IT professionals) from both fields i.e., universities and the software industry in New Zealand.

Contents

Copyright **3**

Declaration **4**

Acknowledgements **5**

Abstract **8**

1 Introduction **15**
1.1 Topic . . . 15
1.2 Background . . . 16
1.3 Rationale . . . 17
1.4 Research Question . . . 20
1.5 Overview . . . 21

2 Literature Review **22**
2.1 System Development Methodologies . . . 23
2.1.1 Introduction to System Development Life Cycle . . . 24
2.2 Related Research . . . 29
2.2.1 Curriculum/Teaching SA&D . . . 30
2.2.1.1 Software Engineering Curriculum . . . 31
2.2.1.2 Information System Curriculum . . . 32
2.2.1.3 Curriculum guidelines . . . 35
2.2.2 SA&D Methodologies . . . 37
2.2.3 Duration . . . 37
2.2.4 Topics . . . 38
2.2.5 Requirements Elicitation . . . 41
2.2.6 CASE Tools . . . 41
2.2.7 Notations . . . 43
2.2.8 Techniques . . . 43
2.2.9 Skills . . . 45
2.2.9.1 Research in New Zealand . . . 47
2.3 Motivation for the present study . . . 50

3 Research Methodology **52**
3.1 Choice of Research Method . . . 52
3.2 Research Philosophy . . . 54
3.3 Ethics Approval . . . 55
3.3.1 Obtaining Informed Consent . . . 55
3.4 Data Collection Methods . . . 56
3.4.1 Rationale and Purpose . . . 57
3.4.2 Research Protocol . . . 58
3.5 Interview Questions . . . 59
3.5.1 Constructing Questions . . . 59
3.5.2 Pilot Interview Questions . . . 60
3.6 Selection of Participants . . . 61
3.6.1 Criteria for Academic Participants . . . 61
3.6.2 Criteria for Industry Participants . . . 61
3.6.3 Invite to Selected Participants . . . 62
3.6.4 Summary Information on Interviews . . . 62
3.7 Data Analysis Methodology . . . 63
3.7.1 Data Analysis Framework . . . 65
3.7.2 Data Transcription . . . 66
3.7.3 Data Cleaning . . . 66
3.7.4 Coding . . . 67
3.7.4.1 Electronic Coding . . . 67
3.8 Summary . . . 70

4 Findings, Analysis, & Discussion **71**
4.1 Overview of SA&D Course content . . . 71
4.2 Findings . . . 73
4.2.1 Participant Information . . . 73
4.2.1.1 Participants from Industry . . . 74
4.2.1.2 Participants from Academia . . . 75
4.2.2 Findings from Academia and Industry . . . 75
4.2.2.1 Research Question #1 . . . 75
4.2.2.2 Research Question #2 . . . 77
4.2.2.3 Research Question #3 . . . 77
4.2.2.4 Research Question #4 . . . 82
4.2.2.5 Research Question #5 . . . 84
4.2.2.6 Research Question #6 . . . 87
4.2.2.7 Research Question #7 . . . 91
4.3 Analysis and Discussion . . . 95
4.3.1 Methodology . . . 98
4.3.2 requirements elicitation . . . 99
4.3.3 Techniques . . . 100
4.3.4 Tools . . . 101
4.3.5 Notation . . . 102

4.3.6 Skills . . . 103
4.3.7 Gap Analysis . . . 105

5 Conclusions and Further Research **108**
5.1 Conclusions . . . 108
5.1.1 Findings Gaps . . . 109
5.1.1.1 Methodology . . . 109
5.1.1.2 Techniques . . . 110
5.1.1.3 Tools . . . 110
5.1.1.4 Skills . . . 111
5.2 Limitations of the study . . . 111
5.3 Implications . . . 112
5.3.1 Implications for Academics . . . 112
5.3.2 Implications for Industry . . . 113
5.4 Future Study . . . 114

References **116**

Appendices **121**

List of Tables

2.1 Literature Review Roadmap 23

3.1 Information on Interviews 63
3.2 Approaches to Content Analysis (as adapted from (Hsieh & Shannon, 2005)) 64
3.3 Process of Interview Transcription 66

4.1 Demographic Information of Industry Participants 74
4.2 Demographic Information of University Participants' roles 75
4.3 Course Info as per programmeme /Major 76
4.4 requirements elicitation Techniques Taught in Universities 79
4.5 Modelling Techniques Taught in Universities in New Zealand 81
4.6 CASE Tools Taught in New Zealand Universities 83
4.7 Methodologies Participants from the software Industry used 85
4.8 requirements elicitation Techniques used In the NZ Software Industry 88
4.9 Modelling Techniques Used In the Industry 89
4.10 Tools Used for requirements elicitation in New Zealand Software Industry . 92
4.11 Modelling Notation Used in Industry 94
4.12 Summary of Academics Findings from Literature Review and Present Study 95
4.13 Soft Skills Expected by Industry for an SA&D Role 103

F.1 A snapshot of the curriculum taught in different universities in New Zealand 144

List of Figures

1.1 *Present Framework: SA&D Course Taught vs Software Industry in NZ* Framework. 18
1.2 *Research Roadmap* 20

2.1 *Role of the Literature Review in the present study* 23
2.2 *Waterfall Model Life Cycle (Dennis & Wixom, 2002)* 25
2.3 *The unified process* (Keith, 2003) 27
2.4 *Life cycle og an Agile model* (Keith, 2003) 28
2.5 *Suggested Topics and Contact hours for Software Modelling Analysis LeBlanc et al (2014)* 33
2.6 *Suggested Topics and Contact hours for Requirements Analysis and Specification LeBlanc et al (2014)* 34
2.7 *Suggested Topics for System Analysis and Design (LeBlanc et al., 2014)* 35
2.8 *IS 2010 Model Curriculum-Career (Topi et al., 2010)* 36
2.9 *ICT Business and Systems Analysts' employment* 49
2.10 *ICT Forecast for Business and Systems Analysts' employment* 49

3.1 *Research Framework* 65
3.2 *NVivo output for Nodes* 68
3.3 *NVivo output for Word Cloud* 69
3.4 *NVivo output for Word Search for 'UML'* 69
3.5 *NVivo output for searching BPMN* 70

4.1 Distribution of three common tools used for requirements elicitation in NZ Software Industry 93

G.1 *NVivo Nodes* 145
G.2 *NVivo Output showing References for BPMN in Industry* 146
G.3 *NVivo Word Cloud* 147
G.4 *Comparison of Tools& Techniques between Academia and Industry* . . 148

Chapter 1

Introduction

1.1 Topic

This study investigates the alignment of System Analysis and Design (SA&D) education with software industry expectations in New Zealand.

The aim of this research was to explore the content, skills, methodologies, tools and techniques offered in a one-semester undergraduate SA&D course, and compare them with the skills, methodologies. and techniques entry level computing/business graduates are expected to have in New Zealand's software industry.

A literature review was conducted first to gain an in-depth understanding of concepts and related studies in the extant literature. Data was collected by conducting 15 semi-structured interviews with both universities academics (5) and software practitioners (10). This data was analysed using content analysis to understand the relevance of the course to the requirements of the industry. It is hoped that this research will benefit universities as they reflect on whether their courses meet the expectations of the contemporary software industry, to understand the needs of the practitioners, and to provide the practitioners with an overview of the SA&D curriculum taught at the undergraduate level.

This chapter is divided into four sections. Section 1 gives the background of SA&D, and Section 2 focuses on the rationale for this research. The research questions are presented in Section 3 and finally, Section 4 provides an overview of the organisation of the thesis.

1.2 Background

Whereas Information Technology (IT) refers to a combination of hardware, software and services, an Information System (IS) combines IT, people and the available information to support a business. This is because IS handles day to day business needs to help managers and their teams (such as developers, business and system analysts, and testers) to make sound business decisions (Rosenblatt, 2014). Since these team members work in different domains, they use different terminologies. Therefore, it is important that they are able to communicate with each other and business stakeholders for who they will develop solutions. This will help them to understand the stakeholders' requirements and give the team members the ability to manipulate business needs.

System Analysis and Design (SA&D) is a course that focuses on addressing the above skills and tasks, and is often a core requirement for an undergraduate degree programmememe in IS and Software Engineering (SE) curricula (Burns & Klashner, 2005). In fact, it is the process of looking at a business problem, identifying and analysing the business requirements and developing a system that will add value to the stakeholders (Dennis & Wixom, 2002).

In New Zealand, education institutes and universities normally teach this course to second year undergraduate students of different majors/programmes, such as Computer Science (CS), and Information Science and as a course for business students completing an Information Systems major.

1.3 Rationale

Typical content for an SA&D undergraduate course includes the introduction to system development methodologies, requirements elicitation, and data and process modelling tools and techniques for analysis and design. In New Zealand six universities teach this course under different curricula. A detailed outline can be found in appendix F in Table F.1.

In an SA&D course, students are typically expected to analyse the problem domain, collect the requirements, produce models and create related documentation using tools and techniques. Based on the content taught, students are required to apply the knowledge and skills to course assessments and projects. This is designed to help the students develop skills they may require in the industry.

In a three-year undergraduate university degree programme, this course is normally taught to second year students. For example, at university #1, an SA&D course is taught as Data and Process Modelling (DPM) to second year undergraduate students. This one-semester course's objectives may includer:

- To train students in areas from requirements gathering to system designing
- To introduce the concept of object-oriented techniques to students
- To complete an individual and group assignment that teaches them soft skills such as:
  - Business needs: To understand the business needs students are taught to gather requirements using different elicitation techniques.
  - Problem solving skills: The course encompasses a group assignment. This helps students to solve issues/ conflicts, and deal with lack of understanding of a given case study. This helps them to learn to cope with stress and remain focused on the task.
  - Interpersonal skills: The group work helps students to learn communication skills, team building skills, and time management skills.

To deliver such a course, academia from universities requires sufficient time (contact hours and tutorials) and resources such as course-related assignments (exercises in tutorials, group/individual assessments and sometimes a mid-semester exam).

At the end of each semester, the university students are invited to evaluate the course by commenting on their satisfaction with the structure and delivery of the course. As per a study conducted in New Zealand, it helps students to give their feedback as it is considered important when universities are re-designing or reflecting on the course content (Erturk, 2014). This evaluation, therefore, assists in improving the design and delivery of the course to benefit the students. This framework is shown in Figure 1.1. However, such an evaluation practice is missing when it comes to understanding the needs of the software industry. This present study has indicated that universities in New Zealand do not collaborate with industry for any input/feedback in terms of the course content, such as what content should or should not be taught.

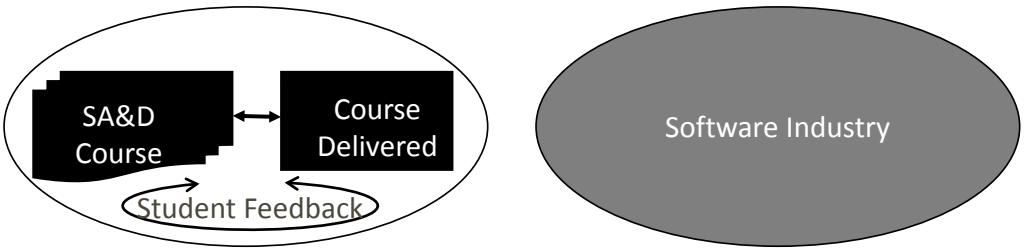


Figure 1.1: *Present Framework: SA&D Course Taught vs Software Industry in NZ* Framework.

Course content, learning outcomes, and activities including associated teaching are designed in an attempt to provide students with exposure to what is expected in the workplace. In order to simulate real life projects, students get case studies/scenarios where the teaching team act both as facilitators and clients. There is sufficient evidence that universities play an important role not only in providing the disciplinary knowledge but also helping students develop project management skills (Robert, 2000).

Apart from classroom learning, the Department of Information Technology of my university invites industry to conduct seminars to interact with students either as a speaker or a client, facilitates workshops, organises career fairs, and conducts on-campus interviews. Yet universities are often criticized because of their obsolete curricula, poorly designed courses that do not attract good students, insufficient emphasis on soft skills, devoting too much attention to theoretical topics rather than practical applications, and providing students with too little experience in the practical techniques of building a large system (Robert, 2000).

This raises a question- if on completion of their degree programme, would our university graduates be ready to meet the needs and expectations of the software industry? While there is research emerging in this area in other parts of the world, such as the United States ([1]hapter 2 will discuss studies studies conducted by (Guidry, Stevens & Totaro, 2011; Powell & Yager, 2013; Guidry & Stevens, 2014; Radermacher, Walia & Knudson, 2014; Karanja, Grant, Freemen & Anyiwo, 2016) to the best of my knowledge there is no evidence in a New Zealand context of industry involvement and evaluation in terms of whether the SA&D courses taught by universities reflects the needs of the software industry .

This limited research in New Zealand context prompted me to conduct a study which involved six New Zealand universities including my own university. SA&D courses were investigated in order to understand the gap between the courses taught and the needs and expectations of the local software industry. A research roadmap is shown in Figure 1.2.

Universities course content of SA&D were investigated along with the practices of the software industry by conducting semi-structured interviews. The interviews were transcribed and then coded using NVivo. Content analysis was used for analysing the findings to find the similarities and gaps if any.

[1]C

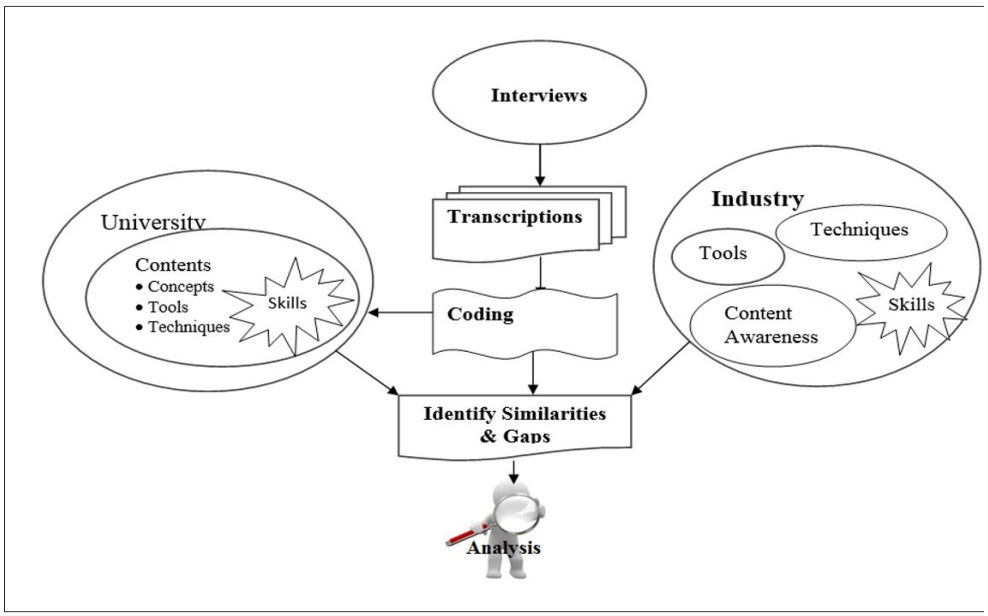


Figure 1.2: *Research Roadmap*

1.4 Research Question

The main objective of this research is to explore the following question:

Does the course content of an undergraduate SA&D course taught in a computing curriculum (e.g. Information Systems, Information Technology, and Computer Science) **in the New Zealand universities meet the expectations of the New Zealand software industry?**

This question is divided into following sub-questions to explore content of SA&D course (RQ):

- **Research Question# 1 (RQ1):** Under what major is an SA&D course is taught in the New Zealand universities and what is the duration of these courses?

  This question is relevant for this research because it will help to find if university graduates are suitable for jobs the software industry is looking to fill or create. Investigating the duration of the course is important because that will determine the time and depth of the topics covered during the course.

- **Research Question# 2 (RQ2)**: What methodologies are covered in a typical SA&D course in New Zealand?
- **Research Question# 3 (RQ3):** What skills and techniques are covered in an SA&D course in New Zealand?
- **Research Question# 4 (RQ4):** What tools and notations are covered in a typical SA&D course in New Zealand?
- **Research Question# 5 (RQ5):** What methodologies are used in the software industry in New Zealand?
- **Research Question# 6 (RQ6):** What skills and techniques are needed in the software industry in New Zealand?
- **Research Question# 7 (RQ7):** What tools and notations are practised in the software industry in New Zealand?
- **Research Question# 8 (RQ8):** What are the main differences (i.e., gaps) between (RQ2, RQ3, RQ4) and (RQ5, RQ6, RQ7)?

1.5 Overview

This thesis is divided into the following chapters:

Chapter 2 - Literature Review: This chapter will review related literature and similar research that has been done in different parts of the world.

Chapter 3 - Research Methodology: This chapter will describe the methodology and rationale for selecting the data collection method used for this research.

Chapter 4 - Findings Analysis and Discussion: This chapter will present findings and detailed analysis of the data gathered during this research

Chapter 5 - Conclusion: This chapter will conclude the research, identify the study's limitations, and offer some recommendations for future research.

Chapter 2

Literature Review

This chapter is divided into ten sections. Section 2.1, gives an outline of the System Development Life Cycle (SDLC) and software development methodology. Section 2.2 highlights the studies that have investigated related research in an SA&D environment. Section 2.2.2 gives a brief description of methodologies covered in this present study, while Section 2.2.3 points to the duration of a typical SA&D course at various universities. Topics, tools, notations, and techniques of an SA&D course are discussed in Sections 2.2.4, 2.2.6, 2.2.7, 2.2.8 and 2.2.9.

A literature review helps to give a background of the existing research so that duplication can be avoided, to highlight any erroneous research, to assist a researcher in selecting a methodology for data collection and analysis, and identify any gaps in the research (Hart, 2005). Consequently, the literature review assisted in finding the research that has taken place in the field of SA&D and any investigations that have taken place that could relate to this research.

This present research used an inverted paradigm approach. It first attempted to understand the scope of SA&D, followed by investigating the research conducted overseas and in New Zealand. The scope was then narrowed down to the present study's hypothesis i.e., investigating the SA&D courses taught at New Zealand universities and

then mapping their relevance to New Zealand's software industry.

Figure. 2.1 shows the role of the literature review in conducting the present study.

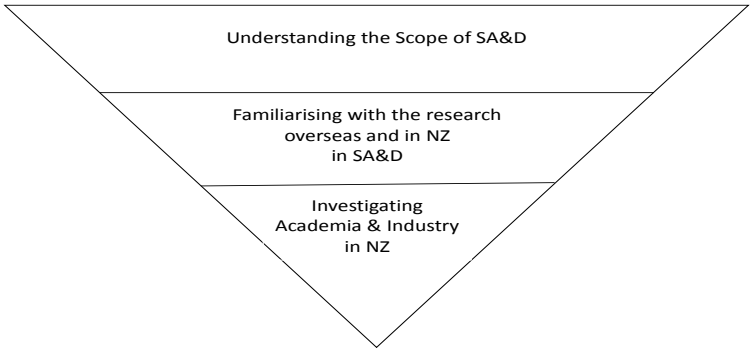


Figure 2.1: *Role of the Literature Review in the present study*

A road map for this research is shown in Table 2.1

Table 2.1: Literature Review Roadmap

Section#	Title of Section	Description
1.	System Development Methodologies	This section will cover the importance of: Structured, Object-oriented, and Agile methodology
2.	Related Research	In this section SA&D topics taught in the course will be explored, such as the curriculum and its guidelines, SA&D methodologies, duration, requirements elicitation, CASE tools,notations, techniques and skills. World perspective of these topics will be discussed wherever researcher could find it
3.	Motivation for the Current Research	As the title indicates this section will highlight the rationale for this research

2.1 System Development Methodologies

For the development of an information system, it is important that business requirements are met so that the developed system can add value for the stakeholders (Dennis, Wixom

& Tegarden, 2015). For this purpose, a system goes through a life cycle so that high-quality standards are maintained and a system is delivered that is on time and is within budget (Mishra & Deepty, 2013).

2.1.1 Introduction to System Development Life Cycle

Before exploring the system development methodologies, it is important to understand the life cycle of a system/IS project. The system or IS project undergoes the four fundamental stages (phases) of a system development life cycle (SDLC):

Planning: During this stage, the assessment of risks is evaluated so that the feasibility of a system can be evaluated.

Analysis: In the analysis stage, the requirements of the stakeholders are gathered and only those requirements that will add value for the stakeholders are considered. It first inspects the existing system so as to develop a new system that will add value for the stakeholders. Then the requirements are gathered and used to develop the model for a potential system. Thereafter, a proposal documentation for stakeholders is created that has the proposed changes for the existing system(Dennis et al., 2015).

Design: Based on the analysis, a design is proposed.

Implementation: Once the system is designed, it is ready for implementation.

Depending on the course curriculum of the SA&D course, universities may or may not teach all of these stages. One of the objectives of this research was to investigate the content of the SA&D courses taught at different universities.

To develop a system that will add value for the stakeholders, it is necessary to follow a formalized step-by-step approach to help employees to remain focused on developing a system that will cater to the requirements of the stakeholders. This is known as 'SA&D methodology'.

A good software methodology is one that can be learnt in a reasonable amount of

time and will achieve similar results whenever it is used (Klopper, Gruner & Kourie, 2007). According to Burn and Klashner (2005) a number of methodologies are known to have evolved with time but debate continues as which one is the best to use.

Therefore, this research will focus on three methodologies, namely, structured or traditional, object-oriented and Agile methodologies. This is because I believe if students have an understanding of these methodologies, they will be able to learn any other methodology they will encounter at their workplace.

1. **1.Structured Methodology** Structured or traditional methodology is the first known and accepted methodology that spends a lot of time in understanding the requirements of the stakeholders. This was the first methodology that introduced the formal modelling or diagramming techniques to represent data and processes. It uses a set of diagrams to represent data and processes (Dennis et al., 2015).

   Waterfall methodology, which is a structured methodology, is also known as a 'Top down Approach' (Isaias & Issa, 2015), relates to the structured/traditional methodology and follows a structured step-by-step approach that logically moves from one stage to another (Dennis et al., 2015). Each stage is quite distinct from the others and requires a sign-off before moving to another step. Public sector,

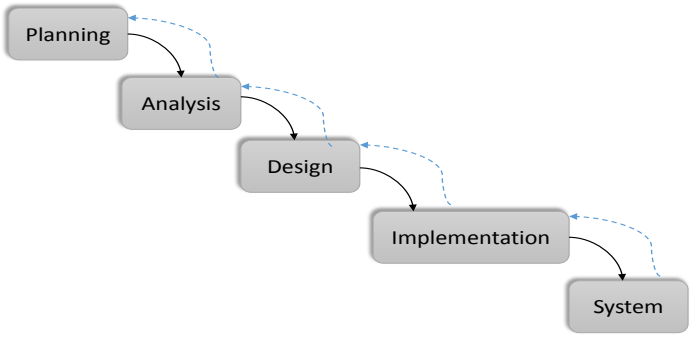


Figure 2.2: *Waterfall Model Life Cycle (Dennis & Wixom, 2002)*

government organizations and large scale projects use structured methodology.

Therefore it is a popular methodology and is taught in a typical SA&D course and is investigated by many researchers (for example,(Powell & Yager, 2013);(Bajaj, Batra, Parsons & Siau, 2005);(Harris, Lang, Oates & Siau, 2006)). Since understanding the requirements of various stakeholders requires very clear, concise and direct questions and communication, teaching this methodology may help students to learn various data gathering techniques.

In this present study one of the areas explored was if this methodology is being used in the software industry and/or being taught in the participating universities in New Zealand.

Figure 2.2 shows the Waterfall Model of the System Development Life Cycle

2. **Object-oriented Methodology (OOM)**

   This methodology aims to capture the structure and behaviour of information systems. It breaks the system into smaller modules (Dennis et al., 2015) that are simpler to understand, and therefore easier to share with team members and stakeholders in order to discuss the feasibility of a new system. It looks at a small entity first, and then builds the system and hence it is also known as a 'bottom-up approach' (Lin, 2011).

   OOM was invented by software engineers from the aerospace, business and process control domains (Rob, 2004). It is an incremental and iterative approach adapted from IBM's Rational Unified Process (RUP) that aims to define who is responsible for an activity, measures an input and an output for the activity being performed, schedules an activity, and instructs someone on how to perform an activity(Hunt, 2003) .

