# Agile Effort Estimation Usage in the Sri Lankan Software Industry

Sean Jonathon Lee and Mali Senapathi(✉)

Auckland University of Technology, Auckland, New Zealand
mali.senapathi@aut.ac.nz

**Abstract.** Accurate effort estimation in the software development industry remains a significant challenge due to requirements' complexities, technology variability, and insufficient skilled members. To provide up-to-date insights on the state of effort estimation practices in Sri Lanka, we surveyed agile practitioners to identify the effort estimation techniques, types of metrics employed, levels of accuracy, and reasons for inaccuracies in estimates. Our analysis of 93 valid responses reveals that Planning Poker was the most popular estimation technique at 50.5%, while story points were the most widely used metric utilized by 61.3% of participants. Expert estimation was employed by 30% of respondents, and man-hours were used by 23.7%. The combination of Planning Poker and story points was most used, with a prevalence of 70.2%. Regarding the accuracy of estimations, respondents who used a combination of Planning Poker (61.7%) and expert estimation (25.5%) could complete their work within the estimated time without any extra effort. The top three categories of inaccurate estimates were quality-related, project management, and team-related issues.

**Keywords:** Effort estimation · Agile software development · Software engineering

## 1 Introduction

Effort estimation accurately predicts the effort, cost, and duration needed to schedule work items [1] effectively. During planning an iteration, such as a sprint, an agile software development (ASD) team selects a set of work items (e.g., task, user story) to include in the sprint, ensuring that the total estimated effort for these items aligns with the sprint's capacity [1]. Sprint capacity refers to the total effort a team can dedicate to work during a sprint, which is determined based on the estimated effort involved in the work items completed in previous sprints [1]. For effective sprint planning, it is crucial that the estimated effort accurately represents the size or development time required for each work item. Timely delivery of high-quality software begins with an accurate estimation process. The estimation includes identifying necessary resources, such as manpower, assessing dependencies, recognizing potential risks, estimating the time needed to deliver a quality product or feature, and the associated costs [2]. Accurate software estimations offer numerous benefits. They enable decision-makers within an organization to make informed choices regarding project feasibility, optimal resource allocation,


S. Peter et al. (Eds.): XP 2025, LNBIP 545, pp. 195–209, 2025.
https://doi.org/10.1007/978-3-031-94544-1_14

and effective risk management [3]. Furthermore, reliable estimates can assist in determining initial budget requirements and delivery timelines, facilitating alignment among all stakeholders [4]. However, factors such as customer pressure, varying demands, and outdated estimation methods often lead to overly optimistic estimates, affecting software delivery, quality, and the budget allocated for development [5]. Effort estimation in ASD, presents particular challenges due to the constantly evolving requirements, as the estimates must be progressively adjusted for each sprint to ensure timely delivery [6]. Despite the existence of various estimation techniques within ASD, achieving accuracy continues to be a major hurdle.

While effort estimation has been extensively studied in structured software development such as the waterfall method, there has been a recent rise in interest, particularly within the ASD community, to better [2, 7, 8] understand and improve the effort estimation practices [9]. Most research on Agile effort estimation has focused on conducting a large number of systematic literature reviews (SLRs)) [10–12], and a limited number of case studies [9, 13], and a few surveys such as [14, 15] to explore the state of estimation practices (e.g., techniques, metrics). Most studies have not examined the accuracy of estimations or the reasons behind inaccurate estimations [1]. Recent research indicates that a high accuracy rate in effort estimation significantly increases the likelihood of delivering a successful and high-quality product [16]. On the other hand, inaccurate estimates can negatively affect software project development. This can lead to two outcomes: underestimations may cause project terminations due to exceeded budgets and schedules, while overestimations can result in wasted resources [17]. By identifying the common causes of these inaccuracies, we can address the fundamental issues in effort estimation and ultimately aim to enhance prediction accuracy.

To provide empirical evidence about the state of effort estimation practice, level of accuracy, and the relationship between accuracy and estimation technique, and to contribute to providing more generalizable findings, we conducted an exploratory survey to address the following questions: (a) What agile methodologies are used in the Sri Lankan software industry? (b) What estimation techniques are used? (c) What estimation metrics are used to capture effort estimates? (d) What is the level of accuracy of the estimations? (e) What are the reasons for inaccurate estimation?

