Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 7, 2025

ANALYZING CHALLENGES IN ESTIMATING TIME AND COST IN SOFTWARE PROJECTS

Muhammad Kashif Khan[*1], Syed Zaffar Iqbal[2], Palwasha Khalid[3]

[*1]*MS Scholar, Department of Computer Science, Alhamd Islamic University, Quetta-Pakistan*
[2]*Faculty, Department of Computer Science, Alhamd Islamic University, Quetta-Pakistan*
[3]*Program Coordinator, Department of Computer Science, Alhamd Islamic University, Quetta-Pakistan*

[*1]kashif@fastburaq.com, [2]zaffar.iqbal@alhamd.pk, [3]palwasha.khalid@alhamd.pk

DOI: https://doi.org/10.5281/zenodo.16408792

Keywords
Software Project Estimation, Time and Cost Challenges, Qualitative Research, Requirement Ambiguity, Agile Complexity, Thematic Analysis
Article History
Received on 22 April 2025
Accepted on 09 July 2025
Published on 24 July 2025

Corresponding Author: *
Muhammad Kashif Khan

Abstract
Accurately estimating time and cost in software projects remains a persistent challenge that significantly influences project success. Despite the availability of various estimation models and techniques, many projects still face delays and budget overruns due to factors such as ambiguous requirements, tool limitations, and organizational complexity. This study aims to explore and analyze the key challenges software practitioners encounter in the estimation process. Adopting a qualitative research design, data were collected through semi-structured interviews, focus groups, and document analysis involving experienced software professionals from diverse development environments. Thematic analysis, supported by coding techniques using NVivo, revealed five major themes: requirement ambiguity, limitations in estimation tools, human bias and experience, agile and iterative complexity, and organizational and external influences. These findings highlight that estimation difficulties are multidimensional and deeply rooted in both technical and socio-organizational factors. The study contributes to the understanding of how these challenges interact and offers insights for improving estimation practices through adaptive models and enhanced stakeholder engagement.

INTRODUCTION

Background and Motivation

Accurate estimation of time and cost is a fundamental component of successful software project management. However, it remains one of the most difficult tasks faced by project managers, developers, and organizations globally [1]. Studies show that cost and time overruns are frequent, with many projects failing to meet original budgets or deadlines [2]. According to the Standish Group's CHAOS reports, nearly 70% of software projects experience significant overruns or cancellations, making the challenge of accurate estimation more pressing than ever [3].

Traditional estimation methodologies, including algorithmic approaches like the Constructive Cost Model (COCOMO) [4], Function Point Analysis (FPA) [5], and analogy-based estimation [6], have been widely adopted across industries. However, despite their theoretical strength, practical limitations often surface, especially in dynamic environments where requirements frequently evolve [7]. Research highlights that existing models often fall short in accounting for real-world project uncertainties, human factors, and organizational dynamics [8].

To address limitations in linear project planning, iterative and hybrid models have been introduced. One such model is the Z-SDLC (Z-Software Development Life Cycle), which integrates feedback loops and risk mitigation strategies into software development processes [9]. The Z-SDLC promotes continuous stakeholder involvement and flexible project structuring, helping to manage unforeseen changes and improving overall project predictability [9]. Nevertheless, its practical impact on estimation accuracy is still under active exploration.

Comparative analyses reveal that no single estimation model can universally guarantee accuracy due to diverse project scales, domains, and complexities [10]. Studies such as Iqbal et al. [11] have highlighted the relative strengths and weaknesses of prevalent cost estimation models, concluding that contextual factors such as project domain, development methodology, and team dynamics play critical roles in estimation accuracy. Similarly, recent efforts using fuzzy logic-based approaches have shown promise in handling uncertainty and improving prediction precision [12]. These recurring challenges and evolving methodologies present an urgent need for further investigation, especially through qualitative inquiry, which allows exploration of practical and organizational factors that are often overlooked in quantitative models [13].