   Therefore, instead of focussing on stages, OOM focuses on phases.

(a) **Inception**: This defines the feasibility of the system. A vision document is delivered as a milestone.

(b) **Elaboration**: This phase captures project-related functional and non-functional requirements. A deliverable is a baseline architecture document or a prototype that may have a construction plan.

(c) **Construction**: This phase concentrates on completion of the analysis, designing, and building of the product. A full beta release is usually the milestone for this phase.

(d) **Transition**: This phase involves activities such as deploying and maintaining the system. Final release of the system is the deliverable for this phase.

Figure 2.3 shows the phases of the object-oriented model.

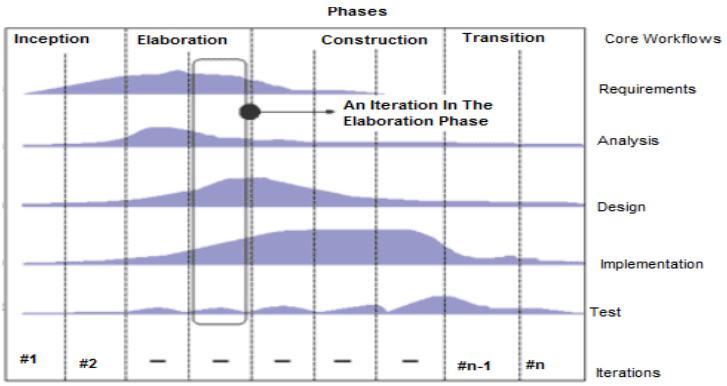


Figure 2.3: *The unified process* (Keith, 2003)

OOM is iterative and incremental in nature and is considered to improve the quality of a system built on this methodology with each iteration. Furthermore, it uses defined workflows to complete its work products. It is highly versatile and can be tailored according to the complexity of a given project and also an organization's culture, as an Agile software process (Keith, 2003).

3. **Agile Development Approach**

Agile is an emerging methodology. Empirically, it has been derived from the iterative methodology and is based on iterative development (Williams, 2007). In this methodology, the focus is on development without spending an extensive time solely on elicitation and establishing fixed requirements. This is because it is not possible for stakeholders to identify what they need until they see a functional

system (Keith, 2003). Therefore, stakeholders are kept involved throughout the process so that they can see the prototype at a much earlier stage. A release, which may be internal in nature, may help them to see if the resultant system will meet their requirements (Lin, 2011). In case stakeholders and the development team are not on the same page, then the requirements are refined and are incorporated in the next release.

Agile methodology adapts itself quickly as per the changed requirements of stakeholders, and helps to supply the deliverables more frequently than other approaches(Balaji & Murugaiyan, 2012).

Figure 2.4 shows the life cycle of an Agile model

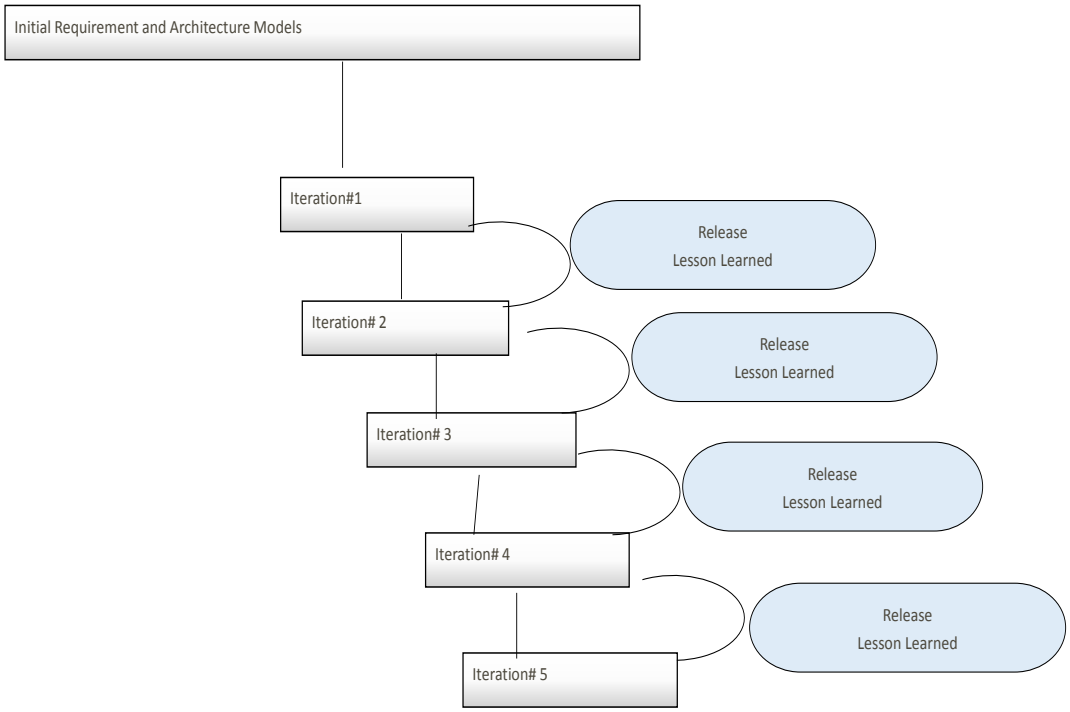


Figure 2.4: *Life cycle og an Agile model* (Keith, 2003)

However, there is not a single approach that 'fits for all' (Dennis, Wixom & Tegarden,

2009); (Dennis et al., 2015). The approach is project-driven; consequently, it is important that students are familiar with some methodologies when they graduate. As stated earlier, covering the structured, object-oriented and Agile methodologies in an SA&D course will help students to have a foundation for SA&D methodologies, and they can enhance their knowledge in whatever environment they will be working.

2.2 Related Research

As SA&D course is taught to students from different disciplines (CS, IT, and IS) with different objectives (technical as well as business). Therefore, it is important to investigate how the course is designed because different disciplines will have different requirements. However, whatever courses' requirements may be, an SA&D course in general teaches to cater the requirements of businesses; this requires a number of technical and non-technical skills such as gather requirements, analysing them and then modelling these requirements to show it to their stakeholders how a business solution may look. , brainstorming, problem solving skills, analytical skills, team work and team building skills. Hence It is important for students to know importance of various methodologies, knowledge of tools and techniques to develop a system and produce documentation. These topics make an outline for an SA&D course. The section below will investigate the research that has taken place on various components of the SA&D course curricula that overseas universities teach.

On investigating a number of articles from overseas, Karanja et al (2016) discovered that the typical SA&D course covers a systematic methodology for analysing a business problem or opportunity, determining what role, if any, computer-based technologies can play in addressing the business's requirements. Therefore, it is important that the students have both skills from the technology side as well as the business side; in fact, many researchers have focused their research on the skills needed for entry-level

graduates in an SA&D role (Powell & Yager, 2013; Karanja et al., 2016).

In Section 2.2.1 SA&D curriculum is discussed including the software engineering and information system curricula. In the remaining sections, SA&D methodologies, duration of the course, topics, requirements elicitation, CASE tools, notations, and skills are explored. Wherever possible, the world perspective is examined. Thereafter, research in the New Zealand context is discussed.

2.2.1 Curriculum/Teaching SA&D

As discussed in Chapter 1, the SA&D course is taught in the undergraduate curriculum of IS, SE, and CS programmes. For such an essential course, it is recommended that an outline or guideline of the course is available that may help universities to choose content for teaching in a specific curriculum. However, because of the volume of the content of an SA&D course, and because of the dynamic nature of the software industry, it is somewhat 'enigmatic in its purpose' (Burns, 2011).

There are a number of technical topics covered in a typical SA&D course such as methodology, tools and techniques and if universities have to choose,then different universities will choose different topics (Harris et al., 2006). It is not that the technical topics are not important, but because it is essential to teach them in such a way that students develop some non-technical skills such as analytical skills, problem solving skills, critical thinking, teamwork, and communicating skills in order to manage the various stakeholders. Not only this, but it is critical to determine if it should be taught in one semester or more, and to ascertain how much time should be spent on each topic.

Looking at these critical factors, research by Batra and Satzinger(2006) highlighted the issues such as dropping some content when revising an SA&D courses . This is because some industries may move to different methodologies and some may keep

using all.[1] Therefore, while teaching traditional methodology, these authors indicated dropping some of the content related to traditional methodology because of a shift in the industry and claim that the teaching teams should try out other methodologies such as Agile. Guidry et al (2011) and Tastle and Russell (2003) investigated the course content in terms of topics (such as methodologies, requirements elicitation, modelling and techniques) that must be taught in the course, or ones that may be dropped.

In an attempt to formulate a set of specific guidelines to teaching SA&D, bodies such as the Association for Computing Machinery (ACM), Institute of Electrical and Electronics Engineers Computer Society (IEEE CS), Association for IS 2010 Curriculum Guidelines for Information Systems (AIS) Association of Independent Information Professionals (AIIP, formerly DPMA) and the International Federation for Information Processing (IFIP), together decided to contribute significantly to curriculum development for these disciplines (Topi et al., 2010).

According to a technical report, in 1998, IEEE and ACM got together to perform a major review of curriculum guidelines for undergraduate programmes in computing. They established ten areas as Software Engineering Education Knowledge (SEEK) areas listing them with one of three indicators of Bloom's attributes(k (knowledge), c(comprehension), or a (application) and topic relevance by e(essential), d(desirable) or o(optional)) (IEEE, 2013).

2.2.1.1 Software Engineering Curriculum

The first set of guidelines for software engineering curricula was published in 2004 when computer societies from Australia, and Britain along with the Information Society from Japan, joined the ACM Education board and IEEE-Computer Society Educational Activities Board to create a Software Engineering 2004 project to suggest a curriculum to be used across numerous computing disciplines (LeBlanc et al., 2014). Then, in

[1] As mentioned earlier that in software industry approach is project driven

2010, a task force was appointed by the ACM and IEEE CS to revisit and check if the curriculum should be updated. To do this, the team met and consulted academics, industry, and government organisations through workshops at technical conferences and through an on-line survey.

Content of SA&D courses was grouped into Software Modeling Analysis (SMA), Software Design (SD), and Requirements Analysis and Specification (RAS). As per this division, SMA could be considered as a core subject in any engineering area because it is vital in documenting and assessing the design and the alternatives available. RAS covers the requirements related to the stakeholders, from elicitation to analysis and documentation which plays a crucial role in SD.

Recommended topics that may be covered under SMA are shown in Figure 2.5 whereas Figure 2.6 recommends topics to be covered under RAS under the SE curriculum. These figures show the suggested numbers of hours to be spent on each topic.

2.2.1.2 Information System Curriculum

The widely-accepted curriculum for IS was developed in 2002. The curriculum was updated so that academia could produce knowledgeable, skilled, and confident entry-level graduates. Besides, for a number of emerging domains IS is vitally important. Therefore, it is important to reassess and revise the IS curriculum so that the graduates are equipped with fundamental knowledge and skills (both technical and non-technical) of IS domain (Topi et al., 2010). The authors believe that it is equally important that the course remains flexible, and elective and core papers remain separated in core curriculum so as to support specified degrees.

According to the report, the core course in SA&D focuses only on the business requirements, process of gathering, analysing, documenting and converting these requirements into system requirements. This will initiate learning in communicating

Reference		k,c,a	E,D	Hours
MAA	**Software modeling and analysis**			28
MAA.md	Modeling foundations			8
MAA.md.1	Modeling principles (e.g., decomposition, abstraction, generalization, projection/views, and use of formal approaches)	c	E	
MAA.md.2	Preconditions, postconditions, invariants, and design by contract	c	E	
MAA.md.3	Introduction to mathematical models and formal notation	k	E	
MAA.tm	Types of models			12
MAA.tm.1	Information modeling (e.g., entity-relationship modeling and class diagrams)	a	E	
MAA.tm.2	Behavioral modeling (e.g., state diagrams, use case analysis, interaction diagrams, failure modes and effects analysis, and fault tree analysis)	a	E	
MAA.tm.3	Architectural modeling (e.g., architectural patterns and component diagrams)	c	E	
MAA.tm.4	Domain modeling (e.g., domain engineering approaches)	k	E	
MAA.tm.5	Enterprise modeling (e.g., business processes, organizations, goals, and workflow)		D	
MAA.tm.6	Modeling embedded systems (e.g., real-time schedule analysis, and interface protocols)		D	
MAA.af	Analysis fundamentals			8
MAA.af.1	Analyzing form (e.g., completeness, consistency, and robustness)	c	E	
MAA.af.2	Analyzing correctness (e.g., static analysis, simulation, and model checking)	a	E	
MAA.af.3	Analyzing dependability (e.g., failure mode analysis and fault trees)	k	E	
MAA.af.4	Formal analysis (e.g., theorem proving)	k	E	

Figure 2.5: *Suggested Topics and Contact hours for Software Modelling Analysis LeBlanc et al (2014)*

with stakeholders. The course content should also include the structured SDLC, object-oriented analysis and design (OOASD), with some learning of Unified Modelling Language (UML), and Agile methods. A project should be designed on this content so that students get exposure to develop documentation. This will give them a real-life experience by incorporating some project management practices. According to the IS report, some of the learning objectives of SA&D include- identifying the problem, using a methodology to analyse the problem, and communicating with stakeholders for specifying and writing the business requirements using CASE tools for data and process modelling.

Figure 2.7 recommends the topics for an SA&D course. However, these topics can be adapted based on the requirements of the specific major/programme/discipline.

Reference		k,c,a	E,D	Hours
REQ	**Requirements analysis and specification**			30
REQ.rfd	Requirements fundamentals			6
REQ.rfd.1	Definition of requirements (e.g., product, project, constraints, system boundary, external, and internal)	c	E	
REQ.rfd.2	Requirements process	c	E	
REQ.rfd.3	Layers/levels of requirements (e.g., needs, goals, user requirements, system requirements, and software requirements)	c	E	
REQ.rfd.4	Requirements characteristics (e.g., testable, unambiguous, consistent, correct, traceable, and priority)	c	E	
REQ.rfd.5	Analyzing quality (nonfunctional) requirements (e.g., safety, security, usability, and performance)	a	E	
REQ.rfd.6	Software requirements in the context of systems engineering	k	E	
REQ.rfd.7	Requirements evolution	c	E	
REQ.rfd.8	Traceability	c	E	
REQ.rfd.9	Prioritization, trade-off analysis, risk analysis, and impact analysis	c	E	
REQ.rfd.10	Requirements management (e.g., consistency management, release planning, and reuse)	k	E	
REQ.rfd.11	Interaction between requirements and architecture	k	E	
REQ.er	Eliciting requirements			10
REQ.er.1	Elicitation sources (e.g., stakeholders, domain experts, and operational and organization environments)	c	E	
REQ.er.2	Elicitation techniques (e.g., interviews, questionnaires/surveys, prototypes, use cases, observation, and participatory techniques)	a	E	
REQ.rsd	Requirements specification and documentation			10
REQ.rsd.1	Requirements documentation basics (e.g., types, audience, structure, quality, attributes, and standards)	k	E	
REQ.rsd.2	Software requirements specification techniques (e.g., plan-driven requirements documentation, decision tables, user stories, and behavioral specifications)	a	E	
REQ.rv	Requirements validation			4
REQ.rv.1	Reviews and inspections	a	E	
REQ.rv.2	Prototyping to validate requirements	k	E	
REQ.rv.3	Acceptance test design	c	E	
REQ.rv.4	Validating product quality attributes	c	E	
REQ.rv.5	Requirements interaction analysis (e.g., feature interaction)	k	E	
REQ.rv.6	Formal requirements analysis		D	

Figure 2.6: *Suggested Topics and Contact hours for Requirements Analysis and Specification LeBlanc et al (2014)*

These recommendations were for content only and did not involve instructions for any pedagogical practices. Figure 2.8 shows the seven core IS courses and nearly 50% require significant coverage of the content of SA&D and the remaining 50% require at least some knowledge of the content covered in the course.

After carefully examining the content of the Data and Information Management courses, it was observed that some of the topics of SA&D are covered in this course as well. Thus, it can be projected that about 75% of the careers recorded in Figure 2.8 need knowledge of SA&D content.

IS 2010 Curriculum Guidelines
Identification of opportunities for IT-enabled organizational change
• Business process management
• Analysis of business requirements
Business process modeling
Information requirements
• Structuring of IT-based opportunities into projects
• Project specification
• Project prioritization
• Analysis of project feasibility
Operational
Tangible costs and benefits (financial and other measures such as time savings)
Intangible costs and benefits such as good will, company image
Technical
Schedule
Legal
Cultural (organizational and ethnic)
• Fundamentals of IS project management in the global context
• Using globally distributed communication and collaboration platforms
• Analysis and specification of system requirements
Data collection methods
Methods for structuring and communicating requirements
Factors affecting user experience o User interface design
System data requirements
Factors affecting security
Ethical considerations in requirements specification
• Different approaches to implementing information systems to support business requirements
Packaged systems; enterprise systems
Outsourced development
In-house development
• Specifying implementation alternatives for a specific system
• Impact of implementation alternatives on system requirements specification
• Methods for comparing systems implementation approaches
• Organizational implementation of a new information system
• Different approaches to systems analysis & design: structured SDLC, unified process/UML, agile methods

Figure 2.7: *Suggested Topics for System Analysis and Design (LeBlanc et al., 2014)*

2.2.1.3 Curriculum guidelines

Based on what curriculum (Information System (IS), Information Technology (IT), or Computer Science (CS) universities decide to teach an SA&D course, academics should examine the available guidelines (Khoo, 2011). For example, if the course is dedicated to IS, and keeping in mind the requirements and expectations of the local industry, universities should look at the guidelines of the latest IS 20XX curriculum. This is because when employers interview new hires they know that graduates are naïve and may need training, but they are still looking for some soft skills (Cappel, 2002). Therefore, universities should focus on the content and the detail they want to cover in the course. This will help them to cater for the needs of the students because they generally enrol in a course to gain some knowledge and learn skills that will make them employable (Khoo, 2011).

Structure of the IS Model Curriculum:Information Systems specific courses

Career Track:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Core IS Courses:																	
Foundations of IS	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Enterprise Architecture	○	●	○	○	○	●	○	○	○	○	●	○	○	○	●	○	○
IS Strategy, Management and Acquisition	○	●	○	○	○	●	○	○	●	○	●	○	○	○	●	○	○
Data and Information Management	●	○	○	●	●	○	○	●	●	○	●	○	●	○	○	○	○
Systems Analysis & Design	●	●	●	○	○	○	●	○	○	○	○	○	○	○	●	●	●
IT Infrastructure	○	○	○	●	○	○	○	●	●	●	○	○	●	●	○	○	○
IT Project Management	●	○	○	○	○	●	○	○	○	○	●	○	○	○	●	●	●
Elective IS Courses:																	
Application Development	●	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●	●
Business Process Management		●	●			○	○	○		○	●				○		
Collaborative Computing						○								○			○
Data Mining / Business Intelligence		●		●	●	○	○	○	●		○	○	○	○	○		○
Enterprise Systems		●	●	○	○	○	●	●	○		●	●	○	○			
Human-Computer Interaction	●					○	○				○					●	
Information Search and Retrieval		○		○	●								○				●
IT Audit and Controls	○		●	○	○	○	○	●		●	○		○	○	○		○
IT Security and Risk Management	○			○	○	○	○	●	●	○	○		●	●	○		○
Knowledge Management		●		○		○	○			○							
Social Informatics													○		○		

A = Application Developer
B = Business Analyst
C = Business Process Analyst
D = Database Administrator
E = Database Analyst
F = e-Business Manager
G = ERP Specialist
H = Information Auditing and Compliance Specialist
I = IT Architect
J = IT Asset Manager
K = IT Consultant
L = IT Operations Manager
M = IT Security and Risk Manager
N = Network Administrator
O = Project Manager
P = User Interface Designer
Q = Web Content Manager

Key:

● = Significant Coverage

○ = Some Coverage

Blank Cell = Not Required

Figure 2.8: *IS 2010 Model Curriculum-Career (Topi et al., 2010)*

With the continuous growth in the IT sector and with businesses becoming multinational, it is very important that the SA&D courses taught at universities should help new graduates to be ready for the software industry (Khoo, 2011). Universities should understand the requirements of the software industry and amend the course accordingly to equip students for their future jobs (Guidry et al., 2011).

As compared to the USA, European countries and neighbouring country Australia, New Zealand has a much smaller population smaller. Majority of them, teach the SA&D courses under Information Systems major (please refer to Table F.1in appendix F) and only two universities teach it under an IT major. To the best of my knowledge, no information is available that they follow a specific curriculum. Therefore it may be said that the SA&D courses taught are designed to train students for both SE and IS

requirements so that they have basic knowledge and skills that make them employable within the software industry (Guidry et al., 2011).

2.2.2 SA&D Methodologies

Looking from the software industry perspective, SA&D is very complex task because it involves a number of stakeholders with number of requirements that vary depending on the role. Therefore it is important to prioritise the requirements, and hence follow some methodology (Banerjee & Lin, 2006). However there is not a single methodology that can be applied to every system (Bajaj et al., 2005).

Overseas studies have revealed that the teaching team of SA&D courses always either cover in detail or at least give an overview of methodologies(Bajaj et al., 2005). Depending on the scope of the course, any textbook on SA&D would either have a dedicated section on methodologies, or at least a topic where the authors would briefly discuss the topic, for example (Dennis et al., 2015)). Research shows that universities teach traditional and object-oriented methodologies. However, Karanja et al (2016) has argued if it is necessary to teach both. In order to investigate the situation, it is important to find out, which methodology is being taught under an SA&D course and to what extent these are covered in a typical SA&D course? If universities do not teach any of these, does it indicate a gap between teaching and researching? This can be found out by investigation the time academics spend on teaching this topic. The following section will discuss the duration of SA&D course.

2.2.3 Duration

As discussed in the previous section, a typical SA&D course covers a number of topics. Depending on the discipline and scope, different universities teach this course differently. A study was conducted by Holmes, Kemm and Yager(2002) reviewing

the SA&D course syllabi of fifty two educational institutes from three regions(North America, Europe and Asia). The majority taught the course in one semester while some indicated that they taught it in two semesters. According to a survey conducted overseas 79% of 172 academics teach this course in two different courses, System Analysis in the first semester and System Design in second semesters (Burns, 2011).

The present study explored how this course was taught in New Zealand universities(i.e., in one semester or two semesters or any other approaches).

2.2.4 Topics

As mentioned in Section 2.2.1, different academics teach this course differently. And as per IS 2010 curriculum guidelines, this course has been treated as a principal course for many majors (Topi et al., 2010). The vast content for this course means the teaching team may sometimes struggle in deciding the importance of topics and include them regardless of the methodologies taught.

Depending on the major/programme, it may be possible that various universities might be teaching different topics to give domain knowledge to students. The SE course focuses on the technical side, whereas IS focuses on the business side. The course also includes project management aspects and considers the financial aspect of the project. According to Bajaj et al (2005), SA&D may be the only course whose content is not covered in other disciplines and recommend high quality research for this course because as per their observation, the textbooks they observed, haven't changed much in content. This indicates a teaching-research gap. Academia in general lacks interest in investigating the course from a teaching perspective, even though there has been interesting developments, such as the software industry going Agile. Lots of articles on SA&D can be found on the web, but the academic community needs to focus on the course by investigating the needs of the software industry.

World Perspective

Methodologies, tools, and techniques is a lot of content to teach,therefore, the question is if the duration of the course is one semester, then how do universities allocate time to teach different topics Tastle and Russell (2003) conducted a survey in which thirty three academics participated to find the hierarchy based on the importance of the topics. It included the structures/traditional and object-oriented methodologies, tools and techniques, system design, and project management. Findings revealed that topics had too many variations. Nonetheless, data modelling was found to be the most important topic, followed by the structured approach. However, for the researchers, the results could not determine the true importance of topics. This is because the number of participants was not enough to make a judgement as not everyone responded to each question.

Bajaj et al. (Bajaj et al., 2005) observed that for such an important core course the research has been very limited. They said as far as the typical SA&D course is concerned there is a teaching-research gap and also there is a need to compare survey results of other research with industry requirements and expectations.