The survey study conducted in 2024 collected over ninety-three responses from Sri Lankan software practitioners. The results of the study are particularly noteworthy given the growing popularity of Business process outsourcing (BPO) services in the South Asian region, which includes countries such as India, Sri Lanka, and Bangladesh [18]. BPO services range from low-cost, repetitive tasks to knowledge-intensive value-added service providers utilized by Western nations such as the USA, Canada, Europe and the UK. According to Statista, a global company specializing in business intelligence through data collection and processing, the BPO market in South Asia is projected to reach a revenue of US$8.99 billion in 2024 [18]. With the growth of the BPO industry in this region, the Sri Lankan software sector primarily comprises companies that offer offshore software development services [19]. Accurately predicting project timeframes is not merely a project management activity; it is vital in BPO services where timely delivery is paramount.

The remainder of this paper is structured as follows: Sect. 2 reviews studies related to agile effort estimation. Section 3 outlines the research methodology, including the data collection process and the survey design. Section 4 presents the results. Finally, Sect. 5 concludes the paper, discussing the study's limitations.

## 2 Related Work

Most related work in agile effort estimation consists of SLRs [10–12, 16, 20], but empirical research providing detailed insights into estimation in ASD settings is limited, with a few exceptions, including case studies [9, 13] and surveys [14, 15].

Usman et al. [10] conducted an SLR on effort estimation in ASD based on analyzing 25 primary studies. They found that subjective estimation methods, such as expert judgment, planning poker, and Use Case Points (UCP), are commonly employed for agile estimation. While UCP and Story Points (SP) were the most frequently used size metrics, Mean Magnitude of Relative Error (MMRE) and Magnitude of Relative Error (MRE) were used as accuracy metrics. Many of the techniques examined did not achieve acceptable prediction accuracy, meaning the estimated values were often far from the actual values. More significantly, the authors concluded that practitioners would find little useful guidance on effort estimation in ASD from the current literature [12]. This is due to the low accuracy of these techniques and the lack of consensus on the appropriate cost drivers in different Agile contexts.

Two updated SLRs were published based on Usman et al.'s [10] original study. The first study used a forward snowballing approach to select 24 new papers published from 2014 to 2017 [11]. The second study analyzed the data extracted from 73 new papers published between 2014 to 2020 [12]. The findings revealed that expert-based estimation methods (e.g., Planning poker) and size metrics (e.g. story points) played an important role in six agile methods: Scrum, eXtreme Programming and four others. While achieving accuracy remains challenging, the updated SLRs observed some improvements. On the one hand, while an increasing number of studies reported acceptable accuracy values, many still demonstrate inadequate results [12]. On the other hand, nearly 29% of the papers that included accuracy metrics also addressed aspects related to model validation, and 18% reported effect sizes when comparing models [12].

Another SLR based on 12 studies published between 2000 and 2015 examined the performance of effort estimation methods, their objectivity, the factors influencing these estimations, and the accuracy of the different methods and approaches [20]. As with previous SLRs, most primary studies employed subjective expert effort estimation techniques, including Planning Poker, Expert Judgment and Story Points. Estimation by analogy was also commonly employed.

A systematic mapping study on effort estimation in ASD analyzed 25 primary studies published between 2018 and 2022 [16]. The findings included commonly used expert-based estimation methods/techniques, such as Story Point Estimation and Planning Poker. MMRE, Prediction Evaluation (PRED), and Mean Absolute Error (MAE) were identified as the most frequently employed performance evaluation measures. The study also identified several challenges and factors that complicate the estimation process, such as feasibility, experience, and the delivery of expert knowledge.

A survey of data collected from 60 agile practitioners from 16 different countries, found that (i) Planning poker (63%), analogy (47%) and expert judgment (38%) were the frequently employed estimation techniques; (ii) Story points, the most frequently (62%) employed size metric; (iii) Team's expertise level and prior experience were the most commonly used cost drivers, and (iv) 52% of the respondents believed that an error under/overestimated their effort estimates on average [10]. Similar findings were reported in a recent survey of 53 valid responses from agile practitioners [6]. Tanveer et al. [21] conducted a case study involving three agile teams in a German multinational company to examine the estimation process and its accuracy in ASD. Their findings highlighted that various factors—such as the developer's knowledge, experience, and the complexity and impact of changes on the underlying systems significantly influence both the magnitude of estimates and estimation accuracy.

In summary, most studies have focused primarily on various estimation techniques and metrics used in ASD without considering the accuracy levels and the reasons behind inadequate estimates. An exception is a recent SLR, which analyzed 82 studies published after 2001 and identified five themes that explain the reasons for inaccurate estimation: information quality, team dynamics, estimation practices, project management, and business influences [1]. To contribute to this area of literature, which has limited empirical evidence, we used an empirical approach to explore and analyze how agile effort estimation is used and experienced by organizations in the real-world software industry. Our aim was to identify the techniques and metrics employed, as well as the accuracy of estimations.

## 3 Research Methodology

This section provides an overview of the research setting, including details about data collection and survey design.