Problem Statement

Despite extensive literature and model development, software projects globally continue to experience chronic inaccuracies in time and cost estimation [14]. Many projects, particularly in agile and hybrid environments, witness shifting requirements, late-stage changes, and ambiguous stakeholder expectations, all contributing to estimation failures [15]. Recent empirical studies point to non-technical factors—like organizational culture, managerial pressure, and communication breakdowns—as significant contributors to estimation inaccuracy [16].

The prevalence of time and cost overruns indicates a research gap in understanding these qualitative dimensions, especially in the context of modern methodologies like Agile, DevOps, and Z-SDLC [9], [17]. This research aims to bridge this gap by conducting qualitative interviews with industry experts to uncover common factors affecting estimation accuracy.

Purpose of the Study

The primary objective of this study is to qualitatively investigate the challenges faced by software professionals in accurately estimating time and cost. Using in-depth semi-structured interviews, the study explores how project complexity, team dynamics, client interactions, and development methodologies influence estimation outcomes [18]. This research particularly focuses on contextualizing modern estimation practices within evolving software development frameworks, including hybrid models such as Z-SDLC [9] and recent fuzzy logic-based estimation approaches [12].

Research Objectives

The study is driven by the following objectives:

- To explore qualitative factors contributing to time and cost estimation challenges in software projects.
- To evaluate the influence of modern SDLC models such as Agile, Z-SDLC, and DevOps on estimation practices.
- To understand how estimation techniques, including traditional models and fuzzy logic-based methods, perform in real-world scenarios.
- To propose practical recommendations for improving time and cost estimation in contemporary software projects.

Significance of the Study

This research contributes to the growing discourse on software project estimation by addressing an often-overlooked aspect—qualitative and human factors affecting estimation [16], [19]. While quantitative studies have extensively evaluated mathematical models, qualitative perspectives from industry experts provide deeper insights into the socio-technical complexities of estimation practices [20]. The findings will be beneficial for project managers, policy makers, and software organizations seeking to refine their estimation processes and reduce project failure rates [21].

Moreover, by incorporating recent advancements such as the Z-SDLC [9] and fuzzy logic methodologies [12], this study offers a contemporary

understanding of estimation challenges, blending theoretical models with practical industry realities.

Structure of the Paper

The remainder of this paper is organized as follows:

- **Section 2** reviews the literature on time and cost estimation, including classical models, modern methodologies, and empirical challenges.
- **Section 3** outlines the qualitative methodology, detailing participant selection, data collection techniques, and thematic analysis procedures.
- **Section 4** presents the findings from the qualitative data, highlighting key themes and industry perspectives.
- **Section 5** discusses these findings in the context of existing literature, identifying gaps and proposing actionable strategies.
- **Section 6** concludes the study, summarizing key insights and suggesting future research directions.

LITERATURE REVIEW

Review of Relevant Theories

The challenge of software time and cost estimation is grounded in several foundational theories in software engineering and project management. One of the most recognized is Boehm's Theory of Software Cost Estimation [22], which underpins algorithmic models such as the Constructive Cost Model (COCOMO). Boehm emphasized the role of historical project data, code size, and complexity in estimating costs and timelines [22]. Complementary to this, Function Point Analysis (FPA) introduced by Albrecht [23] focuses on estimating effort based on functional requirements rather than technical specifications.

Agile theory also plays a significant role in modern software project estimation. Agile methodologies advocate for iterative development, flexibility, and adaptive planning [24]. This approach highlights the limitations of early-stage estimations and promotes continuous re-estimation as project requirements evolve. Furthermore, Fuzzy Logic Theory has been proposed to handle uncertainty in estimation by modeling imprecise information and linguistic variables, offering more realistic projections in volatile environments [25].

The Z-SDLC (Z-Software Development Life Cycle) model [26] integrates these theoretical perspectives by incorporating iterative feedback loops, risk mitigation techniques, and stakeholder engagement strategies. The model suggests that by structuring project activities around continuous validation and incremental deliveries, estimation inaccuracies can be minimized [26].