In another study, SA&D was used to expose "students to different methods, tools, and techniques used in developing new systems" (Harris et al., 2006, p. 242). They found that because an SA&D approach is an essential approach to system development, therefore irrespective of the system development methodology taught in an SA&D course, it helps students to develop their technical as well as non-technical skills such as analytical, problem solving and team building skills.

In his survey mentioned above, Burns (2011) found that those who taught the course in one semester considered planning and analysis important parts of the course. 58% taught both Structured and Object-oriented approaches, though out of 58% only 25% focused on the Traditional/structured approach and ERD and DFD were taught more than Use Case and Class diagrams. However, not everyone taught Activity diagrams.

Guidry et al (2011) investigated the skills, topics and time spent on each topics. 214 participants from the Association to Advance Collegiate School of Business (AACSB) completed their survey. The results showed that academics who taught the Structured or Traditional methodology in an SA&D course considered Watefall methodologies, ERD, and DFD important to teach while those who had taught Object-Oriented methodology considered important to teach related topics such as OO concepts, and UML diagrams but sequence diagrams and state diagrams were not given much focus. At the time of their research, Guidry et al (?)Guidry2011) also found that UML had gained popularity, however to benefit both academics and practitioners, the authors recommend to compare results of their study with the software industry requirements and expectations.

In another study, 124 academics and 98 industry participants with a variety of IT roles participated in a survey (Guidry & Stevens, 2014). Since the roles were not specified in their research, it is difficult to say if all of them were involved in SA&D practices. Their study had differences between academics and industry perceptions. The software industry expected graduates to have two technical skills, such as knowing modelling techniques (data and process modelling) and one managerial skill [2]. The two technical skills may be important because because entry-level graduates would not be hired directly for managerial positions. However, the industry expected them to know what they had learnt at school and have some non-technical skills such as team-work and project management skills. Thus for such a vital course, it is a challenge for academics to cover such a large content in a limited time [3].

From the studies mentioned above, it is observed that academics do not follow a set curriculum for SA&D courses. However, it is also true that they do not teach a course with their psyche or intuitions, but with an intention that the content taught may be

[2]The list of managerial skills the industry expected graduates to have, was not available for their study

[3]As discussed in section 2.2.3 depending on the curriculum academics decides the duration of the course differ between one to two semesters

useful in the workplace. A reason for choosing different topics or spending a set time on each topic or both may be due to the possibility that universities may not be aware of the SA&D course content taught at other universities.

2.2.5 Requirements Elicitation

In SA&D requirements elicitation is an important topic (refer to Section 2.2.1.1). This is because for a system to be built requirements need to be elicited, interpreted, analysed, negotiated, verified, validated and documented (Wohlin & Aurum, 2005).

The professionals who are responsible for requirements elicitation gather, analyse, model and validate requirements so as to ensure that a complete set of requirements has been collected (Nuseibeh & Easterbrook, 2000). Before designing and implementing a system, it is important to complete this process (Pandey, Suman & Ramani, 2010), so that a system can be built that will add value for the stakeholders.

Investigations show that in a typical SA&D course, most universities teach requirements elicitations (such as (Harris et al., 2006; Topi et al., 2010; Powell & Yager, 2013; Erturk, 2014)). However, depending on the workload for the course, time is accordingly-divided between different topics. This time allocation depends on what the teaching team wishes to deliver. Therefore in this present study, I investigated if universities consider requirements elicitation (RE) an important topic to teach in New Zealand.

2.2.6 CASE Tools

Software development is becoming more complex day by day, and therefore, to increase efficiency, it is important that the developers have some way of manipulating the artefacts (gathered requirements, project related documentation for stakeholders, any UML diagrams). Therefore, to meet the increasing complexity of software, there is

a need for improved tools and techniques. Consequently, Computer-Aided Software Engineering (CASE) tools are popular with developers because the tools improve the productivity as well as quality of documentation (Dasgupta, Haddad, Weiss & Bermudez, 2007).

For an SA&D course, use of CASE tools are categorised as follows:

- Upper CASE tools that focus on the business processes and data models and
- Lower CASE tool focus on the data models and source code development (Oracle, 2009).

Universities can teach the SA&D course effectively if they use an appropriate tool because it is a teaching and learning instrument that aids graduates when they join the workforce. It also helps graduates to learn a new tool at their workplace (Senapathi, 2005).

World Perspective

In Section 2.2.1, SE and IS curricula recommend certain diagrams emphasize requirements elicitation, and support the idea that students must have knowledge of the tools and techniques taught in the course. For that reason, it is important that the course must include some modelling tools and techniques that would give a foundation to students to make them confident and prepare them for their workplace. However, no specific recommendation is provided to teach a specific tool; it should be the decision of the universities, depending upon the need of the hiring industry (Topi et al., 2010).

According to Dasgupta et al (Dasgupta et al., 2007) the software Industry uses CASE tools for SA&D practices because different methodologies produce different models, and CASE is a supporting technology that increases the efficiency of software development. They also indicated that students would become anxious when they had to learn a new tool. They would use CASE tools only if they thought that its use would positively impact their performance. Therefore, the teaching team should encourage

students to use a tool in producing their documentation related to coursework.

In section 2.2.1, figures from 2.5 to 2.8 show that SE and IS curricula emphasize modelling concepts. Accordingly, some modelling tools should be used to provide some exposure to students on how to use a tools. By doing so they will feel confident and prepared for their workplace.

This present study explored if there were any tools that academics teach, and if these tools are of any benefit to graduates when they join the software industry.

2.2.7 Notations

A notation is "a visual language for visualising, specifying, constructing and documenting software intensive systems" (Moody, 2009). It is used in SA&D because it helps professionals to communicate with non-technical stakeholders. It consists of sets of graphical symbols (visual vocabulary) and compositional rules (visual grammar), and together they form the visual syntax of the notation.

A Unified Modelling (UML) Notation is known to reflect some of the best modelling notations that have been proven useful in practice (Evans, France, Lano & Rumpe, 2014). The other notation which is gaining popularity is the Business Process Modelling Notation (BPMN). It shows a number of activities followed to accomplish a task, so that people involved in SA&D can understand it (Chinosi & Trombetta, 2012).

This present research investigated which notations were taught in universities and which notations were used in local industry.

2.2.8 Techniques

SA&D techniques such as requirement elicitation techniques and data and modelling techniques are important because they help practitioners to develop solutions based on requirements of stakeholders using some standards (Dasgupta et al., 2007). Tools may

not be important, but techniques help to create a solution for stakeholders that will add value for them. This essential technique of SA&D comes by doing (Waldo, 2006) (e.g., requirements elicitation techniques, document analysis, and developing models). Use of these techniques help to build a prototype that could save a lot of time.

Use of techniques in a SA&D course helps students to understand the purpose of documentation, and the importance of communicating and understanding both at high and low levels (Lending & May, 2013).

SA&D practices expect graduates to work on a task that necessitates analysis of the requirements of their stakeholders.

> *"Task analysis is a list of all the things that the users want to carry out using the system, the preconditions required to achieve these goals, the various steps involved and the interdependencies between these steps, and all the various outputs and reports that need to be produced" (Senapathi, 2005, p.12).*

These requirements are then mapped with the final output using a number of techniques as discussed in Section 2.2.1.

According to Lending and May (2013) a combination of reading about techniques, hearing in lectures, enhances the learning. Knowing a technique may help the graduates to learn a different technique at work because of the background knowledge they already have from their learning at university.

This research explored the techniques taught in New Zealand universities. It has been observed that almost all the academics teach, if not all, then at least some of the techniques mentioned in Section 2.2.1. The details are covered in Chapter 4.

2.2.9 Skills

As previously discussed, requirement gathering is very important in SA&D. It requires an understanding of the needs of different stakeholders and their challenges, analysing the needs by prioritising them, examining if a system can be built on those needs using the available resources, and determining if the resultant system will add value for the stakeholders. This requires both technical as well as non-technical skills.

Technical skills are needed to select a methodology, appropriate tools, and techniques using appropriate notation, whereas the most important non-technical skills are needed to communicate with non-technical stakeholders and interacting within teams. Studies have shown that besides having technical skills, employers look for non-technical skills such as communication, team work, problem solving (Cappel, 2002), and project management, task co-ordination, and brain-storming skills (Harris et al., 2006)

World Perspective

Universities impart knowledge to students by teaching them current topics and designing courses that could provide them with management skills (Robert, 2000). However, skills such as the ability to create documents, follow a software methodology, manage a large project, and work in a software team are not normally supported in university teaching (Begel & Simon, 2008). This is because on entering the industry, graduates have to learn new skills, techniques and procedures; in fact, becoming novices all over again, particularly those, who lack communication skills and have a great deal of self-efficacy especially if they get good grades.

Employers expect the new hires to be equipped with the ability to identify problems, and risks, and have skills that are needed for requirement engineering (Powell & Yager, 2013). Therefore, these emerging changes in technology are forcing IS professionals to develop interpersonal skills and management skills so that they can adapt to ongoing

changes. This may mean that the course should be able to impart these skills to graduates as well.

Today software technology in general is dynamic and to cope with this reality, employers have to invest in infrastructure as well as in either hiring new staff or training the existing staff. As a result, they will prefer to hire graduates who possess the skills the business needs(Bataveljic, Eastwood & Seerfried, 2006). The entry level jobs are disappearing. Hence, the expectation to hire the multifaceted professionals, is increasing because of the involvement of technology in day to day business needs. Hence the software industry is looking for aligning the IS with business goals.

In another study semi-structured interviews were conducted with 23 managerial staff to investigate the skill gaps between the graduating students and the industrys expectations (Radermacher et al., 2014). The authors' suggest that students should be exposed to common tools that are used in the contemporary software industry. They should complete their writing assignments, such as reports and documentation well. This will help them to learn to produce a technical report and aid them in communicating their solutions to others. Problem solving and team work are some other skills that industry's practitioners appreciate.

Karanja et al (2016) conducted a survey and analysed 200 jobs advertised in the United States to find the skills that are important for entry-level positions. The authors agree that a qualification does matter, but apart from this, an SA&D course should be designed so as to impart non-technical skills such as co-operative skills, encourage students to be detail-oriented, and ensure they are able to communicate with a diverse range of stakeholders. This is because, researchers have determined that in the IT field, usually employers look for non-technical or soft skills than mere technical skills.

2.2.9.1 Research in New Zealand

A Survey of Software Development Practices in the New Zealand Software Industry focused on the requirements engineering (Groves, Nickson, Reeve, Reeves & Utting, 2000). The motivation for their research was to find the techniques used in requirements elicitation, and if these gathered requirements were used in designing a new system. Their study was conducted in two phases. An invitation was sent to people in 64 industries. In the first phase, 24 companies participated in telephone interviews. Their investigation showed tools and techniques such as Rational Rose, BridgePoint, UML, and class diagrams were more commonly used in the industry. The second round involved conducting in-depth interviews with four large companies. They found out that companies were using both structured/traditional and object-oriented methodologies. With regards to tools being used for requirements engineering, simple tools such as word processor, spreadsheets excel and techniques such as ERD, DFD and use cases, and templates were commonly used. The findings of the study were specific to the software industry and authors did not compare the findings to teaching a typical SA&D course.

In another New Zealand study, Kemp (2003) interviewed five participants from the New Zealand software industry. The size of the organisations varied from small to large. The research focus was on the use of software engineering practices and tools. Investigations showed that industry uses methodologies (waterfall and prototype), tools (Visio, Rational Rose), and techniques (ERD, Class diagram and State chart). The study did not compare the findings with the SA&D course.

Another New Zealand study conducted by Phillips, Kemp and Hedderley (2005) sent a questionnaire to 147 participants from the New Zealand software industry. The findings provided insights into the commonly-used tools and techniques in the industry. The tools used for requirement gathering were extensive compared to Data and Process

modelling. The most common tools used were MS-Word and MS-Project, however the integrated CASE (ICASE) tools such as Rational Rose were rejected due to various factors such as cost and time. It was interesting to find that some participants did not feel the need for having a tool. Techniques such as Entity Relationship Diagrams (ERD) and Sequence Diagrams were used more than the Class diagram and State Diagrams with use case diagrams being used the least. The researchers were from a New Zealand university but they did not compare the study outcome with an SA&D course.

Kirk and Tempero (2012) conducted a survey with 195 participants from New Zealand software industry. The authors investigated software development practices. Their findings revealed that SA&D methodologies were adopted as per the needs of the stakeholder organisations. Though organisations claim to follow an Agile approach the formalities with clients were maintained. Secondly, organisations use tools to support processes but tools could be ineffective because they are difficult to use, are missing, are poor or are out dated. Though they investigated SA&D practices, but the findings were not compared with the SA&D courses taught at universities.

In New Zealand, the universities education and qualifications are looked after by the Committee on University Academic programmes (CUAP) and the New Zealand Universities Academic Audit Unit (NZUAAU). They follow the criteria written in the approval guide (NZQA, 2014) [4]:

Criterion 1: Qualification to which the programme leads

Criterion 2: Title, aims, learning outcomes and coherence

Criterion 3: Delivery methods

Criterion 4: Acceptability of the programme and consultation

Criterion 5: Regulations

Criterion 6: Assessment and moderation

[4](http://www.nzqa.govt.nz/assets/Providers-and-partners/ Registration-and-accreditation/guidelines-degree-approval-and-accreditation.pdf

Criterion 7: Assessment and review

Where this criteria gives an outline, however, it does not have uniform guidelines for designing course material. It is important that the CUAP and NZUAAU should look into the requirements of the software industry and come up with an outline or suggested outline that will help universities to design the course because according to the Ministry of Business Innovation & Employment (MBIE), the prospects for a Business Analyst role is rising. And as per Immigration NZ, business analysts are listed in the long-term skill-shortage category. Figure# 2.9 shows the actual and projected growth of business analysts in terms of percentage and Figure# 2.10 shows the growth in terms of employment

Current		Projected growth*	
2013	2014	2014–19	2019–24
12,070	12,860 up 6.5%	4.6% per year	4.2% per year

Figure 2.9: *ICT Business and Systems Analysts' employment*

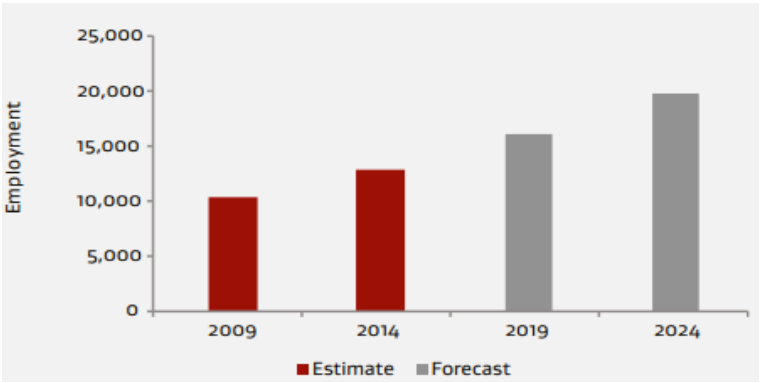


Figure 2.10: *ICT Forecast for Business and Systems Analysts' employment*

Therefore, to fulfil this skilled category universities should prepare graduates with the knowledge and skills industry is looking for.

2.3 Motivation for the present study

The research conducted overseas has extensively examined the course, the topics, their hierarchy, and the time spent on each topic. Some of the research discussed in the literature review investigated the skills required in the software industry. Without having the foundational knowledge and skills that students receive during their university education, it is impossible for IS graduates to exhibit the high level IS capabilities. However, it can be very challenging to design a course that could cover all the requirements of industry.

Students enrol in universities to seek foundational knowledge. With a number of courses to finish in one semester, academics aims to facilitate their knowledge but can only offer so much. Industry should understand and accept this limitation (McCracken, 1997). However, at the same time, academics should also understand that those in the industry recognise that new hires will not be experts in the domain, but they should have certain skills such as time management (i.e. being able to work to deadlines and being punctual),and verbal and written communication skills (so they can communicate effectively with their team). Therefore, it is important that the course is aligned with the needs of the industry.

In New Zealand researchers have studied practices in industry and use of CASE tools, but of all the research explored for literature review, to the best of my knowledge, no specific study was found that could compare the typical SA&D course taught in New Zealand universities with the needs of the industry. Also, it was observed that in the absence of a set curriculum across the universities with different programmes and majors, teaching such a course could not only be challenging but taxing for the teaching staff. This is because it is difficult to choose from a vast number of topics that could be covered in a semester, while at the same time helping students to develop technical as well as non-technical skills. Findings from the literature also show that not all material

covered in a typical SA&D course is utilised in industry. Consequently, the challenge for academia is what should they be teaching that will help graduates to perform well in the industry?

This dilemma was the inspiration for this research. This present study examined six New Zealand universities that taught an SA&D course and compared them with the needs of the New Zealand software industry.

The next chapter will focus on different approaches used to carry out this research.

Chapter 3

Research Methodology

This chapter discusses the methodologies used for the present study and is divided into six sections. Section 3.2 discusses the research philosophy this research is based upon. Since participants from New Zealand universities and the software industry were required for their input, it was mandatory to consider the ethical implications and, hence, ethics approval was required. This is explained in Section 3.3. Following that is Section 3.4 which discusses the data collection methods, rationale for choosing the method and details on creating the research protocol. The next section 3.5 deals with framing and Section 3.6 addresses the participants' information. Lastly Section 3.7 explains the method and process for data analysis.

3.1 Choice of Research Method

The preferred methodology for this present study was a qualitative research method (QRM). This method leads to an understanding of a socio-cultural way of doing things in a set environment (Myers, 2013), and produces findings not arrived at by statistical procedures or other means of quantification (Ritchie & Lewis, 2003). This is because, "Quantitative research is explaining the phenomena by collecting numerical data that

are analysed using mathematically based methods (in particular statistics)'" (Muijs, 2010).

Explaining phenomena means that the method should be able to assist the researcher in investigating questions, such as "Why are tools important in teaching SA&D?" Since in a quantitative research the questions will be closed, and the option for answering will be limited (such as agree, disagree, or may be), the research could be biased.

noindent Quantitative methods require a large number of sampling populations, and researchers need enough time to collect data and process the information. Moreover, analysing data statistically need extra time because of refining data and re-testing again many a times and with such a small sample in the present study, it would not have been possible to do justice to this unique research.

Furthermore, since the objective of this research is to investigate the gap between the SA&D courses taught in universities and the software industry, the option of limiting the participants' replies to questions would not have generated the best data for this study. This is because the intent of this study was not to get the quantitative output. [1]Hence I decided to use QRM, as it has been used in a number of studies for various purposes, such as evaluating IT systems, and understanding the issues associated with the modernisation of information and technology in IS (Kaplan & Maxwell, 2005; Myers et al., 1997).

In the present study, QRM was used to understand the gap between universities' teaching of SA&D and the expectations of software industry in New Zealand by evaluating the SA&D courses as well as and practices prevalent in the industry.

[1]As discussed in a quantitative study the choices of answers to questions is 'Yes/No/May be'

3.2 Research Philosophy

Positivism is a research philosophy that assumes, whether it is believed or not, that reality always remains true and can be measured irrespective of the instruments used by the researcher (Myers et al., 1997). This means that the researcher must focus on exploring and analysing the facts by remaining objectives. In the social sciences, positivism refers to finding the facts of truth using various experiments to make a discovery. This discovery then gets added to the body of knowledge (Grant & Giddings, 2002).

A positivist research philosophy is dominant in IS (Orlikowski & Baroudi, 1991). This is because organisations have a set structure and work in a set environment. Researchers can objectively evaluate or predict actions or processes, without getting involved in moral judgements or subjective opinion.

In the present study, the focus is on finding the factual information in the field of SA&D and analysing it to make a discovery so that a body of knowledge is created. Hence, the current research fits well with the positivist approach. The unbiased analysis from this study will highlight the gap, if any, between academia from universities and the software industry.

The body of knowledge thus created may help the participants from both the fields. Firstly, it may help universities to revisit and reflect on their courses to check if they meet the needs of the software industry. Secondly, the body of knowledge may be able to help industry practitioners when interviewing graduates for their graduate programmes/co-op trainings, or the job advertised, or offered after the campus interviews, because they would know what is being taught at the university level in SA&D courses.

3.3 Ethics Approval

Ethical behaviour is defined as "a set of moral principles, rules, or standards governing a person or profession" (Lichtman, 2012). The major principles of conducting a research with human participants is that they are not harmed. The participation must be voluntary, and participants can withdraw from the research any time without giving any reasons. It is important that interviewee must avoid any inappropriate behaviour; Furthermore, data must be interpreted honestly without altering the information.

To recruits participants Auckland University of Technology Ethics Committee (AUTEC) offers clear guidelines. Norms are provided for conducting interviews, anonymity, confidentiality, integrity pertaining to participants and the information provided, obtaining their consent and designing indicative research questions (Kanter, 2009). Thus it is important to follow certain rules to keep the researcher and the participants' safe and guard their own interests.

AUTEC ethically approved this present study on 9th June, 2015, via reference number 15/166. This approval can be found in Appendix 3.

3.3.1 Obtaining Informed Consent

Any research that involve human participants requires voluntary participation and hence requires consent from the participation. Informed consent is the process by which researchers describe their research project describe their project to potential participants and obtain their consent to take part in the project. The researcher must inform participants of the research background, methods, and goals of the project (Shahnazarian, Hagemann, Aburto & Rose, 2013).

As per guidelines provided by Auckland University of Technology (AUT), an informed consent must be obtained so that participants are aware of the research project, it's methods and objectives, anonymity of their participation, privacy and integrity of

the information participants will provide (AUT, 2014).

Based on the template from AUT, to familiarise participants with the research, an information sheet was created. It contained the information regarding the scope of the research, how it would be conducted, and what would be the expected outcome.

This information sheet was emailed to all the participants. If they agreed to participate, the next step was to get their consent by having them sign the consent form. The information sheet can be found in Appendix A and the consent form can be found in Appendix B.

3.4 Data Collection Methods

The outcome of this present study is based on qualitative data. It is non-numeric data and is collected from a sample of population that satisfies the criteria of the research in question (Myers, 1995). Qualitatively data can be collected by a number of methods such as observations, focus groups, questionnaires and interviews. However, due to the scope of this research focus group, questionnaires and interviews were utilized.

- **Focus Group**: It uses group dynamics to generate qualitative data (Gill, Stewart, Treasure & Chadwick, 2008). Since participants were from different organisations, this method was not considered appropriate for this study because participants may not have felt comfortable sharing their organisations' practices in front of others. Moreover, it would be difficult and time-consuming to arrange a session that was suitable for all of the participants (e.g., scheduling).

- **Questionnaire**: With the potential for large sample sizes, questionnaires make it easier for the researcher to generalize their findings from the sample to the target population (Kelley, Clark, Brown & Sitzia, 2003). However as the focus of this present study is exploratory in nature, a questionnaire was deemed not

appropriate. In addition questionnaires have limitations such as the information does not explain why things are the way they are, the information asked and given both are limited, and the questions formed are based on the perception of the researcher; the responses are based on perceptions of the participants. Finally there may be a problem with interpreting the questions on the part of participants.This will affect the analysis (Munn & Drever, 1990).

- **Interviews:** Interviews are the most common data collection method used in qualitative research and can be used to explore the views, experiences, beliefs and motivations of individual participants (Gill et al., 2008). It allows the participants to provide more detailed responses, which are invaluable for research like the present study. Since the aim of the present study is to map the requirements of the software industry and compare them with the the SA&D courses taught by six universities, semi-structured interviews were considered the best option. This is because they offer insight into how behaviours, systems & relationships change, or are maintained and it is possible to understand how organisations function (Harrell & Bradley, 2009).

3.4.1 Rationale and Purpose

The present study used semi-structured interviews of academics at six New Zealand universities who are involved with SA&D courses, and ten participants from New Zealand's software industry. Semi-structured interviews were the preferred data collection method because they allow the interviewer to be flexible with the questions and ask on-the-fly questions (Myers, 2013). Secondly, the open-ended questions, wherever required, help to elicit information from the participants that would otherwise not be offered by them, or which may not have been researched in the past (Gill et al., 2008).