### 3.1 Data Collection

After receiving approval from the university's ethics committee, the survey was launched on March 1, 2024, and remained active for the entire month. An open invitation outlining the study's purpose was posted on the first author's LinkedIn profile, inviting software practitioners to participate. A reminder was posted two weeks after the survey opened. Consequently, participant recruitment was based on availability, utilizing a convenience sampling method. Although this approach has drawbacks and potential biases, it is still acceptable. Convenience sampling is reported to be the dominant method used in surveys within the field of software engineering [22]. Of the 109 responses received, 93 were deemed valid after omitting missing and incomplete data values.

### 3.2 Survey Design

The survey was developed using the Qualtrics survey tool. It consisted of fifteen questions based on relevant literature and empirical studies on agile effort estimation, including a recent study on effort estimation in ASD [6]. It included a participant information sheet and a consent form that thoroughly outlined the project's topic, goal, and privacy policies. A draft questionnaire was developed based on the guidelines proposed by Molléri et al. [23]. The questionnaire was then piloted by a senior academic and a senior software professional from Sri Lanka to check its consistency and legibility. The questions were divided into the following sections:

*Background information* on the practitioners and their organizations consisted of the following questions: the highest level of education, years of experience in software development, role/position in the organization, and whether they had completed an agile certification program.

*Usage of agile effort estimation* consisted of the following questions: the agile methodology used, team size, frequency of effort estimation, usage of specific estimation techniques, and usage of specific estimation metrics.

*Level of accuracy* section had questions related to the ability to complete within the estimated time and reasons for inaccurate estimates.

## 4 Results

This section presents the results and compares them to earlier studies on agile effort estimation.

### 4.1 Background Information

The number of valid responses was 93. Regarding the highest qualifications of participants, 62 respondents held a Bachelor's degree, 30 had a Master's degree, and one possessed a Diploma/Certificate level qualification. The respondents were working in various positions in their organizations. The main roles of the respondents were developers/programmers (n = 28) and quality assurance engineers (n = 23). Table 1 presents the respondents' positions, and Fig. 1 shows respondents' experiences in software development. Almost 97% of participants had more than three years of software industry experience: 18.28% had 3 to 5 years, 40.86% had 6 to 8 years, 29% had 9 to 12 years, and 8.6% had more than 13 years. 78.5% of the respondents had more than 5 years of experience, which highlights the involvement of senior practitioners in the ASD estimation process, and increases our confidence in the validity of the responses [14]. Comparatively, a recent study found that only 62.3% of respondents had more than three years of experience, with none reporting over five years of experience in ASD [6]. 72 (77.4%) respondents indicated they had completed an Agile certification program.

### 4.2 Usage of Agile Effort Estimation

As most participants were highly experienced practitioners, as expected, 88 (94.6%) participants selected that the key Scrum roles of Scrum Master and Product Owner were identified in their teams.

Regarding the type of agile methodology used (see Fig. 2), Scrum was the most commonly used methodology employed by 74 participants (79.57%), hybrid methods (e.g., Agile – Kanban) used by 10 participants (10.75%), followed by Kanban (n = 6,

**Table 1.** Position of participants

| Position | Count (n) | Percent |
|---|---|---|
| Developer/Programmer | 28 | 30.1% |
| Quality Assurance Engineer | 23 | 24.7% |
| Engineering Manager | 9 | 9.7% |
| Scrum Master/Project Manager | 9 | 9.7% |
| Software Architect | 7 | 7.5% |
| DevOps Engineer | 5 | 5.4% |
| Business Analyst | 4 | 4.3% |
| Product Owner | 3 | 3.2% |
| Other | 5 | 5.4% |
| Total | 93 | 100 |

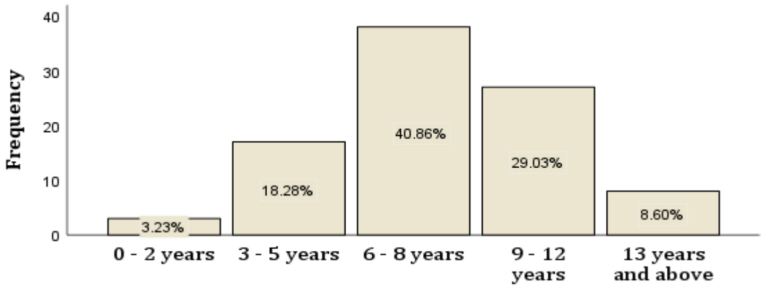


**Fig. 1.** Experience of participants

6.45%) and Waterfall (n = 3, 3.2%) was the least used. While the wide usage of Scrum and the use of hybrid approaches (e.g., Scrumban) aligns with previous surveys such as HELENA [24] and others [6], it is interesting to note that approaches such as XP (none selected) and Waterfall are rarely used in Sri Lanka.