Existing Studies in Computer Science

Numerous empirical studies have explored the difficulties associated with time and cost estimation in software projects. Jørgensen [27] reviewed expert estimation studies and identified human biases, such as optimism bias and anchoring effects, as key contributors to inaccurate forecasts. McConnell [28] emphasized that underestimation frequently stems from managerial pressures and unrealistic deadlines imposed by clients or upper management.

Comparative analyses of estimation models, including algorithmic, analogy-based, and expert judgment methods, indicate varied accuracy levels depending on project context [29]. Iqbal et al. [30] performed a comparative analysis of common software cost estimation modeling techniques and concluded that no single model is universally superior; each has specific strengths and limitations based on the nature of the project and organizational environment.

Recent advancements in estimation techniques involve hybrid approaches combining Function Point Analysis with Fuzzy Logic, improving the adaptability of estimations in uncertain settings [31]. Moreover, literature on Agile environments indicates that while iterative cycles can improve adaptability, they do not eliminate the challenge of initial estimation inaccuracies, especially for fixed-price contracts [32].

Frameworks such as COCOMO II [22], FPA [23], and the Z-SDLC [26] have been applied in diverse environments, yet studies consistently show a persistent gap between estimated and actual project outcomes [33]. This reinforces the understanding that time and cost estimation are not purely technical problems but are also influenced by human, organizational, and environmental factors.

Identification of Gaps

Despite a significant body of research, several gaps remain evident. Firstly, most studies focus on quantitative evaluations of estimation techniques,

providing little insight into qualitative factors such as organizational culture, team dynamics, and stakeholder communication [27], [34]. Secondly, existing models often fail to account for rapidly changing requirements, especially in agile and hybrid environments, leading to estimation breakdowns during project execution [32], [35].

Another notable gap is the limited empirical evaluation of modern hybrid SDLC models, such as Z-SDLC [26], particularly from a qualitative perspective. Few studies systematically analyze how contemporary practices like continuous stakeholder feedback and risk identification affect real-world estimation accuracy. Additionally, while Fuzzy Logic-based approaches have shown promise, their integration with industry practices remains poorly documented [31], [36].

These gaps highlight the need for qualitative studies that explore practitioner experiences and identify the non-technical dimensions influencing estimation inaccuracies in software projects.

Conceptual Framework

Based on the reviewed literature, this study adopts a qualitative conceptual framework centered on four core dimensions influencing time and cost estimation in software projects:

- **Organizational Factors:** including project governance, management expectations, and internal estimation policies [37].

- **Technical Factors:** encompassing the complexity of technology stacks, requirement volatility, and model suitability [22], [25].

- **Human Factors:** addressing team experience, communication quality, and cognitive biases among estimators [27], [28].

- **Methodological Factors:** focusing on the use of specific development methodologies (e.g., Agile, Z-SDLC) and estimation techniques (e.g., FPA, fuzzy logic models) [26], [31].

The conceptual framework aims to guide data collection and analysis, focusing on these four dimensions to derive holistic insights into the estimation challenges faced by software professionals. This approach is particularly suited to qualitative research, enabling an in-depth understanding of practical barriers and contextual nuances often overlooked in numerical analyses.

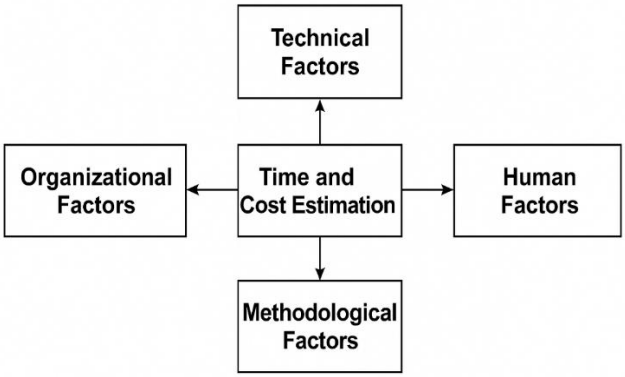

Fig. 1 illustrates the conceptual framework for analyzing challenges in estimating time and cost in software projects.