The purpose of the present study was to investigate the content taught in SA&D

courses, and professional skills and tasks the software industry employs in terms of system analysis and design. Also, it was hoped the research would illustrate a deeper understanding of the opinions, individual experiences, social practices and norms in the New Zealand's software industry. Conducting this study gave me an opportunity to ask questions pertaining to any existing methodologies or tools and techniques to gain an insight into the prevailing practices in the two fields (i.e. academia and the software industry). Ultimately it helped me in understanding the contents of the SA&D courses taught at the six universities, the methodologies, tools and techniques used in the software industry, and comparing the SA&D course contents with the requirement of the software industry

3.4.2 Research Protocol

Research Protocols (RP) are used to guide the researcher in prioritising the data collected from each interview (Harrell & Bradley, 2009). According to the authors, when conducting interviews, it is desirable to have an interview guide or research protocol in place because it helps a researcher to think of all the rules such as length of time, assurance to safeguard information related to the participants, data obtained, and how it will be reported. It helps both interviewee and interviewer to communicate comfortably. The participant does not feel obligated to answer questions they don't feel comfortable with.

The present study followed Goffman's 'The Dramaturgical Model', based on a theory about face-to-face conversation (Myers & Newman, 2007), the individual interviews have a drama concept such as a stage(venue for the interviews), actors and audience (interviewers and interviewees), scripts(interview questions), performance (how the interview is conducted) and entry and exit(the start and end of the interview)[2].

[2]The Dramaturgical Model' description can be found in appendix D

The RP for the present study was adapted from a questionnaire instrument Powell and Yager (2013) had used in their study. This helped me to form a check-list to cover all of the SA&D topics discussed in Chapter 2. The RP used in the present study also had the instructions for an interviewer to greet the participants and re-introduce the purpose of the interview before the participants handed over their consent form.

3.5 Interview Questions

The interview questions were adapted from the studies conducted by Powell and Yager (2013), and Tastle and Russell (Tastle & Russell, 2003) who had already explored SA&D content and obtained the anticipated results. Nonetheless these questions were modified to reflect the context of New Zealand. The questions were organised into specific categories such as methodology, requirement elicitation, modelling, tools and techniques.[3].

3.5.1 Constructing Questions

Since the participants were from different fields, two sets of questions were prepared. One was dedicated to the practitioners involved in SA&D and the other was for academics involved in teaching an SA&D course at the five universities [4]eaching staff from my university were not interviewed, instead the course descriptor was used to get insight into the course. Most of the questions were closed and therefore there was little room for misinterpretation by myself (Bachman & Schutt, 2016). However some questions were open ended so the participants' responses could be explored further.

While framing questions, attention was paid to keeping the language simple and concise, keeping questions short, avoiding repetition, and ensuring the interview lasted

[3]Appendix E contains the questions for both the universities and industry participants
[4]t

between 60 - 90 minutes (Planing, 2014).

To start with, the opening questions were aimed at collecting demographic information and were related to the participant's role, experience, and size of the organisation the participant worked in. The next were specific questions related to methodologies, tools,and techniques. These were asked in same hierarchy; this approach is known as the funnel approach commonly used among researchers (Grover & Vriens, 2006). It is used to keep participants engaged and express their thoughts without feeling pressured. It also helps in getting the unanticipated responses and keeps the interviewee motivated without imposing any burden on them (Wengraf, 2001).

The on-line course descriptors for the SA&D courses were downloaded from the New Zealand universities' websites (where available) to examine what the academics were teaching and proved to be very useful in constructing and framing the questions. Based on the topic information, they were tailored as per the particular university's teaching of the SA&D course.

Since not enough information was known about the SA&D practices in the New Zealand's software industry, the questions were framed to the best of my understanding and knowledge.

In addition to the above resources, some New Zealand studies gave an insight of the software development skills and practices utilised in the industry. [5].

3.5.2 Pilot Interview Questions

A pilot study helps a researcher to discard the unambiguous questions, determine the best length of questions, discover what types of responses and the time taken to answer the questions, and helps in testing the validity of the instruments used. Additionally, the pilot study helps the researcher determine feasibility of a (fullscale) study, designing

[5]The following studies were helpful to understand the software practices in New Zealand: (Kemp et al., 2003; Talbot & Connor, 2011; Kirk & Tempero, 2012)

a research protocol, collecting preliminary data and training a researcher in as many elements of the research process as possible (Teijlingen & Hundley, 2002).

A Pilot study was undertaken, with two academics from my university, and two industry participants who had domain knowledge. The pilot study allowed me to ensure the feasibility and validity of the questions to receive feedback on the questions and interview procedures, to reframe ambiguous questions and remove repetitive questions, and to gain experience in probing further depending on the participants' responses.

3.6 Selection of Participants

In order to carry out the present study, it was important to identify an appropriate audience (i.e., potential participants who had the domain knowledge and expertise of SA&D). The following criteria were taken into consideration for identifying and selecting the participants.

3.6.1 Criteria for Academic Participants

A teaching team who had at least three years' experience in designing or delivering (or both) an SA&D course were considered. To facilitate this, a web search with the keywords: "System Analysis and Design", "System Modelling", "System Analysis, modelling and Analysis" was conducted with results restricted to New Zealand universities. Out of eight, five New Zealand universities were interviewed in this study. They were contacted using the contact information available on the course websites.

3.6.2 Criteria for Industry Participants

An internet search using the keyword "software industry in NZ" was conducted. According to IndexNZ.com there are 100 business management software industries/businesses,

whereas the New Zealand Software Association (NZSA) showed about 35 members. I had joined many professional social groups such as LinkedIn, Meetup group for Business Analysts, and an IIBA chapter in order to establish contacts with the industry. This helped in spreading the word about the present study and identifying potential participants. The criteria were similar to academics, i.e., three years' experience in a role in SA&D such as System Analyst and Business Analyst.

Emails from the website and social groups were obtained. Where email was not available, phone calls were made to get the contacts. Some of the companies provided direct contacts of a potential participant. Twenty-five invitations for participation were emailed. Out of those, ten responded and participated in this research.

3.6.3 Invite to Selected Participants

An information sheet along with some answers to frequently asked questions were emailed to all potential participants.

As mentioned in Section 3.3.1, it is a must to obtain consent from the participants. When a time was organised for an interview, a consent form was presented before going ahead with the interview. The participants were given time to read and then give their permission by signing the consent form. An information sheet was also presented to re-acquaint them with the research.

3.6.4 Summary Information on Interviews

In total, five universities academics ([6]) and ten software industry professionals participated in this present study. Out of these, three interviews were conducted outside Auckland, five were held at the participants' locations in Auckland. The remaining were conducted at the my university (Auckland University of Technology) - six at the

[6] no academics from my university was interviewed in this study.

City campus and one at the South campus.

The duration of the interview ranged from sixty minutes to ninety minutes. Ten minutes were kept for greetings, signing the consent form and for any questions the participants might have.

The summary of the interviews is presented in Table 3.1.

Table 3.1: Information on Interviews

# of Semi-structured Interviews	15
Time/Interview	60 Minutes - 90 Minutes
Time for informal discussion	10 Minutes (Referring to the Information sheet & signing the consent form)
Total Time for Interviews	16 Hours
Interviews conducted outside Auckland	3
Interviews conducted outside AUT in Auckland	5
Interviews conducted at the south Campus	1
Interviews conducted at the city Campus	6

3.7 Data Analysis Methodology

The data was analysed using the Content Analysis technique. It uses a set of procedures to make valid references from text by comparing the 'level' of communication; identifying the intentions and the characteristics of the communicators; reflecting cultural patterns of groups, institutions or societies; revealing the focus of individual, group, institutional or societal attention and describing the trends in communications by using a set of procedures(Weber, 1990).

Content Analysis also aims to provide knowledge and understanding of the phenomena being investigated (Hsieh & Shannon, 2005). Based on how and when coding and themes can be developed, it can be classified into three main categories:

1. Conventional Content Analysis (CCA): This approach is also known as the inductive approach. A conventional analysis is used where knowledge about a phenomenon is limited. The coding is performed during analysis.
2. Summative Content Analysis (SCA): A Summative Content Analysis is used where the context is not clear. The keywords are treated as codes. These are then counted and compared to interpret and understand the meaning of the latent context.
3. Directed Content Analysis (DCA): It is also known as deductive content analysis. It helps in testing the existing theories at different intervals or in different situations (Hyde, 2000). It is also used to test the previous hypothesis or to investigate further.

These categories are summarised in Table 3.2 below.

Table 3.2: Approaches to Content Analysis (as adapted from (Hsieh & Shannon, 2005))

Type	**When Applied**	**Time of generating codes**
CCA	Limited Knowledge	Codes derived during Analysis
DCA	Explore the already known knowledge	Pre-determined codes
SCA	Applied to the complex data	Both pre-determined and post determined after data analysis

Since an SA&D course has already been investigated overseas (e.g., (Powell & Yager, 2013; Harris et al., 2006; Tastle & Russell, 2003)), DCA was deemed an appropriate analysis method for the purposes of present study. This is because it would help to add to the body of knowledge by mapping the results of this research with the previously-completed studies. Furthermore, the research questions were not highly open-ended and the codes were predefined [7]

[7] Please see Section 3.7.4

3.7.1 Data Analysis Framework

The aim of analysing the collected data was to investigate a typical SA&D course and compare it with the SA&D practices in the software industry to examine any gap. In order to aid data analysis, all interviews were recorded and subsequently transcribed, proofread, and annotated by myself. Codes were based on the pre-defined questionnaire categories. Analysis was involved by putting the interview data into the appropriate node[8].

The graphical presentation of this framework is shown in Figure 3.1.

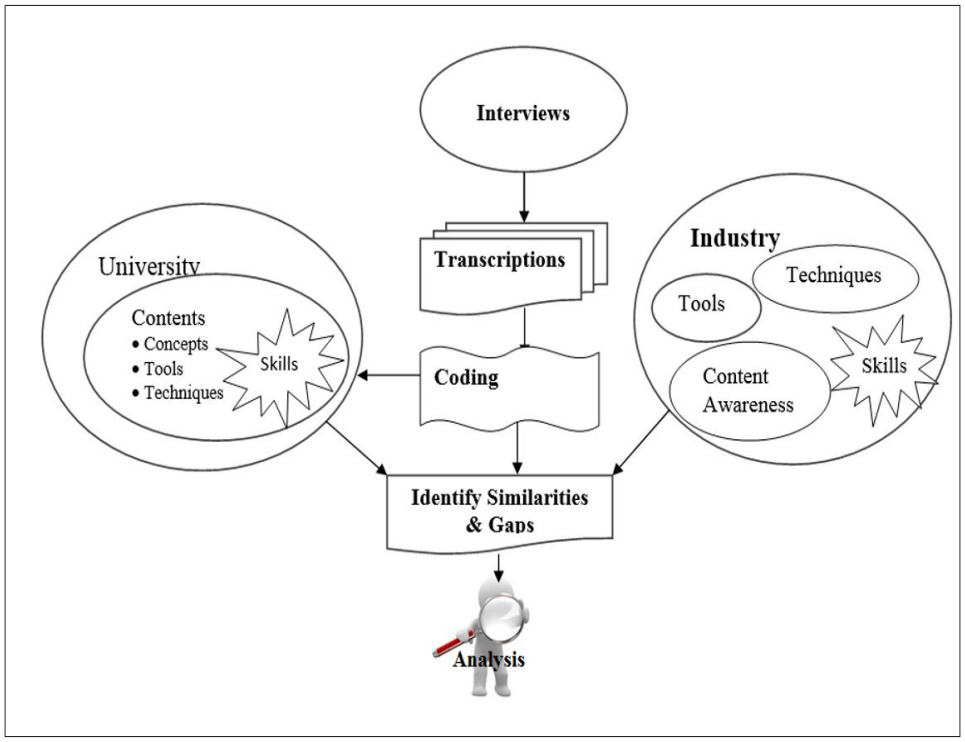


Figure 3.1: *Research Framework*

[8]Information on coding is available in section 3.7.4.1

3.7.2 Data Transcription

Before using the data, interviews were transcribed using Express Scribe Transcription Software. Utmost care was taken so that the identities of the participants were not compromised. Spell check was run for each completed transcription and the transcript was then emailed to the participants for their input. It also gave them an opportunity to reflect on their decision to participate.

The process of transcription is described in Table 3.3 below.

Table 3.3: Process of Interview Transcription

Step No.	Description
Step#1	Interviews were typed while listening to the recording
Step#2	While transcribing, care was taken that no text should identify the participants.
Step#3	Irrelevant information such as 'umm',and 'aha' were not transcribed
Step#4	Spell check was run to ensure that there were no grammatical errors.

The transcribed interviews were converted into an MS-Word and emailed to the participants for them to check the content for accuracy and also to give them an opportunity to reflect on their decision about whether the interview could be used for the present study. Most participants consented to the use of the their interviews.

3.7.3 Data Cleaning

Data needs cleaning before it can be used for analysis (Hellerstein, 2008). This is because a reliable, trustworthy data can contribute to generate a well grounded hypothesis.

Appropriate questions on SA&D contents such as methodology, requirement elicitation, tools and techniques were formed. Information related to these topics from each interviews was put under relevant heading. Interview questions were cleaned to have the

consistency throughout and data was grouped according to topic, such as methodologies covered under a SA&D course, and tools and techniques used or taught.

3.7.4 Coding

Coding is known to be a key part of qualitative data analysis; some researchers even consider coding to be synonymous with analysis and part of the interpretive process of moving from data to idea, and from the idea to all the relevant data (Wit, 2013). Coding helps to condense the data by breaking it into themes and reorganize into categories to answer the research questions (Forman & Damschroder, 2008)

The present study had predefined codes that were related to the SA&D concepts such as methodology used, requirement elicitation, tools, techniques, modelling and skills. This was done first by reading the data a few times before cleaning so that it remained uniform and consistent throughout.

3.7.4.1 Electronic Coding

Data was coded electronically using version 11 of NVivo. It is a Computer Assisted Qualitative Data Analysis Software (CAQDAS); it allows one to capture new concepts or themes, by deriving the nodes directly from the text to be coded and stored for later consideration (Gerbic & Stacey, 2005). It is a powerful tool that helps in literature reviews, word searches, graphical representations, keyword research and many more area. In the present study, the transcribed files were uploaded to NVivo. A list of pre-defined codes was created as nodes (These nodes were the course content areas such as methodologies, tools, and techniques) and are shown in Figure 3.2 below.

NVivo has a feature creating a visualization of data that helps to highlight important words or phrases that might be of interest to the researcher. Figure 3.3 (known as word cloud) shows highlighted words such as use case, diagrams, methodology. Such

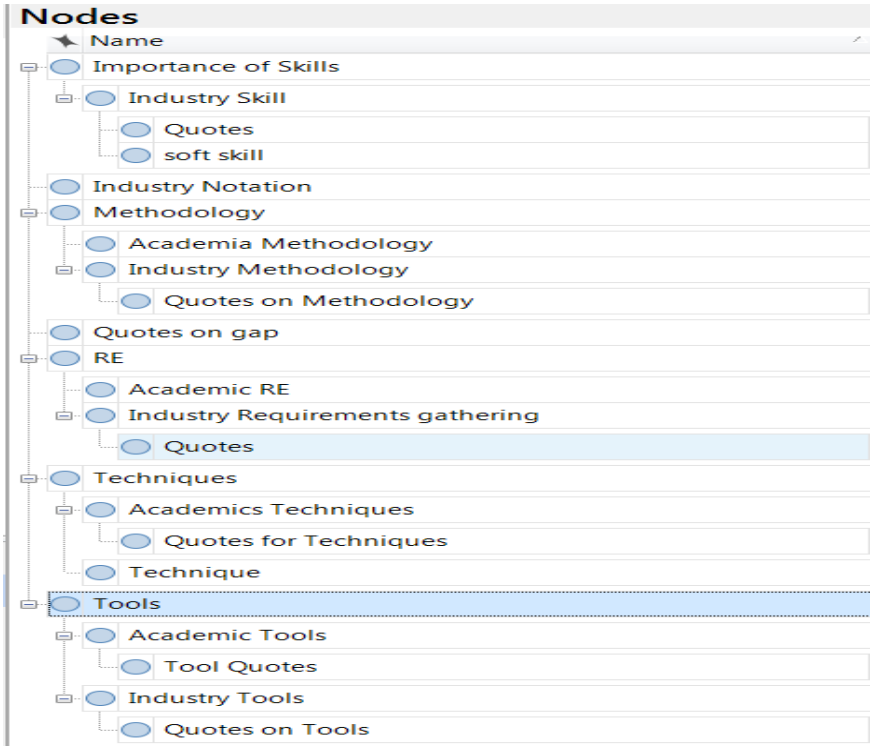


Figure 3.2: *NVivo output for Nodes*

prominent visualization helped me to look for these words in each transcripts to start with the analysis.

The text search feature of NVivo was also used to see the context in which a particular word was used. Figure 3.4 occurrence of Visio was tested. This visualization tries to match word to the context.

Similarly in Figure 3.5 occurrence of BPMN is shown.

Information related to a code across all the transcripts was highlighted by using the drag-and-drop feature to place the information into its respective node. This helped to condense the information, e.g., information related to 'requirements' was grouped, and information about tools was grouped together. This helped with analysing because coding made data concise and grouping saved time during the analysis.

Figure 3.3: *NVivo output for Word Cloud*

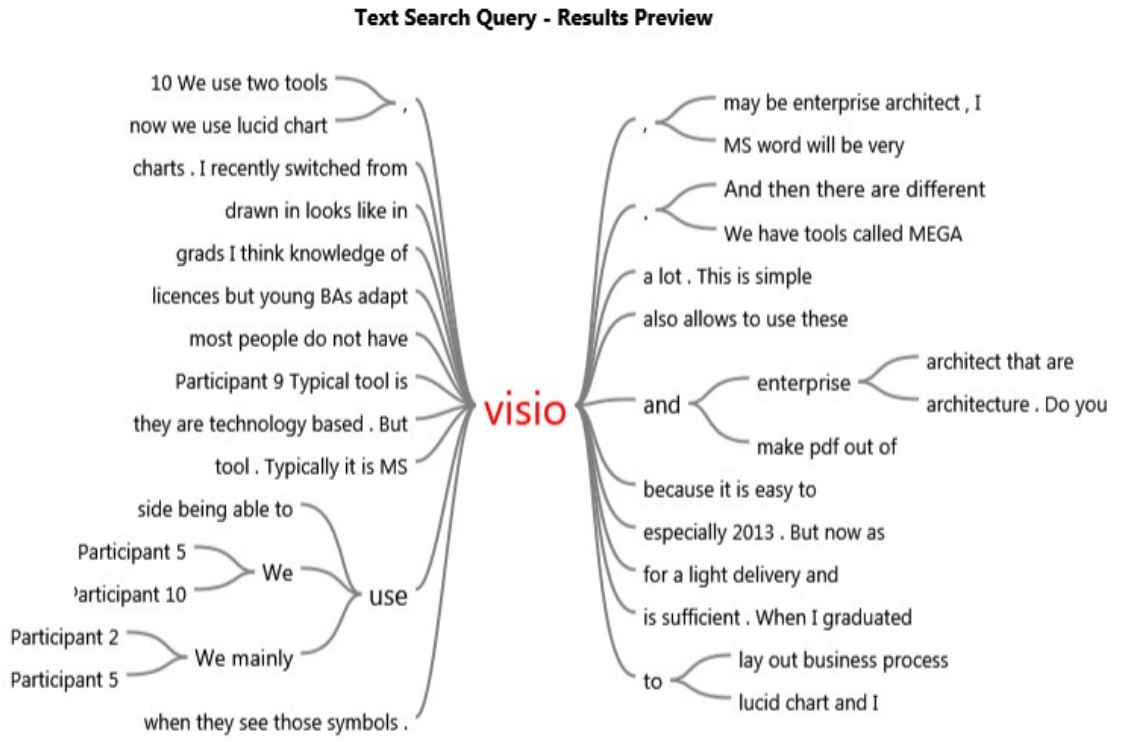
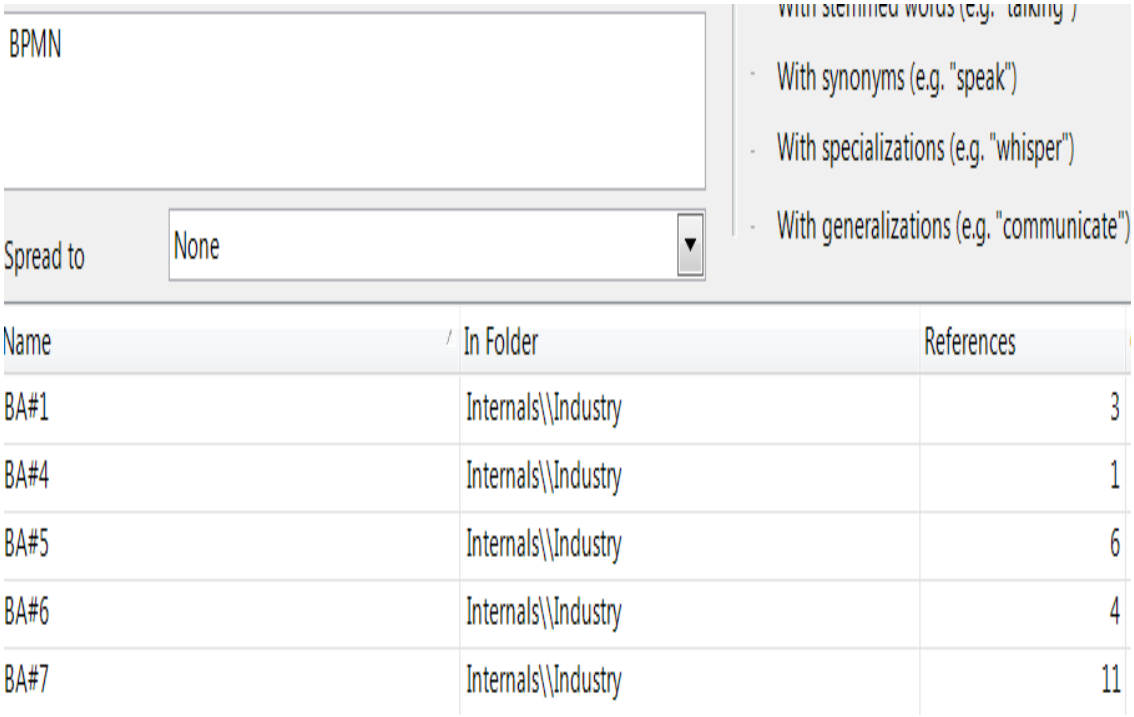


Figure 3.4: *NVivo output for Word Search for 'UML'*



Name	In Folder	References
BA#1	Internals\\Industry	3
BA#4	Internals\\Industry	1
BA#5	Internals\\Industry	6
BA#6	Internals\\Industry	4
BA#7	Internals\\Industry	11

Figure 3.5: *NVivo output for searching BPMN*

3.8 Summary

This chapter discussed the methodology, the data collection methods, and how the semi-structured interviews were conducted. Ethics approval was gained prior to data collection. This chapter also explained the procedure for recruiting participants and getting their consent for participation. Data was analysed using Directive content analysis. Coding was electronic performed using NVivO version 11.

The findings, analysis and discussion will be discussed in next chapter.

Chapter 4

Findings, Analysis, & Discussion

This chapter will focus on the findings of the present study which investigated the typical SA&D course and practices and highlight the gap, if any, between academia and the software industry in New Zealand. It is divided into three Sections, Section 4.1 will given an overview of the SA&D course content, followed by Section 4.2 that will discuss the findings from academia and the software industry of New Zealand and lastly Section 4.3 will focus on the analysis and discussion.

4.1 Overview of SA&D Course content

SA&D is an immense course but does not have a clearly defined curriculum. Therefore, ACM and IEEE have recommended topics that academia could choose from[1]. The main focus of teaching this course is to deliver content in such a way that students understand the concepts of SA&D. However, to teach the course to students from different majors, programmes or disciplines, it becomes difficult to decide which content should be dropped.

Powell and Yager (2013) investigated the important skills, methodologies, tools and

[1]Please see Section 2.2.1

techniques that new hires or recent graduates should have. The present study is an attempt to add value to this body of knowledge by investigating the gap between the skills, practices, tools and techniques between academia and the software industry in the New Zealand context.

Below is a brief description of these topics.