Most respondents work in teams of up to 12 members (see Table 2); 15 participants (16.1%) reported a team size between 1 and 5, 46 (49.5%) between 5 and 8, and 20 between 8 and 12 (21.5%). Twelve reported a team size greater than 12 (12.9%).

The frequency of performing effort estimation is shown in Table 3. While most respondents (n = 68, 73.1%) reported making the estimates at every sprint, 11 said estimates were made ad-hoc, i.e., when the management or the client requested them.

**Effort Estimation Techniques**

This study included Planning Poker, Three-point estimation, Top-down estimate, Dot Voting, Expert estimation, Bucket system, and Swimlane sizing as the estimation techniques. Figure 3 details the responses to ASD estimation techniques practiced, along

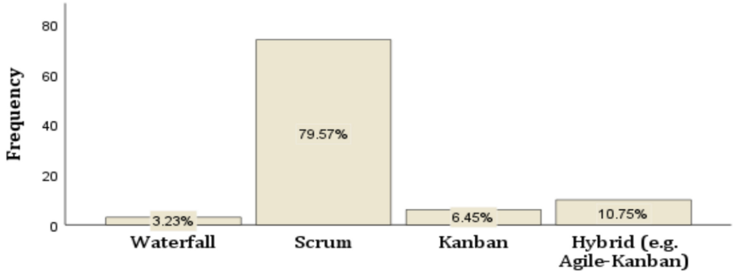


**Fig. 2.** Agile methods used

**Table 2.** Team size of participants

| Team Size | Count (n) | Percent |
|---|---|---|
| 1–5 | 15 | 16.1% |
| 5–8 | 46 | 49.5% |
| 8–12 | 20 | 21.5% |
| more than 12 | 12 | 12.9% |
| Total | 93 | 100 |

**Table 3.** Frequency of estimates

| Frequency | Count (n) | Percent |
|---|---|---|
| Every Sprint | 68 | 73.1% |
| Before a task/story/requirement | 6 | 6.5% |
| Monthly/Quarterly | 3 | 3.2% |
| After reaching a defined project milestone | 5 | 5.4% |
| Adhoc | 11 | 11.8% |
| Total | 93 | 100 |

with the corresponding frequencies and percentages. Planning poker is the most frequently used effort estimation technique, employed by 50.5% of respondents, followed by expert estimation at 30.1%. The estimation techniques used in Sri Lanka align more closely with an older study [14], which reported that 63% of participants practiced planning poker, and 38% used expert judgment. In contrast, in a more recent study [6] story points was the most popular estimation technique, utilized by 26.1% of respondents, followed by planning poker at 20.7% and expert estimation at 17.6%. It is important to note that their study [6] included story points as an estimation technique. However, we chose not to include story points in our survey because they measure the effort required to implement a product backlog item or user story fully. In contrast, story point estimation refers to assigning story points to a product backlog item or user story [25]. Nevertheless, the popularity of subjective assessment techniques like planning poker and expert judgment aligns well with the agile philosophy, which manages much knowledge tacitly and emphasizes people and their interactions [14]. Our survey did not allow participants to select multiple techniques, so we could not explore the combinations in which effort estimation techniques were practiced in Sri Lanka.

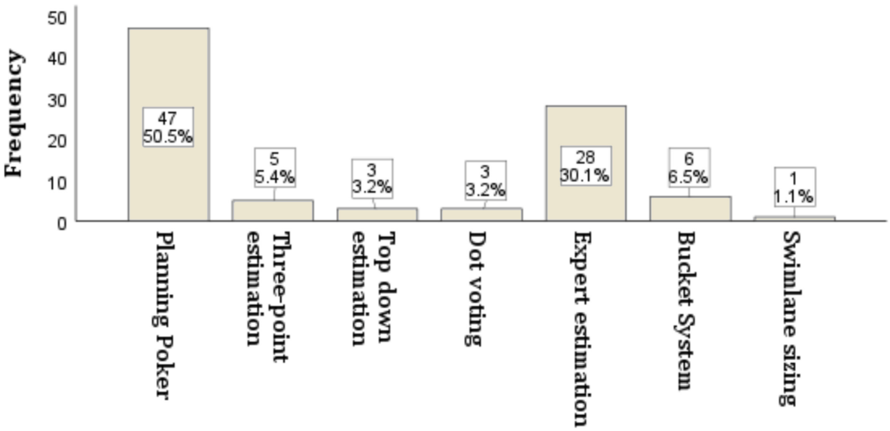


**Fig. 3.** Effort estimation techniques

**Effort Estimation Metrics**

To determine the estimation metric used by software practitioners in Sri Lanka for capturing effort estimates, the survey presented four types of metrics: man-hours (estimating the number of hours to complete a task), man-days (estimating the number of days required to finish a task), story points, and t-shirt sizing. Respondents did not report any additional metrics in the *Other* option provided. Figure 4 displays the responses regarding the estimation metrics, including their corresponding frequencies and percentages. 57 respondents, representing 61.3%, chose story points as their preferred sizing metric. This finding is consistent with the study by Usman et al. [12], which reported that 61% of participants also utilized story points. Additionally, this result aligns with the previously discussed estimation techniques, where planning poker emerged as the most commonly used estimation method.