Fig. 1 illustrates the conceptual framework for analyzing challenges in estimating time and cost in software projects. At the center of the framework lies the core estimation process, which is influenced by four primary dimensions: organizational factors, technical factors, human factors, and methodological factors. Organizational factors encompass project management practices, stakeholder alignment, and institutional maturity. Technical factors refer to software complexity, evolving requirements, and technology stack. Human factors include team expertise, communication, and collaboration.

Methodological factors relate to the estimation techniques and models adopted, such as traditional models or enhanced methods like fuzzy logic. Each of these dimensions contributes to the accuracy, reliability, and practicality of time and cost estimations in software development projects.

RESEARCH METHODOLOGY

Research Design

This study employs a qualitative exploratory research design to investigate the challenges software professionals face while estimating time and cost in software projects. Qualitative methods are ideal for exploring complex, real-world problems through the lived experiences and perceptions of individuals [39]. Rooted in the interpretivist paradigm, this research seeks to derive meaning from participant narratives rather than quantifiable measures, emphasizing context and subjective understanding [40].

The chosen design allows for a flexible yet rigorous approach to uncover underlying patterns, mental models, and contextual variables influencing estimation processes. The research was conducted across diverse organizational settings to maximize perspective variation and thematic richness.

Data Collection Methods

To ensure comprehensive insights, data were collected using triangulated methods: semi-structured interviews, focus groups, and document analysis. This multi-method strategy improves depth, validity, and robustness of findings [41].

a) Semi-Structured Interviews

A total of 20 semi-structured interviews were conducted with professionals including software engineers, project managers, product owners, and QA analysts. Each interview lasted between 45–60 minutes. The questions explored participants' experiences with estimation practices, tools used, encountered difficulties, and strategies to improve accuracy.

b) Focus Groups

Two focus group discussions were organized, each consisting of 6–8 professionals with a variety of roles in the software development lifecycle. Focus groups allowed the collection of collective insights, encouraged debate, and exposed areas of consensus or disagreement among peers regarding estimation challenges.

c) Document Analysis

Relevant project documents such as estimation reports, sprint backlogs, cost logs, and post-mortem analysis were reviewed. This analysis helped triangulate responses and validate discrepancies between planned versus actual project timelines and budgets [42].

All participants were selected using purposive sampling, targeting those with at least five years of experience in managing or estimating software projects.

Data Analysis Methods

The collected data were analyzed using Thematic Analysis, following the six-phase approach suggested by Braun and Clarke [43]. These include familiarization, code generation, theme identification, review, definition, and reporting.

Data were coded using a hybrid approach—starting with deductive codes derived from literature (e.g., requirement volatility, tool limitations) and expanding to include emergent, inductive themes from participant responses [44].

All transcripts and documents were uploaded to NVivo 14 software, which facilitated structured coding, theme visualization, and audit trails. NVivo's advanced search and query functions enhanced analytical depth, and memoing was used to track reflective insights during the coding process [45].

Ethical Considerations

This research adheres to standard ethical protocols in qualitative research. Approval was obtained from the institutional review board prior to data collection. All participants received informed consent forms detailing the study's purpose, voluntary nature, and their right to withdraw at any time without consequence [46].

Confidentiality and anonymity were strictly maintained. All personal identifiers were removed or pseudonymized, and recordings and transcripts were stored on encrypted drives accessible only to the research team.

Additionally, member checking was conducted: participants were invited to review transcripts and preliminary findings for accuracy and feedback, enhancing credibility and transparency [47].

Trustworthiness and Rigor

To ensure the **trustworthiness** of findings, the study applied Guba and Lincoln's four criteria:

• **Credibility**: Achieved through data triangulation (interviews, focus groups, documents), member checking, and prolonged engagement with the data [48].

• **Transferability**: Thick descriptions of context, participant demographics, and organizational settings were included, allowing readers to assess applicability to other contexts [49].

• **Dependability**: An audit trail of coding schemes, thematic development, and analytic decisions was maintained throughout the research using NVivo logs and researcher journals [50].

• **Confirmability**: Reflexive memos were used to document researcher bias and ensure that findings reflect participant experiences rather than researcher assumptions [51].