Methodologies: A methodology is a list of steps and deliverables that help to implement a system (Dennis et al., 2015). As discussed in Section 2.2.2, a number of methodologies are taught in an SA&D course, however, for the scope of this research, three methodologies investigated [2] were structured, object-oriented and Agile.

Requirements Elicitation: As discussed in Chapter 2, the success of any system depends on how well the requirements are gathered, analysed, and documented. Thus, as discussed in 2.2.1.1, RAS is a recommended topic for the course as it caters to the requirements of stakeholders, starting from requirements elicitation to analysis, thereby creating a document which is crucial for a system design[3]. In the present study importance of requirements elicitations as an SA&D technique was evaluated.

Tools: As discussed in Chapter 2, the software industry prefers tools because they reduce software errors (Baram & Steinberg, 1989) and improves the quality of documentation and productivity (Dasgupta et al., 2007)[5].

Many researchers such as Tastle and Russell (2003), Powell and Yager (2013) and Senapathi (2005) have investigated the relevance of CASE tools to SA&D. In the present study, the importance of teaching tools in an SA&D course was examined and was evaluated as to the usefulness to the software industry.

Techniques: Techniques are recommended because they help to produce better artefacts[6]. It has been observed that SA&D books have a chapter dedicated to techniques

[2]For details please refer to Section 2.1.1

[3]As discussed in Section 2.2.1.1, SA&D can be grouped into three parts such as SMA, SD and RAS.[4]

[5]Please also section 2.2.6

[6]Please see Section 2.2.8

that may give an overview or discuss the topic in detail (e.g.,(Dennis & Wixom, 2002; Dennis et al., 2009; Rosenblatt, 2014; Dennis et al., 2015)). Researchers have explored the different techniques that are taught in a typical SA&D course and the time spent on them (e.g.,(Bataveljic et al., 2006; Harris et al., 2006; Powell & Yager, 2013; Guidry & Stevens, 2014)). The present study explored the techniques academics taught, and if these techniques were also used in industry.

Skills: To analyse the present system and a potential system, it is important to understand the requirements of the stakeholders clearly(Dennis et al., 2015). For this purpose, it is important to have many skills such as effective verbal and written communication plus be able to conduct interviews and run workshops(Powell & Yager, 2013; Banerjee & Lin, 2006; Radermacher et al., 2014). In addition to these skills, use of tools and techniques in producing good documentation, and non-technical skills such as working in teams, are needed for the success of a project[7].

The present study explored the role of SA&D course in acquiring such skills and the skills industry professionals expect graduates to have.

The following sections present the investigation of these topics in both academia and the software industry.

4.2 Findings

This section will focus on the participants and their input from both fields (Universities and the Software Industry).

4.2.1 Participant Information

Fifteen participants were interviewed for the present study. Out of these, five were academics and ten were from the software industry. The following section will provide

[7]For details please see Section 2.2.9

their demographic information.

4.2.1.1 Participants from Industry

Nearly 70% of the participants were from an SA&D-dominated field and were either system analysts or business analysts. They represent both the public and private sectors. Though only 40% were from the public sector, they were able to provide some valuable insights on SA&D practices in New Zealand software industry because they had high levels of experience in roles related to SA&D.

The project managers who participated in this present study, were either involved in hiring system analysts or business analysts or had a strong involvement in systems development projects either as project managers or professional consultants.

Table 4.1 shows information about these participants.

Table 4.1: Demographic Information of Industry Participants

Participant#	Role		Organisation	Experience
Participant #1	Business Analyst/Consultant	BA#1	Private Sector	8+
Participant #2	Business Analyst/System Analyst	BA#2	Public Sector	12+
Participant #3	Business Analyst/System Analyst	BA#3	Private Sector	10+
Participant #4	Project Manager	PM#1	Private Sector	3
Participant #5	Business Analysts/System Analyst	BA#4	Private Sector	20+
Participant #6	Project Manager	PM#2	Private Sector	5
Participant #7	Business Analyst Manager	BA#5	Public Sector	>10
Participant#8	Business Analyst/System Analyst	BA#6	Public Sector	10
Participant #9	Practice/Project Manager	PM#3	Public Sector	8
Participant #10	Principle BA Consultant	BA#7	Private Sector	20+

Before moving on to specific roles related to SA&D participants started their careers as either developers or were in business support roles. BA #1 and BA #3 are business owners. Prior to owning a business, BA# 1 had worked in the public sector, whereas BA# 3 had worked for a well-known private sector organisation in New Zealand.

4.2.1.2 Participants from Academia

Participants from academia were involved in teaching an undergraduate SA&D course at their respective universities. At the time of the interviews, they were holding more than one role; anything from putting the course together to delivering it. Most academics had been involved in the SA&D course since their university had first offered the course, while some joined as part of the teaching team.

Table 4.2 below shows information about the academic participants.

Table 4.2: Demographic Information of University Participants' roles

Participant#	Role	Experience
Academic#1	Tutor	13+
Academic# 2	Course-Administrator/ Co-Teacher/ Course Co-ordinator	3
Academic# 3	Lecturer/ Course Co-ordinator	11
Academic# 4	Course Co-ordinator Lecturer	18+
Academic# 5	Course Co-ordinator/ Lecturer	20+

The academics, who were teaching this course were able to provide insight into how the course content had changed over time.

4.2.2 Findings from Academia and Industry

In this Section, research sub-questions from Section 1.4 will be revisited and related findings from academia and industry will be presented.

4.2.2.1 Research Question #1

- **Under what major is an SA&D course is taught in the New Zealand universities and what is the duration of these courses?**

Predominantly, in New Zealand universities this course is taught within the business faculty as an IS major. However, two universities teach it under an IT major and is normally taught as a one-semester course, in some universities it is offered over a period of two semesters [8].

Table 4.3 shows the SA&D course information at various universities.

Table 4.3: Course Info as per programmeme /Major

Uni#	Name of the Course	programmeme/ Major	School/ Faculty	Duration of The Course
1.	Data & Process Modelling	Information Technology	School of Engg Comp. & Math.Science	1 Semester
2.	Business System & Analysis	Information Systems	School of Business	1 Semester
3.	System Analysis & Modelling	Information Technology	College of Science	1 Semester
4.	System Analysis	Information Systems	School of Business	1 Semester
5.	Business System Analysis	Information Systems	School of Info Mgmt part of Commerce Faculty	1 Semester
6.	System Analysis Design & Modelling	Information Systems	School of Business	1 Semester

An overseas study has shown that an SA&D course is generally taught in one semester (Burns, 2011), and may be also be taught as two courses such as System Analysis (SA) taught in Semester One and System Design (SD) taught in Semester Two (Holmes et al., 2002). Comparing the results to the findings of the present study most of the New Zealand universities teach this as one course and investigations show that in New Zealand most academia focus on analysis.

[8] University # 3 offers this course through distance learning

4.2.2.2 Research Question #2

- **What methodologies are covered in a typical SA&D course in New Zealand??**

  Findings of the present study show that most New zealand universities teach structured methodology, but two universities teach object-oriented. However, while delivering this topic, academics also give either an overview or they mention other methodologies such as Agile. This is because these are covered in detail in subsequent courses. Only two participants (Academic #4 and Academic #5) mentioned that they have started teaching Agile.

  According to findings, academics spend a maximum of two lectures on teaching methodologies and also give an overview of SDLC. However, Academic#2 spends six to seven lectures on SDLC. This is because they place more emphasis on planning and analysis because according to them, these are the crucial stages of the life cycle of a project. Whatever methodology graduates may use in the software industry, they will be spending time in planning and analysis. The idea is to familiarise students with the SDLC stages so that they can benefit from this knowledge and dispense it as and when needed when they join the software industry.

  > *"We teach them SDLC, planning phase, investigation phase, feasibility (that sort of thing) and take them through different phases of the life cycle and reasons that whatever methodology you teach you are still planning, analysing, designing etc" (Academic#2)*

4.2.2.3 Research Question #3

- **What skills and techniques are covered in an SA&D course in New Zealand?**

This is a very important question because it covers the topic that has direct content delivery (e.g., requirements elicitation techniques, modelling techniques) as well as indirect content (skills assessment). This is because it leads to the development of technical skills (by delivery mechanisms) as well as non-technical (by non-delivery mechanism such as focusing on developing course work assignments).

RAS (Requirements Analysis and Specification) [9] highlights the requirements elicitation and the techniques that will turn the requirements into a suitable software system. Since academia uses case studies and scenarios to teach this topics both academics and students find this topic, both academics and students find this topic very challenging. This is because it is difficult to teach and comprehend the requirements of an imaginary client.

In content delivery academics use case studies to teach various concepts such as understanding the difference between functional and non-functional requirements, user and system requirements, gathering and analysing, writing concise and clear requirements using template and semi-formal grammar, and user stories. This present research found that academics choose two different case studies or scenarios for the course; one is used to deliver the course and the other is used to assess the coursework.

To overcome some of the challenges, some academics plays the role of a client and facilitator. At times they assign roles to students groups by introducing a role-play in class. This exercise helps them to understand the difficulties and uncertainties associated with the requirements elicitation. In addition to these, other requirements gathering techniques[10] are taught. Teaching different techniques at universities enable students to use various strategies as and when needed in the workplace. Students are also encouraged to use combined techniques. This may

[9] In Section 2.2.1.1, SA&D was grouped into three different topics (SMA, RAS and SD)
[10] such as interviews, questionnaires, survey, working in teams, document analysis

be because academia may want students to know that gathering requirements is a difficult task and one technique may not be sufficient.

Table 4.4 shows the techniques academics teach to elicit requirements.

Table 4.4: requirements elicitation Techniques Taught in Universities

Number	Name of the Techniques
1.	Concept Map
2.	Document Analysis
3.	Interview
4.	Prototypes
5.	Questionnaire
6.	Role Play
7.	Survey
8.	Working in teams

Once the requirements are gathered and analysed, students are taught to model these requirements using different modelling techniques. Participant #3 stated that they focus on modelling because:

> *"There is a lot of emphasis in the idea of converting the requirements elicitation output into the models. It helps students to understand have they got it, have they got it all correct before going into processes of validating these."*

Academics teach both data and process modelling in the SA&D course.

Academics teach different techniques for data modelling that assist in identifying and establishing a relationship between data elements (entities/objects) and business processes. Thus each model created by different techniques gives a real life experience to students.

> *"We take home aboard the idea that if you are walking in to project that say has length time and that has a set number of different requirements and client knows exactly what they want to compare what they not*

> *want, what sort of techniques would you apply, what sort of elicitation techniques would you apply, what sort of modelling techniques would you have. As sort of why thinking about that one size does not fit at all and you really have to think about with this particular modelling technique what sort of techniques would be useful in this type of project? That's what we intend to teach students to think. Instead of thinking we use OO because it is latest and greatest whereas this may not be the case but they should step back and consider other options" (Academic #3).*

This implies that academics aims to teach the significance of techniques rather than considering it - teaching - a mere topic.

The most popular modelling techniques academics teach are the use case diagrams, and the activity diagrams. However it was found that *"it's very confusing. Use case is a term that they take a week to understand"(Academic #2).*[11]

Depending on the methodologies, academics also teach ERD, class diagrams, Use Case descriptions and sequence diagrams.

Table 4.5 shows all the techniques academics teach in the SA&D courses at the six universities.

Findings from the present study show that some academics encourage students to use a template to document the requirements from the case study or scenarios. This template is taken from the capstone project that students take in the final year of their study.

> *"we are looking around into an idea of creating a standard template on international standards because there are so many templates out*

[11] Please refer to Section 4.3

Table 4.5: Modelling Techniques Taught in Universities in New Zealand

Number	Name of the Modelling Techniques
1.	Entity Relationship Diagram
2.	Class Diagrams/Component Diagrams
3.	Data Flow Diagram
4.	Context Diagrams and Context Map(c-map)
5.	Use Case Description
6.	Use Case
7.	Activity Diagram
8.	Behaviour Diagrams(Sequence and State Machine)

there on how you actually structure your requirements" (Academic#3)

At university #1 academics teach the use of semi-formal grammar in documenting requirements. This topic is given priority in tutorials as well. Students are expected to use this format in documenting the requirements as a part of their class exercise and assignment deliverables. The idea is to encourage students to develop professional documentation experience.

Academics develop a number of tasks and assignments to assess students' understanding of the various concepts they teach in the course. These include individual assignments, or group or both[12]. Group assignments are important in this course and are designed so that students develop soft skills, such as:

1. Communications: This helps students to communicate their ideas and issues within the team to contribute to the progress of the project. This exercise helps them to communicate with stakeholders when students join the industry.

2. Time Management: A deadline for each task will help them to be organised and manage their time.

[12]For the structure of these assignments please see Table F.1 in the appendix section.

3. Critical thinking: On receiving feedback from their lecturer and/or team, they will reflect on their task to improve the work. This will help them to assess their tasks critically.
4. Team building: With cultural diversity students learn to work in a team putting aside their differences.
5. Technical Writing: Using techniques, templates and semi-formal grammar will help them to work with templates and develop suitable professional documentation. This will help them to write technical documents. This is because, according to investigations, templates are used frequently throughout the software industry.

> *"For requirements documentation we provided them with the template and example requirements document. First step would be they would co-author a document with someone and they would go to meetings and write parts of the document and the person who co-authored with them will review the documents. The second documents they would produce themselves" (BA#7)*

By working on assignments, students develop other skills and attributes such as team management, responsibility, flexibility, persistence, problem solving, and decision making.

4.2.2.4 Research Question #4

- **What tools and notations are covered in a typical SA&D course in New Zealand?**

Academics encourage students to develop coursework documentation using tools. Participant #1 thinks it is important to teach tools in this course because using

tools at a workplace is a common practice. Knowing a tool is considered essential because having a working knowledge of a CASE tool will benefit the students when they join the software industry.

However, it was found that there were no specific tools that academics taught for document requirements. There is no use for highly sophisticated tools, as simple applications such as MS-Word or MS-Excel are sufficient to produce such documents.

Yet tools are needed for modelling because models, such as data models, process models and behaviour models, are built using various techniques[13]. For this reason, academics introduce various tools. Some of these tools are SA UML, Rational Rose, Star UML, Visual Paradigm, Aris express, Visual Express, and Visio. Open-source tools such as Lucid chart and UMLet are also suggested for students to try. It was found that most of the universities leave the choice of using a tool up to students; in university #1, sometimes students do not work with tools at all but develop documentation manually.

The tools used for the SA&D course at the six universities are summarised in Table 4.6.

Table 4.6: CASE Tools Taught in New Zealand Universities

Number	Name of Requirements gathering Tool	Name of the Modelling Tools
1.	MS-office	SA UML
2.	MS-Excel	Rational Rose
3.		Star UML
4.		Visual Paradigm
5.		Aris express
6.		Visual Express
7.		Visio
8.		Lucid Chart
9.		UMLet

[13]Detailed description of the CASE tool is available in Section 2.2.6

This present study found that academics tend to demonstrate the working of tool in a tutorial or lab session and then students learn on their own. It is not mandatory for students to use a recommended tool. They may choose a tool that they have used in the past or in a previous course or any tool that they are comfortable with. However, academics encourage students to work consistently with one notation. They do not teach it as a separate topic but devote time in teaching UML notations in detail. For example, while teaching usecase diagrams they spend a great deal of time in teaching the concepts such as:

1. Usecase diagrams: concepts of actor, usecase, extend of a usecase and include a usecase
2. Activity diagrams: concepts of activity, decisions, swim lanes, fork and merge nodes
3. Class diagrams: concept of generalisation, association, aggregation, and composition

This study also found that academics encourage students to be consistent with a notation; most of them teach UML, but it was also found that four academics also use BPMN. Academics #3, #4 and #5 stated that they have incorporated teaching BPMN into their SA&D courses because it is used extensively in the software industry. Another academic recently introduced it because, *"from a lecturer's experience it is used quite a lot in industry"(Academic#1).*

4.2.2.5 Research Question #5

- **What methodologies are used in the software industry in New Zealand?**

  On examining the use of methodology in industry, the present study found that, out of ten participants, four followed structured methodology, while six followed

Agile. There was only one participant who followed object-oriented methodology for a project. Out of ten only two participants had followed a hybrid methodology (combination of structured and Agile), and one out of ten had used object-oriented methodology in the past.

The participants also indicated that they follow structured methodology because of the culture of the industry - participants find it hard to switch over to a different methodology once they have used one particular methodology in the organisation. Secondly, stakeholders keep changing their minds, which makes it difficult to start the project. Therefore, the participants use structured methodology to document their requirements, and get it signed-off so the project cannot be modified. Thirdly large projects prefer structured methodology because of the cost involved in developing and implementing systems Table 4.7 summarises the techniques used by the research participants in the NZ software industry.

Table 4.7: Methodologies Participants from the software Industry used

Structured	Object Oriented	Agile	Hybrid (Structured and Agile)
4	1 (+1 had used in past)	6	2

However, it was noted that choice of methodology also depends on the project. It is a practice to follow structured methodology for large projects but there are projects that are also developed using Agile.

Structured methodology is also used if, *"there is no option because you say your scope, budget and delivery date are all fixed. So you have no flexibility" (PM #3)*. Thus it is also used where stakeholders know what they want.

It was also found that Agile works very well for start-up organisations and for small projects that need to be completed fast. One participant (BA #2) mentioned that though they followed structured methodologies, for a small project they experimented with Agile using SCRUM practices such as Sprint planning, five

day Sprints and retrospectives. They said that, after using Agile for the project, they realised it was useful for projects where quick results were needed.

Yet Agile is not for everyone as projects such as building medical and defence equipment require a structured approach because they are complex and mission-critical. Therefore they need effective planning and a sign-off is a must to ensure that everything is as per the agreed plan.

Sometimes stakeholders determine the use of methodologies. If they make quick decisions, Agile can be adopted, but if they are not able to make a decision or keep changing their mind, structured methodology may be the answer.

In addition, it was found that an organisation could follow any methodology but usually a team used a mix of methodologies for their own understanding. At times, a team with developers may work with the particular methodology used for the project, but also use Objected-oriented (iterative and incremental).

Use of methodology in the New Zealand software industry is project driven. Projects like a website can be built on Agile, but sensitive projects like insurance and banking may need a structured methodology. Also, government or public sector projects are mostly completed using structured methodologies. This is because of the involvement of finance and the third parties, everything needs to be transparent, and every stakeholder needs to know about the progress of the project. Since the private sector is flexible and easily approachable, they are moving to Agile irrespective of the project they have. *"I found it's much easier to get things done here if it's a good idea it's just get done because everybody sees that it's a good idea and it's done"(BA#7).*

4.2.2.6 Research Question #6

- **What skills and techniques are needed in the software industry in New Zealand?**

  Industry expects graduates to collect and draft requirements because they consider that requirements are very important in SA&D. Nearly all of the participants, other than project managers, were directly involved in either requirements elicitation, analysing or tracking requirements. Sometimes they depend on external agencies to support them with the process of developing requirement specifications. Thus the participants from the software industry consider it to be the backbone of the project. When graduates join the software industry, participants said that they seek involvement of the graduates in assisting the senior team working on a project to collect and draft requirements. They use different techniques because it helps them to understand all aspects of the business: *"We take diagrams to client, this is a very interesting phase, and the clients may at times find the blind spots that point to the right directions"(BA #6).*

  This present study found that the software industry uses different techniques for requirement gathering. The common techniques are workshops, interviews, meeting minutes, any available diagrams, research, questionnaires, user stories, note-taking, telephone conversation, emails and interviews.

  Table 4.8 shows the requirements elicitation techniques used in the industry.

  Apart from the techniques, shown in Table 4.8, the industry uses specific templates such as Software Requirements Specification (SRS).

  After gathering requirements, the next step is to analyse the requirements so that a system feasibility report can be created. Once analysed and documented, the team meets with different stakeholders to check that everyone understands and

Table 4.8: requirements elicitation Techniques used In the NZ Software Industry

S.No.	Name of the Techniques	Count
1.	Brain Storming	1
2.	Diagrams	4
3.	Emails	2
4.	Interview	4
5.	Meeting Minutes	2
6.	Notes Taking	1
7.	Phones	2
8.	Process Mapping	1
9.	Prototype	1
10.	Questionnaire	2
11.	Research	1
12.	Use Case Descriptions	2
13.	User stories	2
14.	Workshops	2

agrees with what is being suggested. Once requirements are analysed and signed off, they are then modelled into data and processes. However not all projects need data modelling. Sometimes they model data during development (BA #4) and sometimes participants model for their own understanding (BA #5).

> *"It is important for me to understand how the system works, how different users are working in relation to each other. What are the attributes that are hidden and might evolve during the operation?"(BA#4).*

This may mean that data models sometimes may remain within the teams and are not handed to the stakeholders.

Different techniques are used to create different models. ERD and class diagrams are created to model the data (who is involved and what is the relationship); Context diagrams, use case diagrams, activity diagrams, sequence diagrams, and workflows are used to model the processes (how a system will behave, how entities will interact with each other and with the system, and how a system will grow)

Table 4.9 shows a list of the modelling techniques participants use in the software industry.

Table 4.9: Modelling Techniques Used In the Industry

No.	Models	Techniques	Count
1.	Data Models	ERD	3
		Class diagram	1
2.	Process Models	Data flow diagram	1
3.	Behaviour Models	Context diagram	4
		Use Cases	4
		Use Case Descriptions	2
		Activity/Process/workflow diagram	8
		Sequence diagram	1
		State diagram	1

The present study found that the industry realises that graduates cannot have expertise of all of these techniques when they join the software industry. However, when graduates join the industry, they should have knowledge of what they studied at university, such as familiarity with techniques for requirements gathering, understanding and working with templates, interpreting the diagrams, using the knowledge and understanding of notations, working knowledge of important tools and interpreting what stakeholders are saying and interpreting diagrams (BA #3 and #9).

> "*The methodology, tools and techniques and skills we talked about can interchange but the way you use, you communicate with people depending on the environment, how you present that would be different. If you are a graduate, they will give you what you learnt to succeed. You learnt what you learnt which is a key."(BA #6)*

For a SA&D role industry participants find soft skills equally important:

> *"With BA work lots of soft skills are required. Good communication skills, critical thinking, and able to use your understanding of*

> *the project into a flow diagram to explain to your client. You don't need fancy stuff. Then they follow you. But if you are working with developers then you need your user stories, it depends on who are your audience but good communication pays everyone." (BA#6)*

Also in an SA&D role, it is all about working with clients, and the the industry participants think that communication- both verbal and written- is very important. Also persistence is a vital skill:

> *"that depends on what your stakeholders want or I should say how well they are able to give information to you. Here comes the actual art of a BA. You should know how to bring them back to the point where you can understand their core requirements. They take you to different world and if you are lost then you will not be able to understand their needs" (BA#4).*

This is because stakeholders can only give information they think is important. Therefore the BA should dig deep to get any problems to the surface. This requires listening patiently, asking questions in different ways, and confirming the requirements by narrating them in a language that stakeholders can understand. It is important to understand and clearly identify the requirements, so that these can be documented.

Other skills that the industry expects from IS or IT graduates, are project management skills including an aptitude for learning, ability to organise meetings, and critical thinking. Most of these skills help in the analysis phase. Other skills and attributes valued by the industry are being eager to learn and curious, problem solving, flexible in approach, and self-motivated; being able to self- evaluate so that they can improve themselves. Since business is part of a multi-cultural

society, graduates should be able to fit into any culture.

Another vital requirement of graduates, is to consider the hierarchy of the client organisation. Since it consists of a number of stakeholders who could influence the project, it is important to know the key stakeholders and communicate with them accordingly.

Two participants (BA #2 and BA #3) suggested that IT/IS graduates should undertake a course in Psychology:

> *"That is very important. Because different people are participating and you should be able to get maximum information from them. When you are gathering requirements, you are dealing with so many people and you should be able to tackle them to get the right information out."(BA#3).*

This means it is important to win the trust of stakeholders so that they are comfortable in sharing their experiences and objectives.

Another participant (BA#2) suggested that graduates should have some kind of entrepreneurial training as it can help in understanding the laws, economy and culture of the company. Such training may help graduates to work well with their co-workers, learn time management, and work well under pressure and improve their productivity. This can be achieved if the graduates have soft-skills when they join the industry.