Table 4 illustrates the use of effort estimation metrics in relation to the estimation techniques employed by survey respondents. The use of Planning Poker with story points is the most common method, with a prevalence of 70.2%. Man-hours, man-days, and T-shirt sizing are frequently utilized metrics in expert estimation techniques. Given that most participants were experienced senior practitioners, it is unsurprising that there is a higher usage of man-day and man-hour metrics within expert estimation. T-shirt sizing is also employed in Planning Poker, top-down estimation, and expert estimation

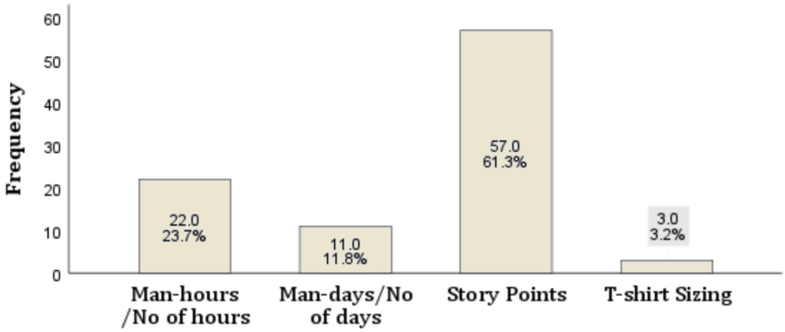


**Fig. 4.** Estimation metrics

techniques. On the other hand, Swimlane sizing and dot voting are the least commonly used methods.

### 4.3 Accuracy of Estimations

**Level of Accuracy**

This section of the questions aimed to determine the accuracy of estimations and the reasons behind inaccurate estimates. Figure 5 displays the results of the first question, which asked respondents to rate the accuracy of their estimations using five options: (i) never able to meet the estimate provided, (ii) a lower chance of meeting the estimate, (iii) able to reach the estimate with extra effort, (iv) able to complete the task within the estimate (with extra effort), and (v) always able to complete the task comfortably within the estimate.

Out of the 93 respondents, 47 (50.5%) indicated that they could complete their tasks within the estimated time without requiring any extra effort, while 33 respondents (33.5%) reported that they could meet the estimate but needed to exert additional effort. The finding that 33.5% of practitioners needed extra effort suggests a trend of underestimating the actual effort required. This aligns with existing literature indicating that the tendency to underestimate effort is more prevalent than the tendency to overestimate it. Usman et al. [14] found that 35% of their respondents reported effort estimates were, on average, underestimated by 25% or more, while 7% indicated underestimation by 50% or more. Underestimation results from various factors, including overoptimism, team members' inexperience in ASD, neglecting non-functional requirements, and disregarding test and code review efforts [1, 14].

**Accuracy Level vs Estimation Techniques**

The accuracy levels of completing tasks comfortably within estimated timeframes, as well as finishing within estimates without extra effort, can be viewed as a healthy and sustainable approach to estimation without the risk of burnout. The analysis of the accuracy level of various estimation techniques (see Table 5) shows that 61.7% (n = 29) of respondents used Planning Poker along with expert estimation 25.5% (n = 12) to complete their work within the estimated time without any extra effort. All other

**Table 4.** Estimation metrics used by estimation technique

| Estimation Metric %wi = %within Estimate Metric | | Estimation technique | | | | | | | Total |
|---|---|---|---|---|---|---|---|---|---|
| | | Planning Poker | Three-point estimation | Top down estimation | Dot voting | Expert estimation | Bucket System | Swimlane sizing | |
| Man-hours/ No of hours | Count | 5 | 4 | 1 | 1 | 9 | 2 | 0 | 22 |
| | %wi | 22.7% | 18.2% | 4.5% | 4.5% | 40.9% | 9.1% | 0.0% | 100.0% |
| Man-days/ No of days | Count | 1 | 0 | 1 | 0 | 7 | 2 | 0 | 11 |
| | %wi | 9.1% | 0.0% | 9.1% | 0.0% | 63.6% | 18.2% | 0.0% | 100.0% |
| Story Points | Count | 40 | 1 | 0 | 2 | 11 | 2 | 1 | 57 |
| | %wi | 70.2% | 1.8% | 0.0% | 3.5% | 19.3% | 3.5% | 1.8% | 100.0% |
| T-shirt Sizing | Count | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 3 |
| | %wi | 33.3% | 0.0% | 33.3% | 0.0% | 33.3% | 0.0% | 0.0% | 100.0% |
| Total | Count | 47 | 5 | 3 | 3 | 28 | 6 | 1 | 93 |
| | %wi | 50.5% | 5.4% | 3.2% | 3.2% | 30.1% | 6.5% | 1.1% | 100.0% |