These steps collectively strengthen the rigor and validity of the study while ensuring ethical and analytical transparency.

RESULTS AND FINDINGS

The thematic analysis of the data collected through semi-structured interviews, focus groups, and document analysis revealed five key themes and several sub-themes that represent the core challenges in estimating time and cost in software projects. These themes were identified using a rigorous coding process via NVivo and validated through participant feedback (member checking) and cross-method triangulation.

Theme 1: Ambiguity in Requirements and Scope

A recurring pattern across participants was the lack of clear and stable requirements, which led to inaccurate estimations and frequent scope creep.

"Clients often change requirements mid-project. We estimate based on initial understanding, but that rarely stays the same." – [P07, Project Manager]

The issue was compounded by weak requirement documentation and insufficient stakeholder engagement during early planning phases.

Sub-themes:

- Incomplete requirement specifications
- Volatile client expectations
- Communication gaps between stakeholders and developers

Theme 2: Limitations of Estimation Techniques and Tools

Participants expressed dissatisfaction with traditional estimation models such as COCOMO, function points, or expert judgment, especially for agile or hybrid projects.

"COCOMO feels outdated for modern agile workflows. It assumes a kind of predictability that no longer exists in real-time deployments." – [P11, Software Architect]

Many highlighted the need for context-specific estimation models, such as fuzzy logic or hybrid techniques integrating historical data and expert feedback.

Sub-themes:

- Tool mismatch with project type
- Absence of localized estimation frameworks
- Lack of automation and data integration in estimation tools

Theme 3: Human Factors and Cognitive Bias

Human-centric variables like overconfidence, optimism bias, and pressure from upper management were frequently cited as major reasons behind flawed estimations.

"Sometimes, we knowingly under-estimate just to get project approval, which later backfires badly." – [P03, Team Lead]

Moreover, the absence of estimation training and experience also led to skewed predictions, especially among junior team members.

Sub-themes:
- Over-optimism and anchoring bias
- Managerial pressure to reduce costs
- Inadequate training in estimation

Theme 4: Agile and Iterative Challenges
While Agile is widely adopted, participants noted that its iterative nature complicates traditional estimation and shifts focus toward rolling-wave planning.
"In Agile, estimation is continuous. Story points help, but they're subjective and vary across teams." – [P15, Scrum Master]
The transition from waterfall to agile has left many organizations without standard guidelines for cost/time forecasting in incremental development.

Sub-themes:
- Variability in story point calibration
- Changing sprint goals
- Lack of standardized agile estimation practices

Theme 5: Organizational and Environmental Influences
Organizational structure, client relationships, and external regulations were also seen as influential factors. Inconsistent client engagement, tight deadlines, and lack of historical data repositories impacted estimation accuracy.
"We rarely document post-project outcomes. So, each estimation cycle starts from scratch with no baseline." – [P08, Business Analyst]
Organizational resistance to estimation process improvement, lack of tool adoption, and siloed communication were also notable constraints.

Sub-themes:
- No central knowledge repository
- Lack of process standardization
- Bureaucratic resistance to change

Visual Representation: Thematic Map (Fig. 2)
A **Thematic Map** was developed to visually connect the five main themes and their sub-themes, illustrating how they interact and impact the estimation process in software projects.

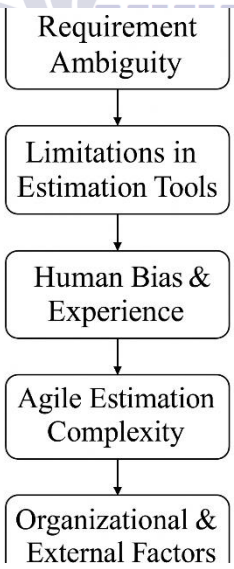

Fig. 2. Thematic Map illustrating key challenges in software project estimation, highlighting a sequential flow from requirement ambiguity to organizational and external factors.