4.2.2.7 Research Question #7

- **What tools and notations are practised in the software industry in New Zealand?**

The participants use different tools for requirements elicitations. The idea of using a tool is to document requirements using different techniques. Clearly documented requirements help to keep the team informed of their progress and the project can evolve accordingly.

According to the participants, use of a tool is also important for tracking the requirements as they are documented. It is easier for the team to revisit them, check for any modifications, make alterations and sign off on completion throughout the SDLC stages. Common applications such as Microsoft Excel and other Microsoft Word are used for documenting and tracking requirements. Investigations show that for requirements elicitations, participants also take notes using a pen and paper.

Table 4.10 shows the tools for Requirements Elicitation in the New Zealand Software Industry

Table 4.10: Tools Used for requirements elicitation in New Zealand Software Industry

Name of the Tool	**# Participants Used**
Enterprise Architectural Tools	5
MS-office Tools	4
MS-Visio	5
Paint	2
Trello	2
Valerie	1
AHA	1
JIRA	1
MEGA	1

The findings from the present study show that three popular tools, Microsoft-Office tools, Enterprise Architecture Tool and MS-Visio, are the most commonly used in the requirements elicitation; also as presented in Figure 4.1, participants who used MS-Office tools and Enterprise tools, also used Visio.

"We use two tools, Visio for a light delivery and for more technical

and details we use Enterprise Architect extensively specifically where we have to show different models" (PM#3).

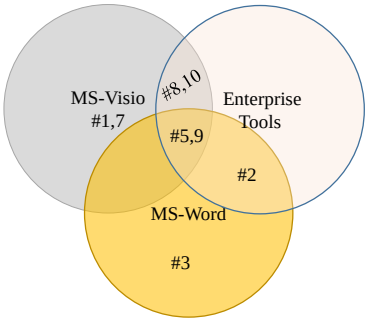


Figure 4.1: Distribution of three common tools used for requirements elicitation in NZ Software Industry

It was found that industry uses the following tools to show different perspectives of the system, such as data, process and behaviour.

1. Bizagi is a free BPMN Compliant tool and allows users to publish work as a document in both Word and PDF.
2. Biztalk is a Microsoft tool that allows automation of business processes
3. BlueWorks Live is an IBM BPMN compliant tool
4. Lucid chart is an open source web-based diagrammatic tool that allows users to create UML diagrams
5. Enterprise Architecture Tools is a modelling and building applications tool that can be used for the organisation's technical needs
6. MS-Visio is a Microsoft diagrammatic tool. It can be used for most diagrams such as UML and process.

This present study found that it is more important to have awareness of some of the most popular tools used in the industry, such as Microsoft tools. If graduates have familiarity and a basic working knowledge of some of these popular tools, then it becomes easier to adapt their learning to any similar tool that they might

encounter in the workplace. This implies that they don't need to gain mastery in a specific proprietary tool.

> *"So if someone is mentioning 'Rose', you know that it is not a flower but a tool. So familiarity and knowledge of a tool is important. It may be Visio or any. You should not be a master of it, but its good and if you go to industry you learn that and take up a job its good to know a tool"(Academic #2)*

It was found that the software industry is flexible in using a notation. This is because they prefer to use one that stakeholders can easily interpret, understand and comprehend.

Table 4.11 shows two popular notations used in the industry (UML and BPMN).

Table 4.11: Modelling Notation Used in Industry

Name of Notation	Number of Participants used Notation
UML	4
BPMN	6

> *"BPMN is a standard and usually no one sticks to a standard. We do not have any standard for process flow diagram. Everyone knows BPMN. I think business does not care what you use as long as they understand. Some standards, I use BPMN I have some training as well in XXX country, some of the standards like symbols can be very tricky like business would never know what is it. They need to know what has to happen in a very simple way. So may be some companies internally understand one standard they understand religiously they use these things and they are very strict to it but with the clients whatever works that is the way to go" (BA#5).*

This section illustrated the findings of SA&D practices (methodologies, techniques, tools, notations and skills) in both academia and industry. Table 4.12 will summarise the findings from the Literature Review and the present study.

Table 4.12: Summary of Academics Findings from Literature Review and Present Study

Topic.	**LR**	**Current Research**
Methodology	Overseas studies show that academia teaches Structured & Object oriented methodologies	In addition to structured and OO some also teach or give an overview of Agile
Requirement Elicitation	Academia teach this topic and is very popular	NZ universities also cover this topic in a typical SA&D course
Techniques	Teaching the techniques depend on the methodology that is being taught. ERD, DFD were the most taught topic followed by UML & activity diagrams.Sequence & State diagrams were seldom taught	Every university in NZ teach this topic. Since in NZ this course it is taught in business school, therefore these differ depending on curriculum. However diagrams e.g.,DFD, sequence, & class diagrams were not popular
Tools	Tools are introduced. However no specific tools are taught	In NZ universities tools are introduced. However, no single tools is preferred
Notation	Not much information could be found	As such universities do not teach this as a separate topic, however, students are introduced to notation during delivery of the course
Skills	Technical Skills are taught such as requirement gathering techniques, data & process modelling Non-technical skills are supported in teaching course such as time management	In NZ universities also the course is designed so that students get to learn some technical skills. Emphasis is also given on non-technical skills by designing assessments that will help students to attain these skills

In the following sections, these findings with reference to the research questions from Section 1.4 will be analysed.

4.3 Analysis and Discussion

This study was conducted with the intention of exploring SA&D in both academia and the software industry. The main goal was to investigate the typical SA&D course and

map its relevance to the existing practices in the New Zealand software industry to highlight any gaps if they exist. As discussed in Section 3.7, Direct Content Analysis was used for data analysis.

Before answering RQ#8 stated in Section 1.4, it is important to note the duration of the typical SA&D course in New Zealand university. As mentioned in Section 2.2.3, the duration of an SA&D course in New Zealand is usually one semester[14]; in fact nearly 75% universities teach this course over one semester. This is similar to the findings conducted overseas (Holmes et al. (2002) and Burns (2011)) where, an SA&D course does not extend beyond one semester

As per the present study's findings, most universities in New Zealand, teach content from SMA and RAS under one semester[15] and focus on the analysis part of an SA&D course.

Participant #2 mentions that they ask students to show a non-working prototype at the end of the analysis phase. However, Academic#4 sometimes asks students to create a working prototype. This may imply that Academic#4 may be teaching all aspects of SA&D. Thus depending on what universities are aiming to teach, and the focus of the course, they may choose to teach selective content [16].

A typical SA&D course contributes to the 15 points, which means a total 150 hours workload is required. [17]

Depending on schedules, academia divides the teaching hours as follows:

Contact hours: The course content is delivered through lectures. Tutorials/labs are conducted for practical sessions.

Student Directed Learning: Students work on their course related assignments and

[14]Table 4.3 shows the course duration and programme/major/discipline under which SA&D is taught in New Zealand

[15]as discussed in Section 2.2.1.1 SA&D is grouped into SMA, RAS and SD

[16]This depends upon the objective of the course they decide to teach, e.g., academics who teach SA&D in business faculty, will focus on business side and will have less technical contents as compared to ones who teach it under Computer Science or IT

[17]The details about the contact hours/point system can be found in Table F.1

group work and prepare for tests and examinations in their own time

In this short period academics try their best to teach methodologies, requirements elicitation, modelling using tools and techniques, and notations. All topics are taught so that graduates gain domain knowledge, which they can apply when they join the software industry.

My findings show that across the universities, most academics teach similar topics'. However, the content are not covered equally, e.g., different academics teach different methodologies, requirements elicitation and modelling techniques and not all of the behaviour diagrams are included. This supports the statement made by Harris et al (2006) mentioned,

> *"If you were to assemble 50 IS professors into one room, you would likely end up with 50 different opinions about how best to teach SA&D. The one thing that they would probably all agree on is that it is a very challenging course to teach".*

This implies that they may have investigated similar situations where the content of a topic were taught differently, e.g., methodologies may have been covered differently or under a different course than an SA&D course, requirements elicitation techniques may or may not have been taught in detail, the modelling techniques may have been different and/or designing may or may not have been a part of the course.

The following section will analyse the typical SA&D course to answer research question #8.

- **What are the main differences (i.e., gaps) between (RQ2, RQ3, RQ4) and (RQ5, RQ6, RQ7)?**

4.3.1 Methodology

It was discovered that most academics teach this topic. In addition to the structured, object-oriented and Agile methodologies discussed in Section 2.1.1[18], they also give an overview of other methodologies such as spiral.

My findings from industry show that there is not a single methodology that can be applied to every project, and the choice of methodology depends on a number of factors and is usually project driven.[19] The methodologies used in the New Zealand software industry are structured,(Waterfall), Agile (Kanban, SCRUM).

As seen in Chapter 2, Powell and Yager (2013) conducted a survey with 100 industry participants to find the methodologies employers considered most important. They found that about one third of them used structured/traditional, object-oriented and Agile. Another survey with 214 academics conducted by Guidry et al (2011) found that academics had their own perceptions of importance of content, those who taught structured methodology, considered important to teach all techniques found under structured methodologies, such as ERD, DFD and context diagrams and those who taught object-oriented methodology considered it to be the most important. The present study found that, though, in New Zealand 50% of academia either teach or have introduced Agile, yet they still teach structured and object-oriented methodologies. It was also found that whatever methodology they teach, nearly all of them give an overview of all methodologies.

This may mean that students who are enrolled in courses where object-oriented methodology is taught will have the concept of an iterative and incremental development. Since Agile is also based on an iterative and incremental approach, graduates will be able to apply the understanding of this approach in learning Agile if they follow it at their future workplace.

[18] Structured, Object-oriented and Agile

[19] Please refer to section 4.2.2.5

However, as the course duration is only one semester, and the focus is to cover the content discussed in Section 2.2.1, it is given that universities cannot cover all the three methodologies in detail. Therefore, the academics should decide what methodology and how much detail they want to teach in the course.

4.3.2 requirements elicitation

The present study found that the New Zealand software industry considers requirements elicitation to be the backbone of a system and a critical component of SA&D. This is because development of a system depends on the requirements of the stakeholders. If these are not met and if end users are not satisfied with the system delivered, then both the effort and the time invested in gathering requirements are wasted.

My investigations show that industry mostly takes the ownership in requirements elicitation and either involve themselves or employ external agencies to help them to find out the requirements of the stakeholders, and if they would add value to the stakeholders in solving business problems, eliciting, analysing, and signing off the requirements.

RE is one of the many topics that is taught in a typical SA&D course at NZ universities over the duration of one semester. Also, industry involves graduates in the process from gathering to documenting and elaborating the initial requirements.

This is one of the many topics that academics teach in one semester. Time does not seem sufficient to teach all topics that help students with gathering techniques, analysing, writing the functional and the non-functional requirements, and working with a template or semi-formal grammar to document these. Also, academics teach the SA&D course through case studies and scenarios (no real client exists) so that students develop some skills (such as brainstorming and communication) to get an initial set of requirements. Analysis and prioritisation of the requirements is a challenge in itself. Therefore, it

is important that time and teaching materials should be thoughtfully prepared to give students an understanding of analysing the problem domain. The real-world examples should be introduced to reflect the contemporary needs of the industry. This will help students gain competency and proficiency.

Similar findings to the present study were found in overseas research (i.e., (Powell & Yager, 2013)) that investigated the importance of similar content (such as SA&D skills, tools, practices, and methodologies) by employers when hiring the new staff. It was found that 93 out of 100 participants recognised the importance of requirements elicitation techniques.

4.3.3 Techniques

As discussed in Chapter 2, industry and academia both consider the use of techniques equally important because they help to build a prototype for a system (Lending & May, 2013). This enables stakeholders to understand how the new or amended system will behave.

Powell and Yager (2013) found that employers considered that techniques such as user technical documentation, project scope statement, use case diagrams, DFD, process models, and ERD were important for new hires to know.

The present study is in conjunction with overseas studies conducted by Khoo (2011). Students often find it difficult to understand the object-oriented concepts and it becomes more challenging for them to apply to the course work examples and assignments. When working on a case study, it becomes even more difficult for them to identify the requirements clearly because there is no communication between a real client and students. To solve this problem, academics New Zealand universities teach techniques such as use cases and use case diagrams, and process diagrams. Students are required to complete their written assignments using these techniques. Academics put in a lot

of effort into choosing case studies and scenarios so that students are able to grasp the knowledge needed, apply it to their coursework and then take this technical skill with them when they graduate. However, there is disagreement between academia and industry in terms of the importance of the techniques. Techniques such as sequence diagrams and state machine diagrams are not popular within the New Zealand software industry. These results are similar to the ones found in overseas studies (e.g., (Guidry et al., 2011)). Thus, the effort of teaching such techniques is wasted.

> *"Yeah there is always a gap. Even personally myself, 95% of what I learnt, I never used that. It's not that university is out of touch. Challenge is what you learnt in university, is typically at crude level someone senior will be doing."BA#6*

This indicates a huge gap between academia and industry. This is because the focus is on topics that take a lot of time and effort to teach which may not be useful for graduates when they join the software industry.

There is no doubt that techniques train students to produce a proficient documentation. To do so, generally one has to make use of some tool. The following section will investigate the relevance of tools in developing a useful documentation.

4.3.4 Tools

The present study and overseas research (Powell and Yager (2013)) agree that specific tools are not important in SA&D practices open-source tools are as good as 'fancy' tools.

MS-Visio and Lucid-chart are gaining popularity in the software industry. This may be because MS-Visio is a Microsoft tool and is easily available, whereas Lucid Chart is an open-source UML tool. These tools are preferred because they are less expensive and bypass licensing limitations.

> *"We use Enterprise Architect which is very expensive and we have limited licences but young BAs adapt Visio especially 2013. But now as a company we are switching to Lucid Charts. I recently switched from Visio to Lucid Chart and I think I will be using that."(BA#5)*

Both industry and academia use tools and Tables 4.6, 4.10, and Sections 4.2.2.4 and 4.2.2.7 show the tools industry and academia prefer to use. However, none of them considers tools as the key to SA&D. Though industry highlights the need for knowing, the employability of graduates does not depend on knowing or not knowing a tool. This is in agreement with the study conducted in New Zealand by Kirk and Tempero (2012), who had also found that open-source tools are recommended because tools are required for documentation only.

4.3.5 Notation

As seen in Section 2.2.7, the New Zealand software industry uses diagrammatic techniques to model the data and processes of a system. UML and BPMN are the two most common notations the industry uses. BPMN is becoming increasingly popular and both academia and industry seem to be moving towards adopting it. However, this can only be used for developing processes. UML is still needed for other diagrams until some other notation replaces it. However, to the best of my knowledge UML has not been able to get replaced yet.

In Chapter 2, the studies that have investigated SA&D content (e'g', (Powell & Yager, 2013; Bataveljic et al., 2006; Guidry et al., 2011; Tastle & Russell, 2003)) did not investigate the use of notation in SA&D practices.

However, in New Zealand, academics spend a lot of time teaching UML notations. This is not taught as a separate topic, but while teaching UML modelling techniques it is

explained in great detail.[20].

4.3.6 Skills

It was found that both industry and academia find some skills vital of SA&D courses. The learning outcomes of SA&D courses can help students gain technical skills[21]. However, when students graduate, they are expected to have non-technical or soft skills. Table 4.13 lists some of the soft skills that both industry and academia think are important for every one in the SA&D field.

Table 4.13: Soft Skills Expected by Industry for an SA&D Role

S.No.	Soft-Skills
1.	Team Building
2.	Communication
3.	Problem Solving
4.	Critical Thinking
5.	Cultural Fit
6.	Time Management Skill
7.	Stakeholder Management Skill

In general these days a number of stakeholders are involved in a business. It is a challenging task to get the right information about their business needs so that their requirements can be understood. Not all stakeholders are technical, therefore it is very important to interpret the information, understand their requirements, analyse and document it. This requires many skills.

The present study is in agreement with a study conducted by Powell and Yager (2013) that team work, communication, and stakeholder management skills are all needed and are very important. Also, when compared with other studies such as (Cappel, 2002; Khoo, 2011; Powell & Yager, 2013; Karanja et al., 2016)[22], the present study agrees that in absence of SA&D experience, it is vital for graduates to acquire a variety of

[20]please see Section 4.2.2.4

[21]Learning outcomes can be found in Table F.1

[22]These studies were investigated during literature review

skills in areas such as communication, critical thinking, problem solving, and evaluation plus they need persuasion skills, decision-making skills, and time management skills. This is because with these skills and the academic knowledge graduates bring to the industry, they would be able to apply these to the projects they would work on.

The goal of this research is not to compare New Zealand universities academia with those overseas, but to interpret and discuss the findings with previous research so that new insights can be generated to create a body of knowledge.

Powell and Yager (2013), conducted their study with their graduate students to investigate which topics should be included in an SA&D course and what are their perceptions about how much time should be allocated to these topics. This is an on-going and therefore conclusions cannot be derived from the results of their study. However, the present study shows that in New Zealand, use of CASE tools is neither emphasised by academia nor is a must in industry. It is recommended that graduates have knowledge of a tool, and soft skills, like communication and working in teams.

Guidry et al (2011), who assessed the importance of SA&D course topics overseas, found that the least amount of time should be spent on methodologies and state machine diagrams. Comparing this research with New Zealand, based on the investigations of this present study, it is difficult to comment on methodologies because this topic is spread around various courses. Also from the interviews and the on-line descriptors, it is difficult to comment which course, teaches methodologies in detail.

Table 4.9 indicates that class diagrams, DFD, sequence diagrams and state diagrams are not popular with New Zealand software industry. Also industry findings of the present study are somewhat similar to another overseas study that indicates that graduates should bring knowledge of the concepts they have learnt at university. (Guidry & Stevens, 2014).

In addition to the above mentioned studies, Radermacher et al (2014), investigated the skills that industry expected from graduates. One deficiency they found was lack of

experience. However, it is recommended that graduates have knowledge of the basic concepts of SA&D and soft skills such as good oral and written communication, and be able to apply that knowledge when they join the industry.

4.3.7 Gap Analysis

The main objective of this research was to answer RQ#8 raised in Section 1.4 :

> **Does the course content of an undergraduate SA&D course taught in Computing curriculum** (e.g. Information Systems, Information Technology, and Computer Science) **in NZ universities meet the expectations of the New Zealand software industry?**

This question will be answered based on the analysis from Sections from 4.3.1 to 4.3.6. My investigations show that the content covered in an undergraduate degree SA&D course are similar to the needs of the software industry but not all the topics were consistently covered. This indicates that there is sufficient evidence of a gap between academia and industry.

To answer the research question #8, and to highlight the gaps, research sub-questions from Section 1.4, relating to the similar topics and practices, will be answered jointly so that a comparison can be made between the two participating fields.

1. **RQ #2 and RQ #5- Gap in Methodology:**[23]

   Academics are teaching structured, object-oriented, and Agile methodologies. Industry participants are using structured methodology but some are moving to Agile. Depending on the projects, they also use a hybrid approach i.e., a combination of Structured and Agile methodologies. This is because the choice of methodology depends on many factors and project domination is one of them.

[23] **1. RQ #2**: What methodologies are covered in a typical SA&D course in New Zealand?
2. RQ #5: What methodologies are used in the software industry in New Zealand?

Industry participants seldom mentioned the use of object-oriented methodology. Only two out of ten participants said that they used it. This indicates a gap because universities are using different methodologies in teaching this topic and methodology used in the industry is mainly project driven, i.e. driven by organisational context, business and project requirements.

2. **RQ#3 and RQ#6- Gaps in skills and techniques**[24]

   The typical SA&D course teaches both, the requirements elicitation and the modelling techniques. Although academics aim to provide awareness of most requirement gathering techniques, that only gives a theoretical knowledge to students. The gap lies in that they cannot get the practical experience using these techniques. Therefore, they will lack the skills to conduct interviews, workshops, focus groups and techniques. Furthermore, they would not be able to use their latent thinking to dig deep to get the requirements from different stakeholders when they join the software industry.

   Academics teach various techniques such as ERD, class diagram, context diagrams, process diagrams and UML diagrams. It was found that depending on the methodologies they are teaching, academics may not be teaching all techniques. Therefore the graduates who did not learn these techniques or concepts would not be skilled to use them in the software industry. Another issue is that academia teaches sequence diagrams and state machine diagrams. However, participants from industry state that they seldom use these techniques. Does this mean that academia lack the understanding of the needs of the industry?[25].

3. **RQ#4 and RQ#7- Gaps in tools and notation** [26]

[24] **1. RQ3:** What skills and techniques are covered in an SA&D course in New Zealand?
2. RQ6: What skills and techniques are needed in the software industry in New Zealand?

[25] Since number of participant is not huge it can not be concluded that the software industry does not use these techniques at all in New Zealand

[26] **1. RQ4:** What tools and notations are covered in a typical SA&D course in New Zealand?

The software ndustry does not use any specific tools for SA&D practices. However they do recommend that graduates should know how to use a tool. Students at universities are required to use a tool for their learning and are expected to use it for documenting their work. This shows that there is not much difference in the perception of both the groups of the participants in the present study.

Yet there seems to be a gap in terms of notation. The software industry is using BPMN for processes but it has been found that academia has not fully integrated this notation in their curriculum. It takes time to prepare the course content as well as delivering it. Since the semester is only twelve weeks long, spending it on the topics and practices the software industry is not going to use is wasteful.

In addition to the above, gaps are also evident in terms of lack of awareness in terms of the content taught at universities as well as a lack of experience or the perception of a lack of experience among the recent graduates

This chapter has analysed the typical SA&D course and the SA&D practices in the New Zealand software industry. The investigations have helped to evaluate the typical course taught at the New Zealand universities. This evaluation assisted in exploring the SA&D practices in the software industry. The information obtained, helped to assess the gaps between academia and the industry.

The next chapter will summarise and conclude the research with lessons learnt, limitations and recommendations.

RQ7: What tools and notations are practised in the software industry in New Zealand? 1.4

Chapter 5

Conclusions and Further Research

5.1 Conclusions

The present study investigated the alignment of System Analysis and Design (SA&D) education with industry expectations in New Zealand.

The aim of this research was to explore the content skills, methodologies, tools, and techniques offered in one-semester undergraduate SA&D courses, and compare them with the skills, methodologies, tools, and techniques expected of entry-level computing graduates in New Zealand's software industry.

A literature review was conducted to gain an in-depth understanding of SA&D courses and related studies in the extant literature.

To accomplish this, academics from five universities involved in teaching the course, (my university teaching team was not interviewed), and ten participants from industry- business analysts and project managers (who were working closely with business analysts)- participated in this research. Semi-structured interviews were carried out for this research, and to protect the information of the participants, ethical approval was obtained. This research used content analysis for analysing the interview data.

To answer the main research question, research sub-questions were formed to explore

the SA&D content such as methodologies used or taught, tools and techniques for requirements elicitation and modelling, popular SA&D notations and the importance of technical and non-technical skills.

5.1.1 Findings Gaps

5.1.1.1 Methodology

As discussed in Section 4.3.7 a gap between academia and industry was observed. Academics teach structured, object-oriented methodologies, and four universities out of six teach Agile. Some of them give an overview of methodologies because other courses in the undergraduate degree programme covers methodologies in detail. Since the focus of this study was on an SA&D course, the other courses were not explored. Hence I was not able to investigate if students who take an SA&D course also take the other courses where methodologies were taught in detail. Though most participants from industry used structured, Agile or hybrid methodology, they claim that the use of methodology depends onthe projects undertaken and an understanding of the client organisation.

Academics usually start with structured methodology, and teaching the content gives students an awareness of all methodologies under structured methodology or an approach such as Waterfall. To shrink the gap between the SA&D courses and the software industry, it is recommended that instead of teaching object-oriented or Agile methodologies, academics should teach an iterative and incremental approach. This practice will teach students the concept of SA&D methodologies and help them to apply their knowledge to any other methodology they will encounter at their workplace.

5.1.1.2 Techniques

The software industry uses different types of modelling techniques for data, processes and behaviour models. [1] This study found that students from an IS major find it difficult to understand the object-oriented concepts. Therefore, it is recommended that a pre-requisite course should have some SA&D concepts so that students from all majors have an understanding of the object-oriented concepts such as UML techniques.