estimation techniques had lower ratings for higher accuracy options. Additionally, the combination of Planning Poker and expert estimation emerged as the most popular combination of multiple estimation techniques utilized by both high and low-accuracy groups. The findings indicate that respondents frequently employed the techniques in conjunction, as they may not be effective when used independently. In the low-accuracy groups, which indicate a lower likelihood of meeting estimates and achieving them with additional effort, nearly 35% to 40% of respondents selected either Planning Poker or expert estimation techniques. While our findings suggest that using either a single estimation method or a combination of techniques may impact estimation accuracy, further investigation is required before any generalizations can be made.

**Reasons for Inaccurate Estimates**

The second question asked participants to identify the reasons for inaccurate estimates, with the option to select multiple responses. The reasons for these inaccuracies, drawn from relevant literature [1, 6], were categorized into four main groups: *Quality-Related Issues, Team-Related Issues, Project Management-Related Issues,* and *Business Influence Issues.* These reasons are discussed below and summarized in Table 6.

The *Quality-related* issues category received the most contributions, with 117 respondents. Unclear requirements or requirements that are too large to estimate, and overlooking non-functional requirements (NFR) stem from a lack of detail during effort estimation (e.g., user stories, requirements), which can lead to inaccurate estimation [26]. The accuracy of these estimates can be improved when requirements are clearly understood, and when NFRs are considered [14].

*Team-related* issues (102 respondents) such as lack of experience and insufficient stakeholder participation or absence (e.g., development team, clients, and scrum master)

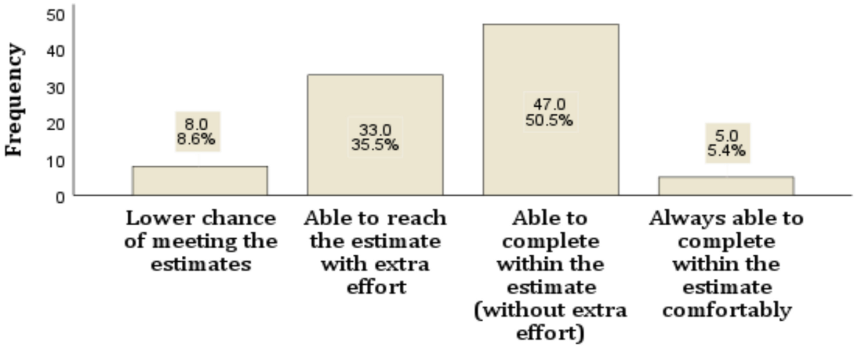


**Fig. 5.** Level of accuracy of estimations

during the estimation process have been reported to significantly influence over- and under-estimations and the team's ability to deliver [14, 26].

**Table 5.** Accuracy level vs estimation techniques

| | | | Planning Poker | Three-point estimation | Top-down estimation | Dot voting | Expert estimation | Bucket System | Swimlane sizing | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| AL | LC | Count | 3 | 0 | 0 | 0 | 3 | 2 | 0 | 8 |
| | | %wi | 37.5% | 0.0% | 0.0% | 0.0% | 37.5% | 25.0% | 0.0% | 100.0% |
| | EE | Count | 11 | 4 | 2 | 1 | 13 | 2 | 0 | 33 |
| | | % wi | 33.3% | 12.1% | 6.1% | 3.0% | 39.4% | 6.1% | 0.0% | 100.0% |
| | WEE | Count | 29 | 1 | 0 | 2 | 12 | 2 | 1 | 47 |
| | | % wi | 61.7% | 2.1% | 0.0% | 4.3% | 25.5% | 4.3% | 2.1% | 100.0% |
| | CC | Count | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 5 |
| | | % wi | 80.0% | 0.0% | 20.0% | 0.0% | 0.0% | 0.0% | 0.0% | 100.0% |
| Total | | Count | 47 | 5 | 3 | 3 | 28 | 6 | 1 | 93 |
| | | % wi | 50.5% | 5.4% | 3.2% | 3.2% | 30.1% | 6.5% | 1.1% | 100.0% |

**Legend used in Table 5:**
**AL:** Accuracy Level
**LC:** Lower chance of meeting the estimates
**EE:** Able to reach the estimate with extra effort
**WEE:** Able to complete within the estimate (without extra effort)
**CC:** Always able to complete within the estimate comfortably

The *project management* category (110 respondents) highlighted issues such as scope creep and technical challenges related to development, testing, infrastructure, and deployments following estimation. If these issues are not managed effectively, they can adversely affect development time and project costs, increasing the risk of estimation errors [14, 26].