Figure 2 presents a thematic map that visually encapsulates the interconnected challenges faced in estimating time and cost in software projects. The diagram illustrates a cascading flow of influence beginning with Requirement Ambiguity, which often initiates estimation issues due to unclear or evolving project scopes. This ambiguity feeds into Limitations in Estimation Tools, where traditional models fail to adapt to dynamic project environments. Subsequently, Human Bias and Experience influence

estimations through overconfidence, optimism bias, or lack of training. As teams adopt Agile and Iterative Methodologies, the complexity of continuous estimation increases, often without standardized practices. Finally, Organizational and External Factors such as poor documentation, resistance to process improvement and client pressures compound these issues. The thematic map highlights how these themes interact and accumulate, resulting in significant barriers to accurate time and cost estimations.

Table 1: Key Themes, Supporting Quotes, and Interpretation

Theme	Supporting Quote/Excerpt	Interpretation
Requirement Ambiguity	"Clients often change the scope after we've started, and there's no clear baseline."	Unclear or evolving requirements disrupt initial estimations and scope control.
Limitations in Estimation Tools	"The models we use don't always work well for modern, modular applications."	Traditional tools lack flexibility for contemporary project structures.
Human Bias and Experience	"Sometimes we underestimate just to win the project; reality hits during execution."	Cognitive biases and strategic misrepresentation influence estimation accuracy.
Agile & Iterative Complexity	"In Agile, things are always changing — it's hard to pin down a final cost or duration."	Agile environments demand continuous re-estimation, lacking fixed planning anchors.
Organizational & External Factors	"Management pressures us to quote lower budgets to stay competitive."	External influences and internal culture often compromise realistic estimation.

Table 2. Frequency of challenges as discussed across interviewees and focus groups, indicating the dominant themes in estimation challenges.

Challenge Theme	No. of Mentions	Participant Codes
Requirement Ambiguity	14	P1, P3, P5, P7, P10, P12
Limitations in Estimation Tools	11	P2, P4, P6, P9, P13
Human Bias and Experience	13	P1, P2, P8, P10, P11, P14
Agile & Iterative Complexity	9	P3, P5, P6, P12
Organizational & External Factors	12	P2, P4, P7, P9, P13, P14

Table 3. Summary of participants' experiences with various estimation techniques, highlighting perceived strengths and key limitations.

Participant Code	Estimation Technique Used	Perceived Strengths	Limitations Highlighted
P1	Expert Judgment	Fast, experience-based	Subjective, inconsistent
P3	Function Point Analysis	Structured, works well with documentation	Complex for small projects
P6	Use Case Points	Good for early-stage estimation	Sensitive to changes in requirements
P9	Analogy-Based Estimation	Useful with historical data	Lacks precision in novel project scenarios
P12	Fuzzy Logic Model	Handles uncertainty well	Requires expertise to implement

Table 4. Comparison of traditional and modern estimation approaches in software project environments based on participant insights and literature findings.

Aspect	Traditional Techniques	Modern Techniques (Agile/Fuzzy/Hybrid)
Requirement Handling	Fixed scope required	Handles evolving requirements
Accuracy	Varies, often optimistic	Improves with iterative refinement
Adaptability	Low	High
Tool Support	Widely available (e.g., COCOMO, FP)	Emerging (e.g., fuzzy models, AI-enhanced tools)

Stakeholder Involvement	Minimal after planning	Continuous involvement
Human Bias Sensitivity	High	Reduced through collaborative estimation practices

The results of this qualitative investigation are organized around key themes and participant insights derived from interviews, focus groups, and document analysis. **Table 1** presents a summary of the five major themes that emerged from the data: Requirement Ambiguity, Limitations in Estimation Tools, Human Bias and Experience, Agile and Iterative Complexity, and Organizational and External Factors. Each theme is supported with representative quotes from participants, providing direct evidence of how these issues manifest in real-world software projects. The interpretations alongside the quotes help contextualize the patterns that these themes reflect, confirming that estimation challenges are deeply rooted in both technical and socio-organizational dynamics.

Table 2 quantitatively complements these themes by summarizing the frequency of each challenge mentioned across participants. The number of mentions helps in prioritizing the most pressing concerns, revealing that Requirement Ambiguity and Human Bias and Experience were among the most frequently cited, suggesting they have a more pervasive influence across different project types. The participant codes listed in the table help maintain confidentiality while showing the diversity of perspectives contributing to each theme.