Not all modelling techniques taught by participant academics are used in the industry.[2] Similarly, ERD and context diagrams are not taught by all academics in a. SA&D course. University#1 teaches these in a different course; this may be the case with other universities. Therefore it is recommended that academics from New Zealand uniformly develop the course work, if possible by seeking input from the software industry.

This current study found academia spends time on teaching topics that are not popular with the industry professionals.[3] As a university semester is only twelve weeks long, it is recommended that the time spent on these unpopular topics should be reduced, and more focus placed on important concepts such as use cases, which are used extensively by the software industry of New Zealand.

5.1.1.3 Tools

As discussed in Section 4.3.7, there was no noticeable gap in using tools. Both academics and participants from the industry use at least one tool in their respective practices. However, findings indicate that although academics encourage students to use tools in producing course documentation, not all students use CASE tools. At university #1, it was observed that some students physically draw data and process models. These students will be at a disadvantage in the workplace. This is because as evident in this

[1] Refer to Table 4.9

[2] Sequence and state machine diagrams are seldom used in the software industry

[3] Investigations show that sequence and state diagrams are not popular in the software industry

present study's findings the industry expects graduates to be familiar with at least one tool. As mentioned by academic participant #1, that if somebody mentions 'rose' at workplace, graduates should know that it is the Rational Rose they are referring to not the flower. Consequently, it is important that academia should encourage students to use appropriate tools. This will help students to learn a technical skill for producing technical documentation. This will give them confidence in the workplace.

5.1.1.4 Skills

Academics invest a lot of hard work and time into designing the SA&D course work to teach students both the technical skills as well as non-technical (soft) skills. However, not all skills, such as people skills and oral communications skills, can be developed with the coursework and written assessment material. Therefore it is important that some coursework should focus on presentation as well.[4] This is because presentations will help students to improve their oral communication , critical thinking (by answering questions during Questions/Answers sessions), team building and team management skills

As discussed in Section 4.2.2.6 in addition to these skills, academics should also incorporate suggestions from industry such as having students take a course in psychology, which is particular important when the graduates would be dealing with different stakeholders, and some entrepreneurial training to help the newly-hired graduates understand the culture of the organisation they find themselves in.

5.2 Limitations of the study

It is rightly said that many researchers complete their data collection before they have given sufficient thought to their data analysis (Burgess, n.d.).This present study also

[4]Table F.1 shows that University #2 allocates marks for presentation

faced similar issues.

In terms of "course", it would have been beneficial if course coordinators were contacted to obtain the recent SA&D course descriptors to get an insight into the course content. This is because on-line versions were either brief or were not updated. If the course descriptors had updated content it would have given a better insight into the content taught in different universities.

Other limitations that affected the findings were staff reshuffling and course re-structuring because, I could not find out if there was any change in the course content or assignments.

The sample size of ten participants from industry was very small. Most participants were either system analysts or business analysts and therefore, this present study was restricted to their perceptions. It would have been useful if there was a wider range of participants who could have shared their viewpoints and experiences in SA&D.

Also, it would have been interesting to include graduates who had co-operative training in software development in the final year of their studies. That would have helped academia to understand the struggle some graduates face when they join the software industry, as well as reveal the strength of their university education. It could also highlight the expectations the software industry has of new graduates.

5.3 Implications

5.3.1 Implications for Academics

The SA&D course content is vast, but the duration of the course is very short.[5] Academics struggle to cover all topics because they cannot decide which topic is the most important from the software industry's perspective. At present the typical SA&D course

[5] 1 semester=12 weeks

taught is introductory meaning it briefly covers the topics discussed in Sections 4.2 and 4.3. Be that as it may,in terms of the work load it is enormous and must be a challenge for academics to teach and students to absorb the content.

In the present study I observed that methodologies needed for the software industry are in fact taught in another course, likewise, there might be a possibility that content such as requirements elicitation might also be taught in other courses. However, it is not known if these courses are core courses or a pre-requisite for the SA&D course that students from all majors (IS/IT) must take. Consequently, for academia it is suggested that:

1. An SA&D course should be spread over more than one semester.
2. While deciding on the content, consult with other (IT/IS) course coordinators to avoid overlapping of the content.
3. Pre-requisite courses should introduce SA&D concepts.[6]
4. In university #1 students in the IT discipline are required to do some courses that teach them about business and project management. Other universities should look at designing the curriculum so that these courses are taught prior to students taking an SA&D course.
5. Rapidly changing technology is dominating the business sector in general, therefore academia should revise the SA&D courses a minimum of once every five (preferably three) years to minimise the gap between the SA&D courses and software industry expectations.

5.3.2 Implications for Industry

Though the topics taught at university SA&D courses were very similar to SA&D practices, there was a considerable gap between academia and industry.

1. Findings show that the New Zealand software industry produces tech-reports such as KPMG report(Brame & Barlow, 2010)). These reports should be summarised

[6] As seen in this presnt study's findings that students from IS find UML concepts difficult to understand

and sent along with a summary of accepted SA&D practices to academia to help them reflect and re-visit their course once in a three to five year period.

i. Similar to these reports, public sector organisations should also share the summaries of their SA&D practices to keep academia up-to-date.

2. Industry should collaborate with academia in designing SA&D courses to ensure the course meets their expectations.

3. Alumni should form a social group such as a meet-up group where they can have an informal talk/chat about SA&D practices. This may help academia to design their coursework or assignments.

4. It is recommended that from their projects the software industry creates scenarios/case studies and share them with academia for them to use in their coursework and to design effective assignments.

5. In New Zealand, the software industry and academia could partner together for events/conferences every year, and also jointly come up with some certifications for university students/graduates.

Industry can play a major role in minimising the gap by working with academics. As BA #4 suggested, pair a student from Computer Science with a student from Business to let them work together. This will help them to learn and benefit from each other.

5.4 Future Study

This current study is the first known step in investigating the typical SA&D course in New Zealand and matching it with the expectations of the country's software industry. It is a contribution to the body of knowledge in education research, but it is not a conclusive study. Future studies are recommended to address the following:

- Most participants from the software industry were either system analysts or business analysts. As the trend of SA&D methodology is shifting to Agile in practice, and the role of IS is increasing, it is therefore mandatory to have the perspectives of other roles such as developers, SCRUM masters, product owners, and domain experts.

- The sample number in this present study was not sufficient to fully reveal the industry's expectations of new graduates. More participants from a diverse range of roles should be invited to take part in future research.
- To quantify the SA&D practices in industry, a nation-wide survey is recommended and, based on the survey results, a further round of interviews is suggested.
- A focus group is recommended that has participants from varied roles in SA&D, such as business analysts and system analysts, developers, testers, and Agile practitioners. This will familiarise researchers with SA&D practices in New Zealand.
- A focus group of both industry and academia will help both groups of participants because academia will be more aware of the needs of the software industry and as industry will be aware of the strengths and weaknesses of graduates, they will accordingly be able to assign appropriate roles to graduates when they join the software industry.
- In New Zealand, many technical institutes and private training institutes teach an SA&D course. A survey including all of these institutes will give insight into the course content on a truly national scale.

It is hoped that the outcome of this thesis will be that it is a building block for further investigations into SA&D practices, both in academia as well as industry.

References

AUT. (2014). *Informed and voluntary consent (2.1).* Retrieved 2016-04-05, from https://www.aut.ac.nz/research/guidelines-and-procedures/informed-and-voluntary-consent-2.1

Bachman, R. D. & Schutt, R. K. (2016). Survey questions. In *The practice of research in criminology and criminal justice.* SAGE Publications.

Bajaj, A., Batra, D., Parsons, J. & Siau, K. (2005). Information Technology and systems-I systems analysis and design: Should we be researching what we teach? *Cais*, *15*(1), 27.

Balaji, S. & Murugaiyan, M. S. (2012). Waterfall vs. v-model vs. agile: A comparative study on sdlc. *International Journal of Information Technology and Business Management*, *2*(1), 26–30.

Banerjee, S. & Lin, W. (2006). Essential entry-level skills for systems analysts. *Journal of Education for Business*, *81*(5), 282–286.

Baram, G. & Steinberg, G. (1989). Selection criteria for analysis and design CASE tools. *ACM SIGSOFT Software Engineering Notes*, *14*(6), 73–80.

Bataveljic, P., Eastwood, M. & Seerfried, H. (2006). An approach to teaching object-oriented analysis and design. *Journal of Information System Education*, *17*(3), 267–272.

Batra, D. & Satzinger, J. W. (2006). Contemporary approaches and techniques for the systems analyst. *Journal of Information Systems Education*, *17*(3), 257.

Begel, A. & Simon, B. (2008). Novice software developers, all over again. In *Proceedings of the fourth international workshop on computing education research* (pp. 3–14). ACM.

Brame, A. & Barlow, G. (2010). *KPMG New Zealand Project Management Survey 2010* (Tech. Rep.). Retrieved from http://www.beconfident.co.nz/files/events/Project-Management-Survey-report%20copy.pdf

Burgess, T. F. (n.d.). A general introduction to the design of questionnaires for survey research. In *Guide to the design of questionnaires* (pp. 1–27).

Burns, T. J. (2011). Defining the contents of the undergraduate System Analysis and Design Course as measured by a survey of instructors. *Information Systems Education Journal*, *9*(5), 1–11.

Burns, T. J. & Klashner, R. (2005). A cross-collegiate analysis of software development course content. In *Proceedings of the 6th conference on information technology*

education (pp. 333–337). ACM.

Cappel, J. J. (2002). A system analysis and design case: Abc church. *Journal of Information Systems Education*, *12*(4), 233.

Chinosi, M. & Trombetta, A. (2012). Bpmn: An introduction to the standard. *Computer Standards & Interfaces*, *34*(1), 124–134.

Dasgupta, S., Haddad, M., Weiss, P. & Bermudez, E. (2007). User acceptance of case tools in system analysis and design: an empirical study. *Journal of Informatics Education Research*, *9*(1), 51–78.

Dennis, A. & Wixom, B. (2002). *System Analysis and Design An Applied Approach*. John Wiley & Sons Inc.

Dennis, A., Wixom, B. & Tegarden, D. (2009). *System Analysis Design UML Version 2.O* (Fourth ed.). John Wiley&Sons,Inc.

Dennis, A., Wixom, B. & Tegarden, D. (2015). *System Analysis Design: An Object Oriented Approach with UML* (Fourth ed.). John Wiley&Sons,Inc.

Erturk, E. (2014). A critical Inquiry:Teaching System Analysis and Design Beyond 2015. In *Computing and information technology research and education new zealand.* Retrieved from `http://www.citrenz.ac.nz/conferences/2014/pdf/2014ITx{_}21{_}CITRENZ{_}Erturk{_}30.pdf`

Evans, A., France, R., Lano, K. & Rumpe, B. (2014). Developing the uml as a formal modelling notation. In *Uml'98, lncs (vol. 1618).* Retrieved from `https://wwwbroy.in.tum.de/publ/papers/EFLR98b.pdf`

Forman, J. & Damschroder, L. (2008). Qualitative content analysis. *Empirical Research for Bioethics: A Primer. Oxford, UK: Elsevier Publishing*, 39–62.

Gerbic, P. & Stacey, E. (2005). A purposive approach to content analysis: Designing analytical frameworks. *The Internet and Higher Education*, *8*(1), 45–59.

Gill, P., Stewart, K., Treasure, E. & Chadwick, B. (2008). Methods of data collection in qualitative research: interviews and focus groups. *British dental journal*, *204*, 291–295.

Grant, B. M. & Giddings, L. S. (2002). Making sense of methodologies: A paradigm framework for the novice researcher. *Contemporary nurse*, *13*(1), 10–28.

Grover, R. & Vriens, M. (2006). *The Handbook of Marketing Research: Uses, Misuses, and Future Advances*. SAGE Publications.

Groves, L., Nickson, R., Reeve, G., Reeves, S. & Utting, M. (2000). A survey of software development practices in the new zealand software industry. In *Software engineering conference, 2000. proceedings. 2000 australian* (pp. 189–201). IEEE.

Guidry, B. N. & Stevens, D. P. (2014). Comparing perceptions of the systems analysis and design course. *Journal of Computer Information Systems*, *55*(1), 40–47.

Guidry, B. N., Stevens, D. P. & Totaro, M. W. (2011). The systems analysis and design course: An educators' assessment of the importance and coverage of topics. *Journal of Information Systems Education*, *22*(4), 331.

Harrell, M. C. & Bradley, M. A. (2009). *Data Collection Methods Semi-Structured Interviews and Focus Groups* (Tech. Rep.). Retrieved from `www.rand.org/`

content/dam/rand/pubs/technical{_}reports/2009/RAND{_}TR718.pdf doi: 978-0-8330-4889-9

Harris, A. L., Lang, M., Oates, B. & Siau, K. (2006). Systems analysis & design: An essential part of is education. *Journal of Information Systems Education*, *17*(3), 241.

Hart, C. (2005). *Doing a Literature Search: A Comprehensive Guide for the Social Sciences*. London: SAGE Publication Ltd.

Hellerstein, J. M. (2008). Quantitative Data Cleaning for Large Databases. *United Nations Economic Commission for Europe*, 42.

Holmes, M. C., Kemm, E. & Yager, S. (2002). Learning outcomes in systems analysis and design courses. *Proceedings of IACIS*.

Hsieh, H. F. & Shannon, S. E. (2005). Three Approaches to Qualitative Content Analysis. *Qualitative Health Research*, *15*(9), 1277–1288.

Hunt, J. (2003). An introduction to the uml and the unified process. *Guide to the Unified Process featuring UML, Java and Design Patterns*, 21–37.

Hyde, K. F. (2000). Recognising deductive processes in qualitative research. *Qualitative market research: An international journal*, *3*(2), 82–90.

IEEE. (2013). *Draft For Public Review: Software Engineering 2013* (Tech. Rep.). Retrieved from http://computingportal.org/sites/default/files/SE2013Draft.pdf

Isaias, P. & Issa, T. (2015). *High level models and methodologies for information systems*. Springer.

Kanter, S. L. (2009). Ethical Approval for Studies Involving Human Participants. *Academic Medicine*, *84*(2), 149–150.

Kaplan, B. & Maxwell, J. A. (2005). Qualitative research methods for evaluating computer information systems. In *Evaluating the organizational impact of healthcare information systems* (pp. 30–55). Springer.

Karanja, E., Grant, D. M., Freemen, S. & Anyiwo, D. (2016). Entry Level System Analysts:What Does the Industry Want? *Informing Science: the International Journal of an Emerging Transdiscipline*, *19*, 141–160.

Keith, E. R. (2003). *Agile Software Development Processes A Different Approach to Software Design.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.84.1697&rep=rep1&type=pdf

Kelley, K., Clark, B., Brown, V. & Sitzia, J. (2003). Good practice in the conduct and reporting of survey research. *International Journal for Quality in Health Care*, *15*(3), 261–266.

Kemp, E. A., Phillips, C., Alam, J. & North, P. (2003). Software engineering practices and tool support: An exploratory study in new zealand. *AJIS*, *11*(1).

Khoo, B. (2011). Evaluating the effectiveness of the constructionist approach to object-oriented systems analysis and design pedagogy. *Review of Business Information Systems (RBIS)*, *13*(4).

Kirk, D. & Tempero, E. (2012). Software development practices in new zealand. In *2012 19th asia-pacific software engineering conference* (Vol. 1, pp. 386–395).

IEEE.

Klopper, R., Gruner, S. & Kourie, D. G. (2007). Assessment of a framework to compare software development methodologies. In *Proceedings of the 2007 annual research conference of the south african institute of computer scientists and information technologists on it research in developing countries* (pp. 56–65). ACM.

LeBlanc, R., Sobel, A., Diaz-Herrera, J. L. & Hilburn, T. B. (2014). *Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering* (No. February).

Lending, D. & May, J. (2013). Teaching tip using activity diagrams to model systems analysis techniques: Teaching what we preach. *Journal of Information Systems Education*, *24*(2), 91.

Lichtman, M. (2012). *Qualitative research in education: A user's guide*. SAGE.

Lin, J.-C. (2011). *Various approaches of systems analysis and design.* Retrieved from `http://www.umsl.edu/{~}sauterv/analysis/termpapers/f11/jia.html`

McCracken, W. M. (1997). Se education: what academia can do. *IEEE Software*, *14*(6), 27–29.

Mishra, A. & Deepty, D. (2013). A comparative study of different software development life cycle models in different scenarios. *International Journal of Advance Research in Computer Science and Management Studies*, *1*, 64 – 69.

Moody, D. (2009). The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, *35*(6), 756–779.

Muijs, D. (2010). *Doing quantitative research in education with spss*. SAGE Publications.

Munn, P. & Drever, E. (1990). *Using questionnaires in small-scale research. a teachers' guide.* ERIC.

Myers, M. D. (1995). Dialectical hermeneutics: a theoretical framework for the implementation of information systems. *Information systems journal*, *5*(1), 51–70.

Myers, M. D. (2013). *Qualitative Research in Business & Management*. SAGE.

Myers, M. D. & Newman, M. (2007). The qualitative interview in is research: Examining the craft. *Information and organization*, *17*(1), 2–26.

Myers, M. D. et al. (1997). Qualitative research in information systems. *Management Information Systems Quarterly*, *21*(2), 241–242.

Nuseibeh, B. & Easterbrook, S. (2000). Requirements engineering: a roadmap. In *Proceedings of the conference on the future of software engineering* (pp. 35–46).

NZQA. (2014). *Degrees and related qualifications guidelines for programme approval and accreditation to provide programmes.* Retrieved from `http://www.nzqa.govt.nz/assets/Providers-and-partners/Registration-and-accreditation/guidelines-degree-approval-and-accreditation.pdf`

Oracle. (2009). JD Edwards World. *CASE Computer Aided Software Engineering Guide*, 74. Retrieved from `http://docs.oracle.com/cd/E24096{_}01/`

import{_}export{_}a921.pdf

Orlikowski, W. J. & Baroudi, J. J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research*, *2*(1), 1–29. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.107{&}rep=rep1{&}type=pdf

Pandey, D., Suman, U. & Ramani, A. (2010). An effective requirement engineering process model for software development and requirements management. In *Advances in recent technologies in communication and computing (artcom), 2010 international conference on* (pp. 287–291).

Phillips, C., Kemp, E. A. & Hedderley, D. (2005). SOFTWARE DEVELOPMENT METHODS AND TOOLS: A NEW ZEALAND STUDY. *Australasian Journal of Information Systems*, *12*(2), 21–49.

Planing, P. (2014). Quantitative Research Approach. In *Innovation acceptance the case of advanced driver-assistance system* (pp. 141–247). Springer Gabler. Retrieved from http://www.springer.com.ezproxy.aut.ac.nz/springer+gabler/management/book/978-3-658-05004-7

Powell, A. & Yager, S. (2013). Research in progress / teaching systems analysis and design: what do students really need to know. *ACM*, 99–104. doi: 10.1145/2487294.2487314

Radermacher, A., Walia, G. & Knudson, D. (2014). Investigating the skill gap between graduating students and industry expectations. In *Companion proceedings of the 36th international conference on software engineering* (pp. 291–300).

Ritchie, J. & Lewis, J. (2003). *Qualitative research practice: A guide for social science students and researchers*. SAGE Publications.

Rob, M. A. (2004). Issues of structured vs. object-oriented methodology of systems analysis and design. *Information Systems*, *V*(1), 275–280.

Robert, E. (2000). Computing Education and the Information Technology Workforce. *ACM SIGCSE Bulletin*, *32*(2), 83–90.

Rosenblatt, H. J. (2014). *System analysis and Design*. Boston: Cengage Learning.

Senapathi, M. (2005). A Framework for the Evaluation of CASE tool learnability in Educational Environments. *Journal of Information Technology Education*, *4*, 61–84.

Shahnazarian, D., Hagemann, J., Aburto, M. & Rose, S. (2013). *Informed consent in human subjects research.* Retrieved from http://oprs.usc.edu/files/2013/04/Informed-Consent-Booklet-4.4.13.pdf

Talbot, A. & Connor, A. (2011). Requirements engineering current practice and capability in small and medium software development enterprises in New Zealand. In *Proceedings - 2011 9th international conference on software engineering research, management and applications, sera 2011* (pp. 17–25). doi: 10.1109/SERA.2011.39

Tastle, W. J. & Russell, J. (2003). Analysis and design: Assessing actual and desired course content. *Journal of Information Systems Education*, *14*, 77–90.

Teijlingen, E. v. & Hundley, V. (2002). The importance of pilot studies. *Nursing Standard*, *16*(40), 33–36.

Topi, H., Valacich, J. S., Wright, R. T., Kaiser, K. M., Nunamaker Jr, J., Sipior, J. C. & De Vreede, G. (2010). Curriculum guidelines for undergraduate degree programs in information systems. *ACM/AIS task force*.

Waldo, J. (2006). On system design. In *Acm sigplan notices* (Vol. 41, pp. 467–480).

Weber, R. P. (1990). *Basic content analysis* (No. 49). Sage.

Wengraf, T. (2001). *Qualitative Research Interviewing: Biographic Narrative and Semi-Structured Methods*. SAGE Publications.

Williams, L. (2007). A Survey of Agile Development Methodologies. *Univercsty of the West England*, 209–227.

Wohlin, C. & Aurum, A. (2005). *Engineering and managing software requirements*. Springer Science & Business Media.

Appendix A

Invitation Cover Letter

To whom it may concern

My name is Vandana Taneja. I am an MPhil student of AUT University. I would like to invite you to participate in my research. The proposed research aim is to compare the Software Analysis and Design course with the requirements of software industry in NZ. This course is taught to undergraduate students in a semester in second year. This aim is to investigate and address any gaps if found during this research.

This research involves understanding current practices in software industry of NZ. You are invited to provide your expert contribution to the body of knowledge in this area.

Please note that your participation in this research is totally voluntary in nature, and you may decline or withdraw your participation at any time without any adverse consequences.

This research is being conducted under the supervision of Dr. Mali Senapathi, a senior lecturer at the School of Computer and Mathematical Sciences, who is a course co-ordinator for System Analysis and Design course at AUT.

Together with this invitation you will find a Participant's Information Sheet which provides some further details of the research, the process and answers to some common questions.

To confirm your willingness to contribute to this project, please mail or email a completed Participant Consent form (also included) to the address below. I will contact you in a few weeks regarding the next step in partnering with us on this research. Please feel free to email me if you require clarification or further information.

We trust you that this research will add value to both the industries involved (education and software) in this investigation. I will be looking forward to your response.

Yours sincerely,

Vandana Taneja

email: vtaneja@aut.ac.nz

Contact#: +64 9 9219999 ext 6376

PARTICIPANT INFORMATION SHEET

DATE INFORMATION SHEET PRODUCED:

dd mmm yyyy

PROJECT TITLE

Data and Process Modelling: Investigating the gap between education and industry expectations in New Zealand

AN INVITATION

Kia Ora!

My name is Vandana Taneja. I am a student of Masters of Philosophy (MPhil) . I write to you in my capacity as a researcher within the Software Engineering Research Laboratory at Auckland University of Technology (www.serl.aut.ac.nz)

The purpose of this sheet is to request you to participate in my research. I intend to research a course that is being taught at AUT. I am interested to investigate that the contents taught in System Analysis and Design (SA&D) course has relevance to the requirements of the industry. The aim is to understand the requirement of software industry of NZ so that course contents can be revisited.

I would like to invite you to participate in my research and take part in a survey related to this topic. Your agreement to participate will be highly appreciated. Your knowledge, experience and expertise will help us to evaluate and improve the course in SA&D. Your participation is completely voluntary and you may withdraw at any time without any adverse consequences. If you need further information, please feel free to contact me. My contact details can be found on page#3 of this document under 'Researcher Contact Details' at the end of this document.

WHAT IS THE PURPOSE OF THIS RESEARCH?

The purpose of this research is to investigate the Data and Process Modelling taught as SA&D course at AUT and its significance to the industry. The outcome will help course co-ordinator at AUT to re-examine the course contents according to the need of software industry in NZ. The investigations of this study will be a part of my MPhil research and may be used towards a PhD.

HOW WAS I IDENTIFIED AND WHY AM I BEING INVITED TO PARTICIPATE IN THIS RESEARCH?

This research aims to take an input from the people working in software industry who use tools for modelling.