**Table 6.** Reasons for inaccurate estimates

| | n | Percent |
|---|---|---|
| **Quality issues** | | |
| Unclear requirements or information | 65 | 70% |
| Too large to estimate (unable to slice the task into sub-tasks) | 27 | 29% |
| Overlooking non-functional requirements (e.g., performance, security) | 25 | 27% |
| **Total** | **117** | |
| **Team-related** | | |
| Inexperienced team members/New members | 30 | 32% |
| Unplanned absence of team members | 33 | 35% |
| Knowledge gaps related to agile estimations | 20 | 22% |
| Distributed team (working from home or different locations) | 4 | 4% |
| Changing team members (no fixed team as team members are allocated and removed periodically) | 15 | 16% |
| **Total** | **102** | |
| **Project Management** | | |
| Scope creep (change of requirements after estimation) | 62 | 67% |
| Technical related (facing tech challenges related to development, testing,infrastructure and deployments after estimating) | 48 | 52% |
| **Total** | **110** | |
| **Business Influence** | | |
| Purposely underestimating to obtain work/retain client | 10 | 11% |

The *business influence* issue relates to over-optimism (10 respondents), and purposefully underestimating the effort needed to secure a contract or retain a client is an unfair practice and a clear violation of the ethical code for software engineers [14].

## 5 Conclusion and Limitations of the Study

By using an exploratory survey of 93 Sri Lankan agile practitioners, this study aimed to analyze the state of agile effort estimation practice. The results indicate that Planning Poker was the most popular estimation technique at 50.5%, while story points were the most widely used metric utilized by 61.3% of participants. The combination of Planning Poker and story points was most used, with a prevalence of 70.2%. Regarding the accuracy of estimations, respondents who used a combination of Planning Poker

(61.7%) and expert estimation (25.5%) could complete their work within the estimated time without any extra effort. The top three categories of inaccurate estimates were quality-related, project management, and team-related issues.

The paper makes the following contributions to research and practice: i) it provides more generalizable and up-to-date results on the state of agile effort estimation practice in Sri Lanka, using first-hand industrial insight on how effort estimation techniques and metrics are being used in the real-world industry. The findings may be advantageously leveraged by other organizations using effort estimation through a better understanding of the methods and practices that their peers are using as well as their benefits and challenges ii) The results identify the accuracy level of estimations and reasons for inaccuracies, which has significant implications for practitioners, underscoring the importance of conducting detailed analyses or confirming information with stakeholders before making effort estimations. Such measures are especially crucial for ensuring the quality of requirements, and iii) The results serve as a foundation for future research by pinpointing the most commonly used agile estimation practices and their accuracy. These findings should inform the research agenda of other initiatives, such as case studies or experiments, that delve deeper into the specific challenges organizations encounter with agile effort estimation and lead to the development of automated approaches for improving the quality of effort estimation.

While our focus on Sri Lanka may limit the generalizability of our results, we believe that our findings offer valuable insights into the software industry as a whole, especially within the South Asian region. We consider the Sri Lankan software industry an appropriate population for this study. To obtain a broad and representative sample of our target population, we took several steps, including advertising our survey on online platforms like LinkedIn. We believe these efforts contributed to achieving a sample that is diverse in terms of experience and job roles.

## References

1. Pasuksmit, J., Thongtanunam, P., Karunasekera, S.: A systematic literature review on reasons and approaches for accurate effort estimations in agile. ACM Comput. Surv. **56**(11), 269 (2024). https://doi.org/10.1145/3663365
2. Jorgensen, M., Shepperd, M.: A Systematic Review of Software Development Cost Estimation Studies. IIEEE Trans. Software Eng. **33**(1) (2007). https://doi.org/10.1109/TSE.2007.256943
3. Qi, K., Boehm, B.W.: Process-driven incremental effort estimation, pp. 165–174 (2019). https://doi.org/10.1109/ICSSP.2019.00030
4. Rosa, W., et al.: Empirical effort and schedule estimation models for agile processes in the US DoD. IEEE Trans. Softw. Eng. **48**(8), 3117–3130 (2022). https://doi.org/10.1109/TSE.2021.3080666
5. Pospieszny, P., Czarnacka-Chrobot, B., Kobylinski, A.: An effective approach for software project effort and duration estimation with machine learning algorithms. J. Syst. Softw. **137**, 184–196 (2018). https://doi.org/10.1016/j.jss.2017.11.066
6. Sandeep, R.C., et al.: Effort estimation in agile software development: a exploratory study of practitioners' perspective, pp. 136–149 (2022). https://doi.org/10.1007/978-3-030-94238-0_8
7. Kitchenham, B.A., Mendes, E., Travassos, G.H.: Cross versus within-company cost estimation studies: a systematic review. IEEE Trans. Software Eng. **33**(5), 316–329 (2007). https://doi.org/10.1109/TSE.2007.1001