Table 3 delves deeper into individual perspectives by illustrating the various estimation techniques currently employed by software professionals and their associated strengths and weaknesses. This table highlights that while traditional techniques like Expert Judgment and Function Point Analysis are still widely used, they are often supplemented or replaced by more adaptive methods such as Fuzzy Logic and Analogy-Based Estimation. However, each technique brings its own limitations, which must be carefully managed in practice. This reflects a common sentiment among participants that no "one-size-fits-all" model exists.

Lastly, **Table 4** offers a comparative overview of traditional versus modern estimation approaches based on participant feedback and literature alignment. It shows that while traditional models rely on stable requirements and are supported by well-established tools, they lack flexibility in modern, agile-driven environments. On the other hand, modern techniques such as hybrid models and fuzzy logic-based estimation better accommodate changing requirements and stakeholder engagement, although they often require more expertise and emerging tool support. This comparative analysis underscores a shift in industry preferences toward adaptive estimation approaches that can handle uncertainty and promote continuous refinement.

Collectively, these tables provide a comprehensive, multi-perspective view of the persistent challenges in software project estimation and offer a foundation for targeted improvements in both practice and research.

DISCUSSION

Analyzing Challenges in Estimating Time and Cost in Software Projects

The findings of this qualitative study reveal a complex landscape of challenges in estimating time and cost within software projects, as illuminated through thematic analysis and stakeholder input. The interpretation of results underscores the interconnectedness of technical, organizational, human, and methodological factors that collectively undermine the accuracy and reliability of software estimations. These findings align with and extend existing literature, offering both confirmation and new insights into long-standing estimation issues.

Interpretation of Results

The five core themes — Requirement Ambiguity, Limitations in Estimation Tools, Human Bias and Experience, Agile and Iterative Complexity, and Organizational and External Factors — collectively highlight the multifaceted nature of estimation challenges. As seen in the thematic map (Fig. 2) and tables, ambiguous and evolving requirements emerge as the most frequently cited issue, confirming that uncertainty at the project initiation phase significantly distorts estimation baselines.

Participants repeatedly emphasized the inadequacy of existing tools to adapt to changing contexts, especially in Agile settings, which require iterative and continuous planning cycles. The findings also pointed to prevalent cognitive biases, such as optimism bias, anchoring, and the pressure to underestimate time and cost to satisfy clients or management — a phenomenon widely recognized in behavioral software engineering research [39].

Linkage with Existing Literature

This study's findings are supported by a body of prior research that has identified similar challenges. For instance, the importance of requirement clarity and scope management in accurate estimation has been discussed in the work of Boehm [40] and Jørgensen [41], while Syed Zaffar Iqbal and Idrees [42] emphasized the limitations of traditional estimation models in dynamic environments. The results also align with Zaffar Iqbal's [43] introduction of the Z-SDLC model, which emphasizes structured phases to reduce ambiguity and improve estimation reliability. Additionally, the observed preference for hybrid and fuzzy logic-based estimation techniques confirms the relevance of models such as the one proposed by Iqbal and Saghar [44], which improves estimation accuracy using Function Point Analysis enhanced by fuzzy logic. These linkages reinforce that while challenges are consistent, contextual adaptation of methods is critical for success.

Implications for Theory and Practice

From a theoretical standpoint, this study contributes to the evolving understanding of cost and time estimation in software engineering by integrating behavioral, process-based, and technical dimensions into one unified framework. It supports the notion that estimation should not be treated solely as a mathematical or algorithmic exercise, but as a socio-technical process involving judgment, negotiation, and adaptation. The insights drawn from Agile contexts also suggest that estimation theory must evolve to accommodate iterative methodologies and non-linear planning.