WHAT WILL HAPPEN IN THIS RESEARCH?

A survey will have a set of questions based on the contents of SA&D course taught at AUT. The answers to survey questions will help course staff to analyse the need of industry according to the contents of the course.

WHAT ARE THE DISCOMFORTS AND RISKS?

There are no discomfort or risks involved in this research

WHAT ARE THE BENEFITS?

This research is going to benefit the course provider to look compare the course contents with the requirement of software industry to produce the graduates according to the need of s/w industry in NZ.

On the other hand if any participant would like to ask for the report of research, this will help him/her to get an insight of teaching practice.

Thus both the industries will be able to work together to meet the need of s/w industry in NZ.

HOW WILL MY PRIVACY BE PROTECTED?

Collected information will be kept confidential at all times during the research. It will be secured from any unauthorized access and shall solely be used for research purposes only. Confidential information gathered incidentally shall be respected. Data and signed consent forms shall be stored separately. Participants shall have a right to access information, they provided for this research at any time by contacting the supervisor, who will ask researcher to furnish that.

Prior consent will be taken to access and analyse sensitive information which will be treated as confidential material.

All data will be destroyed after six years of the completion and submission of this research.

WHAT ARE THE COSTS OF PARTICIPATING IN THIS RESEARCH?

This survey will not take more than 60 minutes of participation's time.

WHAT OPPORTUNITY DO I HAVE TO CONSIDER THIS INVITATION?

Two weeks is a reasonable time to think if participants want to make their valuable contribution to this research. However if you need more information please email me within a week.

HOW DO I AGREE TO PARTICIPATE IN THIS RESEARCH?

To show your interest in this research please complete the attached Consent form and email it to me as an attachment.

WILL I RECEIVE FEEDBACK ON THE RESULTS OF THIS RESEARCH?

This research aims to investigate the course contents of SA&D at AUT. If you are keen to be an observer, please email to my supervisor.

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, Dr.Mali Senapathi, email: msenapat@aut.ac.nz , Contact# 09 921 9999 extn: 5213

PARTICIPANT INFORMATION SHEET Concerns regarding the conduct of the research should be notified to the Executive Secretary of AUTEC, Kate O'Connor, ethics@aut.ac.nz , 921 9999 ext 6038.

WHOM DO I CONTACT FOR FURTHER INFORMATION ABOUT THIS RESEARCH?

RESEARCHER CONTACT DETAILS:

Vandana Taneja

Email: vtaneja@aut.ac.nz

Contact#: 09 921 9999 Extn: 6376

PROJECT SUPERVISOR CONTACT DETAILS:

Dr. Mali Senapathi,

Email: msenapat@aut.ac.nz

Contact# 09 921 9999 extn: 5213

Approved by the Auckland University of Technology Ethics Committee on type the date final ethics approval was granted, AUTEC Reference number type the reference number.

Appendix B

Consent Form

Project title: **Data and Process Modelling: Investigating the gap between education and industry expectations in New Zealand**

Project Supervisor: Dr. Mali Senapathi

Researcher: Vandana Taneja

- I have read and understood the information provided about this research project in the Information Sheet dated dd mmmm yyyy.
- 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 I may withdraw myself or any information that I have provided for this project at any time prior to completion of data collection, without being disadvantaged in any way.
- If I withdraw, I understand that all relevant information including tapes and transcripts, or parts thereof, will be destroyed.

- I agree to take part in this research.
- I wish to receive a copy of the report from the research (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 type the date on which the final approval was granted AUTEC Reference number type the AUTEC reference number Note: The Participant should retain a copy of this form.

Appendix C

Ethics Approval Letter

9 June 2015

Mali Senapathi

Faculty of Design and Creative Technologies

Dear Mali Re Ethics Application: 15/166 Data and process modelling: Investigating the gap between education and industry expectations in New Zealand. Thank you for providing evidence as requested, which satisfies the points raised by the Auckland University of Technology Ethics Committee (AUTEC). Your ethics application has been approved for three years until 9 June 2018. As part of the ethics approval process, you are required to submit the following to AUTEC: • A brief annual progress report using form EA2, which is available online through http://www.aut.ac.nz/researchethics. When necessary this form may also be used to request an extension of the approval at least one month prior to its expiry on 9 June 2018; • A brief report on the status of the project using form EA3, which is available online through http://www.aut.ac.nz/researchethics. This report is to be submitted either when the approval expires on 9 June 2018 or on completion of the project. It is a condition of approval that AUTEC is notified of any adverse events or if the research does not commence. AUTEC

approval needs to be sought for any alteration to the research, including any alteration of or addition to any documents that are provided to participants. You are responsible for ensuring that research undertaken under this approval occurs within the parameters outlined in the approved application. AUTEC grants ethical approval only. If you require management approval from an institution or organisation for your research, then you will need to obtain this. To enable us to provide you with efficient service, please use the application number and study title in all correspondence with us. If you have any enquiries about this application, or anything else, please do contact us at ethics@aut.ac.nz. All the very best with your research,

Kate O'Connor

Executive Secretary

Auckland University of Technology Ethics Committee

Cc: Vandana Taneja

Appendix D

Research Protocol

Stage set-up

1. Venue: Participants who agreed to be interviewed at researcher's university, Auckland University of Technology, researcher booked a room in advance and informed participants via email with a map/direction of the venue.
2. Consent form: Since AUTEC required researcher to get consent form from all the participants, therefore, researcher had keep two copies printed. Participants signed a copy and they were given an oppurtunity to take one with them.
3. Recording device: Researcher made sure that, device was in working condition and was fully charged. However a power source and extra batteries were available in case of emergency.
4. Writing material: Researcher carried a pen, paper and a folder
5. Script: Researcher carried a question guide

Interview Procedure

1. Opening Script: Introduction to interviewee

   Hello/Kia Ora (smile and shake hand). Thank you for coming/meeting with me. I am Vandana Taneja. I am Vandana Taneja. I am doing MPhil at School of Engineering, Computerand Mathematical Sciences at AUT. My research area is the System Analysis and Design, how it is taught and what are the practices in New Zealand.

2. Introducing Research

   i. Being a researcher I would like to know the gaps, if any, by exploring the contents of the course throughout New Zealand academia and map its relevance with

the need of the industry. To help me find this gap, I need your input. I assure you that your participation and anonymity will be kept confidential at all times. You have the opportunity to withdraw from this research at any time by emailing me or my supervisor without giving any reasons.

ii. Before we start, here are the two copies of consent form could you please read and sign one? You are welcome to keep a copy for our reference.

iii. I will be tape-recording this interview. I assure you that this conversation will remain private and confidential. The recording is to get all important information and use it for the investigation for this research.

Before I use this information, I will email you a transcribed copy so that contents can be validated and also to make you aware of the contents that will be reported/used for the current research

3. Closing Script: That was all I wanted to ask. Is there anything that you want to share?

   Thank you for your time, I will transcribe the interview and will email you in about a week. If you think, there could be your colleague who could highlight any related area, please parr on my details to him/her. (smile and shake hand)

Appendix E

Interview guide for Industry

1. What is your role in this organisation?
2. How long have you been in this role?
3. What is the size of IS in organisation?
4. What kind of projects do you take?
5. On an average how many people work on a project?
6. What methodologies have you been using for the projects?
7. How important is the requirement elicitation in your organisation?
8. How much time does it take you to gather their requirements from your stakeholders?
9. Do you use any software tools for requirement elicitation?
10. What techniques do you?
11. Do you have to createany documentations?
12. Do you use any tools for modelling?
13. Do you do any modelling?
14. Are you using any modelling techniques?
15. Have you worked with junior BAs? What is your expectations from them in terms of SA&D practices?
16. What skills are needed for new entrants to gain employment in to industry as BA?
17. Have you ever felt that there is a gap in academia and industry?

Interview guide for Academia

1. What is your role in teaching this paper?
2. How long you have been in this paper?
3. Under what program do you teach this course?
4. What is the duration of this course?
5. What are the contact hours for the course?
6. What contents what do you teach?
7. How many lectures do you spend on SDLC?
8. What do you cover in requirement elicitation and modelling?
9. What techniques do you teach?
10. What is your assignment structure?
11. What do they have submit?
12. Do you use any tools in delivering the course?
13. Do you think universities are helping students for industry?
14. In terms what you teach, does it meet the needs of industry?

Appendix F

	Uni#1	Uni#2	Uni#3	Uni#4	Uni#5	Uni#6
Name of the Course	Data & Process Modelling Business System Analysis	Systems Analysis & Modelling	Systems Analysis	Business Systems Analysis	Systems,Analysis, Design,& Modelling	
Major	Information Technology	Information System	Information Technology	Information System	Information System	Information System
Taught Under	School of Computer & mathematical Sciences	School of Business	College of Science	School of Business	School of Information Management, part of Commerce Faculty	School of Business
Point Sys	15	15		15	15	18
Contact Hours	1 Hour/Week	Not Available	Not Available	1 Hour 40 Mins/Week	2 Hours/Week	1 Hour 40 Mins/Week

Learning Outcome # 1	i. Describe the foundational concepts associated with an iterative and incremental approach to object-oriented analysis and design.	i. Describe the different Information Systems development methodologies, and the major SDLC used in developing Information Systems and the considerations involved in choosing the appropriate one;	i. Describe the fundamental concepts and techniques of systems analysis.	i. Perform systems analysis and design activities with focus on business value	i. Recognise and apply key modelling concepts applicable to both structured and object-oriented approaches to systems development.?	i. To understand the principles of and motivation behind systems analysis and design and where they fit in the context of systems development.

# 2	ii Analyse textual descriptions related to different problem scenarios in order to develop appropriate UML models to represent user requirements.	ii Identify the major activities and deliverables involved in each phase of the basic system development lifecycle (SDLC);	ii Apply the basic principles and techniques for eliciting, gathering and documenting the functional requirements of an IT system.	ii Manage the relationships between business and systems requirements	ii Identify and understand key aspects of the systems development process, from project planning through to analysis, design and implementation including key activities and key deliverables. ?	ii To understand how to successfully gather, analyse, and manage user requirements in preparation for system design.

# 3	iii Understand the benefits of modelling and how it supports the analysis and design process in systems development	iii Perform the major activities involved in the analysis phase of a basic system development lifecycle (SDLC) applying the appropriate strategies and techniques;	iii Discuss and explain different software development life cycle models and issues concerned with the development of high quality software.	iii Manage the relationships between requirements determination, analysis, design, and communication	iii Recognise and perform key activities associated with the analysis of business systems, including the analysis of business problem, determining information needs, and selecting an IS-based strategy/solution to address the business need.?	iii To be conversant with two approaches to system design. These are:

# 4	iv. Select appropriate models and diagrams for a given problem domain.	iv. understand the set of typical tasks and techniques used by BSAs to design an information systems solution to meet business requirements; and be able to read models and create simple models to design a solution;	iv. Incrementally and iteratively use the Unified Modelling Language (UML) to model systems from requirements.	iv. Use BPMN and UML, the languages and associated diagrams, as support to analysis and	iv. Select and apply various strategies, tools, and modelling concepts and techniques related to traditional, object-oriented and other approaches (e.g. develop Entity-Relational/Class models, context/data flow diagrams, use cases) to the analysis and design of a business information system.	a. The traditional, data centred approach in which techniques such as ERDs are used.

# 5.	v. Produce analysis and design models for a problem domain using appropriate UML models	v. Analyse a business need for information and develop an appropriate strategy to solve the problem to provide the required information solution;	v. Demonstrate practical skills in the use of at least one Computer Aided Software Engineering (CASE) tool.	a. Design activities	v. Recognise and apply key principles of good user interface design.	a. The more recent object-oriented approach centred around use case and class diagrams of the UML (Version 2.4).

# 6.		vi. Produce the requisite systems documentation at each point in the analysis of an information system, and to do so with clarity and completeness;	vi. Apply object-oriented principles and architectures to the specification of software systems	a. Design the major types of BPMN and UML diagrams using visual Tools	vi. Work as a team to perform key activities associated with project planning and the analysis and design of an IS solution to a business problem or need.	vi. To have critically evaluated the strengths and weaknesses of each approach and understand when to use one in preference to the other.
# 7.		vii. Design and develop a user interface; and			vii. Use various tools (e.g. MS Project, MS Visio) to support the planning, analysis, and design of an IS project	vii. To be conversant with the use of software modelling tools that each support a particular design paradigm

# 8		vii. Work as a team identifying and applying the basics of project management.				
Assessment	1. Multiple Tutorial Ex 10% 2. Assignment (group/individual) 40% 3. Examination: 50%	1. Individual Assignments (multiple deliverables) 20% 2. Test 20% 3. Group Assignment (multiple deliverables) 20% 4. Presentation 5% 5. Labs 5% 6. Final Exam 30%	1. Individual 35% 2. Group Assignment:15 3. Final Exam:50	1. Not Available	1. Group Project 25% 2. Mid-Semester Test 35% 3. Final Exam/Quiz 40%	1. Deliverable 1: 10% 2. Deliverable 2 :15% 3. Deliverable 3 :10% 4. Assignment: 10% 5. Final Exam:55%

Table F.1: A snapshot of the curriculum taught in different universities in New Zealand

Appendix G

NVivo output This is the NVivo output that is showing how many times each participant

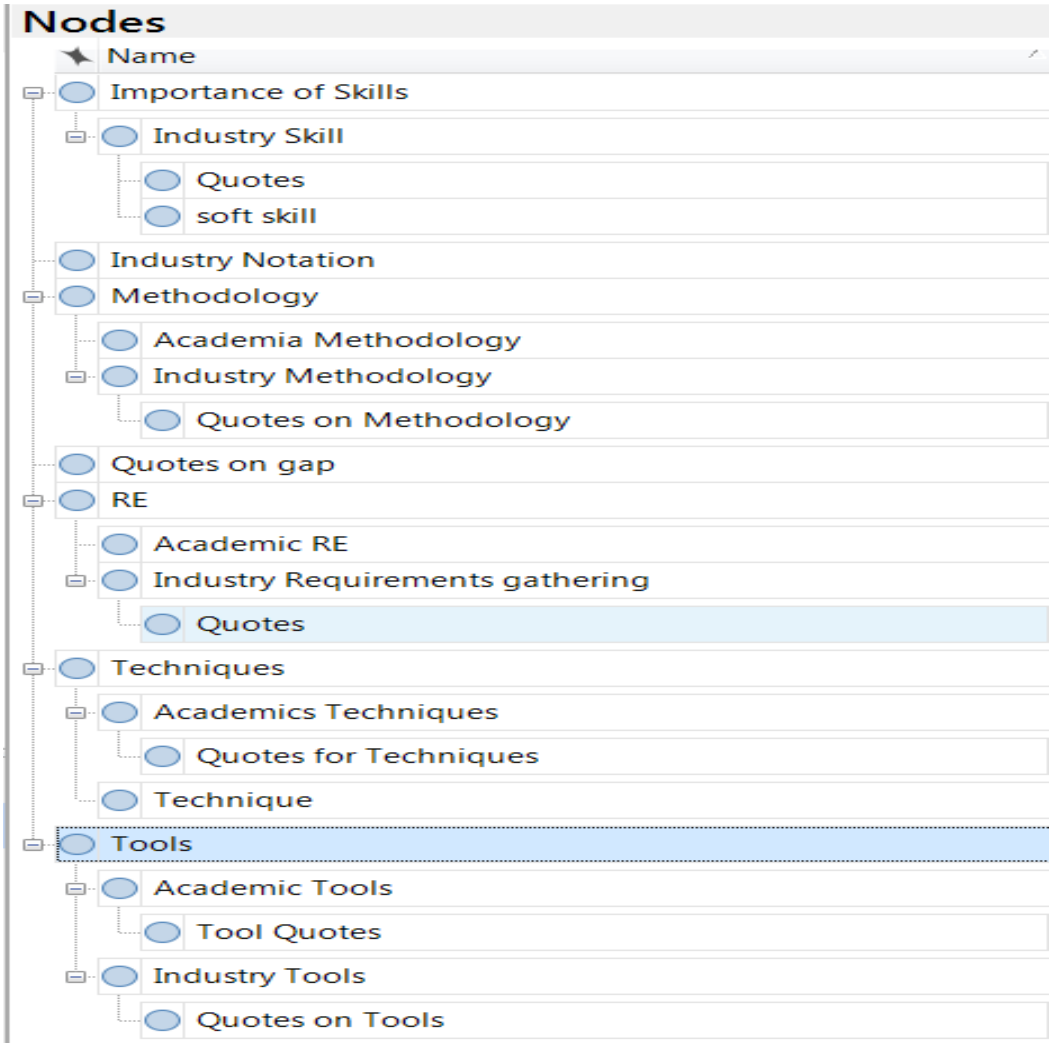
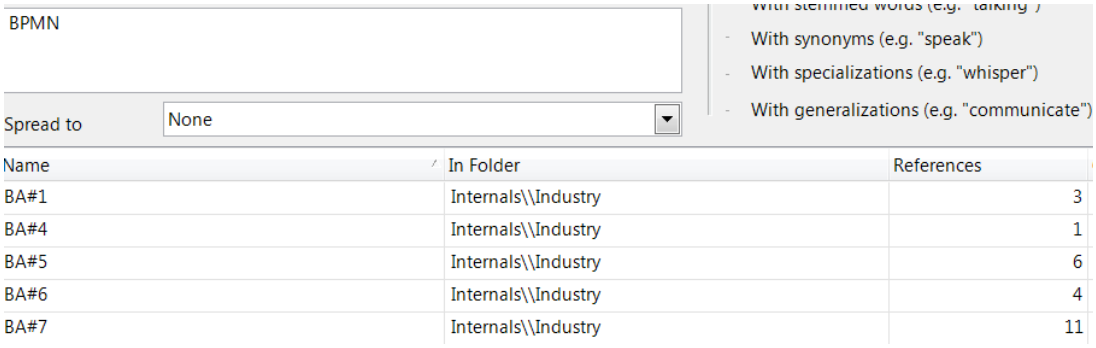


Figure G.1: *NVivo Nodes*

mentioned BPMN


BPMN

With synonyms (e.g. "speak")

With specializations (e.g. "whisper")

With generalizations (e.g. "communicate")

Spread to None

Name	In Folder	References
BA#1	Internals\\Industry	3
BA#4	Internals\\Industry	1
BA#5	Internals\\Industry	6
BA#6	Internals\\Industry	4
BA#7	Internals\\Industry	11

Figure G.2: *NVivo Output showing References for BPMN in Industry*

=

==========

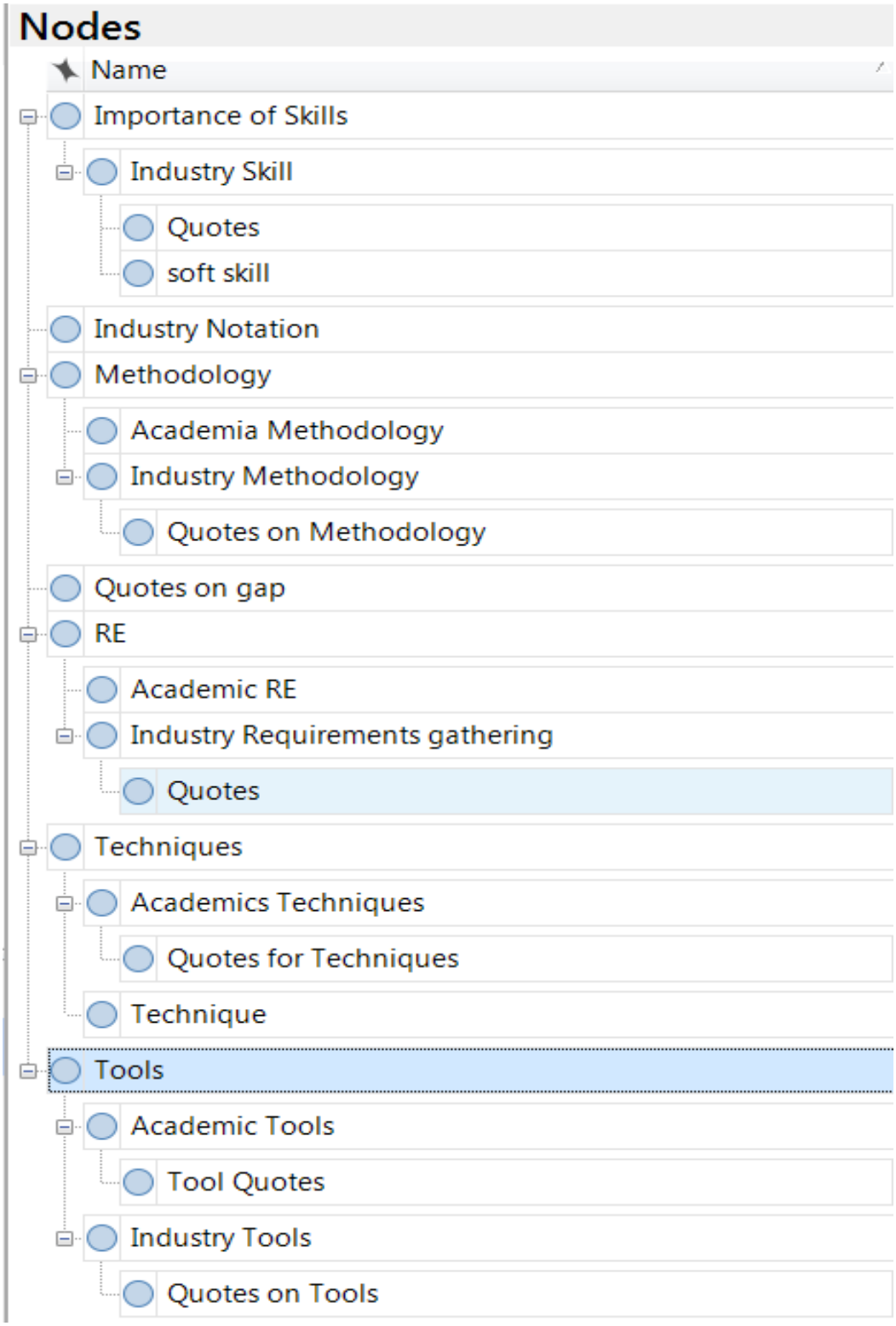


Figure G.3: *NVivo Word Cloud*

4.5: **Modelling Techniques Taught At NZ Universities**

ber	Name of the Modelling Techniques
	Entity Relationship Diagram
	Class Diagrams/Component Diagrams
	Data Flow Diagram
	Context Diagrams and Context Map(c-map)
	Use Case Description
	Use Case
	Activity Diagram
	Behaviour Diagrams(Sequence and State Machine)

Table 4.9: **Modelling Techniques used In the Industry**

No.	Modelling Techniques used in Industry	Purpose
# 1.	Model Data	ERD
		Class diagram
#2.	Process	Data flow diagram
# 3.	Behaviour	Context diagram
		Use Cases
		Use Case Descriptions
		Activity/Process/workflow diagram

Table 4.6: **Tools used At NZ Universities**

mber	Name of Requirements gathering Tool	Name of the Modelling Tools
	MS-Office	SA UML
	MS-Excel	Rational Rose
		Star UML
		Visual Paradigm
		Aris express
		Visual Express
		Visio
		Lucid Chart
		UMLet

Table 4.10: **Tools used For Requirement Elicitation In Industry**

Name of the Tool	# Participants Used
Enterprise Architectural Tools	5
MS-office Tools	4
MS-Visio	5
Paint	2
Trello	2
Valerie	1
AHA	1
JIRA	1
MEGA	1

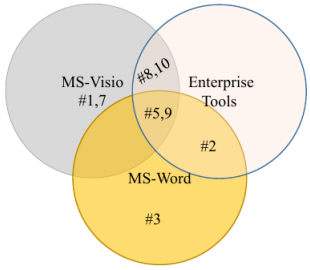


Figure 4.1: Distribution of Three common Tools Used For Requirement Elicitation In NZ Industry

3. **BlueWorks Live** -It is an IBM BPMN compliant tool
4. **Lucid chart** - It is an open Source web based diagrammatic tool that allows to create UML techniques
5. **MS-Visio** - Microsoft diagrammatic tool
6. **Enterprise Tools** - It is a modelling and building applications tool that can

Figure G.4: *Comparison of Tools& Techniques between Academia and Industry*