8. Shepperd, M., MacDonell, S.: Evaluating prediction systems in software project estimation. Inf. Softw. Technol. **54**(8), 820–827 (2012). https://doi.org/10.1016/j.infsof.2011.12.008
9. Tobisch, F., et al.: Investigating effort estimation in a large-scale agile ERP transformation program, pp. 70–86 (2024). https://doi.org/10.1007/978-3-031-61154-4_5
10. Usman, M., et al.: Effort estimation in agile software development: a systematic literature review. In: Proceedings of the 10th International Conference on Predictive Models in Software Engineering, pp. 82–91. Association for Computing Machinery, Turin, Italy (2014)
11. Dantas, E., et al.: Effort estimation in agile software development: an updated review. Int. J. Softw. Eng. Knowl. Eng. **28**(11n12), 1811–1831 (2018). https://doi.org/10.1142/s0218194018400302
12. Fernández-Diego, M., et al.: An update on effort estimation in agile software development: a systematic literature review. IEEE Access **8**, 166768–166800 (2020). https://doi.org/10.1109/ACCESS.2020.3021664
13. Usman, M., et al.: Effort estimation in large-scale software development: an industrial case study. Inf. Softw. Technol. **99**, 21–40 (2018). https://doi.org/10.1016/j.infsof.2018.02.009
14. Usman, M., Mendes, E., Börstler, J.: Effort estimation in agile software development: a survey on the state of the practice. In: Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering, p. 12. Association for Computing Machinery: Nanjing, China (2015)
15. Sandeep, R.C., et al.: Effort estimation in agile software development: a exploratory study of practitioners' perspective. In: Lecture Notes in Business Information Processing. (2022).https://doi.org/10.1007/978-3-030-94238-0_8
16. Rodríguez, C.A.P., et al.: Effort estimation in agile software development: a systematic map study. INGE CUC **19**(1), 22–36 (2023). https://doi.org/10.17981/ingecuc.19.1.2023.03
17. Rastogi, H., Dhankhar, S., Kakkar, M.: A survey on software effort estimation techniques. In: 2014 5th International Conference - Confluence the Next Generation Information Technology Summit (Confluence) (2014). https://doi.org/10.1109/CONFLUENCE.2014.6949367
18. Statista: Business process outsourcing - Southern Asia (2024). https://www.statista.com/outlook/tmo/it-services/business-process-outsourcing/southern-asia
19. Ranasinghe, R.K.C., Perera, I.: Effectiveness of scrum for offshore software development in Sri Lanka. In: 2015 Moratuwa Engineering Research Conference (MERCon) (2015). https://doi.org/10.1109/MERCon.2015.7112364
20. Schweighofer, T., et al.: How is effort estimated in agile software development projects? In: SQAMIA, 5th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications. Budapest, Hungary (2016)
21. Tanveer, B., Guzmán, L., Engel, U.M.: Effort estimation in agile software development: case study and improvement framework. J. Softw. Evol. Process **29**(11), e1862 (2017). https://doi.org/10.1002/smr.1862
22. Sjoeberg, D.I.K., et al.: A survey of controlled experiments in software engineering. IEEE Trans. Software Eng. **31**(9), 733–753 (2005). https://doi.org/10.1109/TSE.2005.97
23. Molléri, J.S., Petersen, K., Mendes, E.: Survey guidelines in software engineering: an annotated review. IEEE Computer Society (2016). https://doi.org/10.1145/2961111.2962619
24. Kuhrmann, M., et al., Helena stage 2 results (2018)
25. Herranz, R.: What is story point estimation*?* (2025). https://resources.scrumalliance.org/Article/story-point-estimation
26. Conoscenti, M., et al.: Combining data analytics and developers feedback for identifying reasons of inaccurate estimations in agile software development. J. Syst. Softw. **156**, 126–135 (2019). https://doi.org/10.1016/j.jss.2019.06.075