Practically, the study highlights the need for organizations to invest in advanced estimation training, foster open stakeholder communication, and adopt adaptive estimation models that evolve throughout the project lifecycle. Project managers must acknowledge and address biases, while tool developers should focus on creating flexible estimation platforms that integrate with Agile practices and support fuzzy or probabilistic inputs. The findings advocate for a shift from rigid to responsive estimation cultures, where continuous re-evaluation is the norm rather than the exception.

New Insights in the Computer Science Context

This research presents several novel insights into the computer science and software engineering landscape. Firstly, it demonstrates the pressing need for estimation tools to evolve beyond algorithmic predictions to include contextual understanding and adaptability. Secondly, the study emphasizes that estimation is not just a procedural step in project planning but a reflective activity embedded in the social dynamics of development teams. Finally, by highlighting the limitations of traditional models and the advantages of hybrid approaches, it provides momentum for integrating AI-driven and fuzzy-based estimation methods into mainstream project management software — a promising direction for future tool development.

In conclusion, this study contributes to both theory and practice by unpacking the root causes of estimation failure and suggesting actionable improvements grounded in qualitative evidence and real-world insights. It reinforces the importance of integrating technical precision with human-centered adaptability for overcoming estimation challenges in modern software development.

CONCLUSION

This qualitative research has explored the intricate and interdependent challenges that software development teams face in accurately estimating time and cost. The study identified five key thematic areas: requirement ambiguity, limitations in estimation tools, human bias and experience, agile and iterative complexity, and organizational and external factors. These themes emerged through interviews, focus groups, and document analysis, underscoring that estimation challenges are not purely technical but are embedded in human, process, and organizational contexts.

The study contributes to the growing body of knowledge in software engineering by offering a comprehensive and empirically grounded framework that connects both classical and modern estimation challenges with evolving development practices like Agile. It reinforces and extends existing theories by incorporating socio-technical dimensions of software estimation — an area that is often underrepresented in traditional cost estimation literature. Moreover, the study validates and contextualizes hybrid models such as fuzzy logic approaches and alternative life cycle models like Z-SDLC, highlighting their relevance in contemporary software environments.

Despite these contributions, the study is not without limitations. The research was confined to a qualitative approach with data primarily sourced from a specific geographic and organizational context, which may limit the transferability of findings to all global settings. The sample size, while sufficient for qualitative saturation, may not represent the full diversity of industry practices across various sectors and project types.

Future research should consider expanding the scope by incorporating mixed-method approaches that combine qualitative insights with quantitative analysis for more generalizable conclusions. Longitudinal studies could also examine how estimation practices evolve over time within organizations adapting to Agile or DevOps transformations. Additionally, future work may focus on developing and empirically validating intelligent estimation tools that integrate context-awareness, human factors, and learning algorithms to provide dynamic and adaptive cost and time estimations.

In summary, the study not only unpacks the persistent and emergent challenges in software estimation but also lays the groundwork for practical interventions and future innovation in estimation techniques, models, and tools.

REFERENCES

[1] B. Boehm, "Software Engineering Economics," IEEE Trans. Software Eng., vol. 10, no. 1, pp. 4-21, 1981.

[2] K. Moløkken and M. Jørgensen, "A Review of Software Surveys on Effort Estimation," Proc. Empirical Software Engineering, pp. 223-230, 2003.

[3] Standish Group, "Chaos Report 2020," Standish Group International, 2020.

[4] B. Boehm, "COCOMO II Model Definition Manual," University of Southern California, 2000.

[5] A. Abran and J. Desharnais, "Function Points Analysis: Concepts and Practices," Annals of Software Engineering, vol. 3, pp. 215-239, 1997.

[6] M. Jørgensen, "A Review of Studies on Expert Estimation of Software Development Effort," J. Syst. Softw., vol. 70, no. 1-2, pp. 37-60, 2004.

[7] P. Mohagheghi and R. Conradi, "Quality, Productivity and Economic Benefits of Model-Driven Engineering," Empirical Softw. Eng., vol. 10, pp. 293-309, 2007.

[8] C. Jones, Estimating Software Costs, 2nd ed., McGraw-Hill, 2007.

[9] S. Z. Iqbal and M. Idrees, "Z-SDLC Model: A New Model for Software Development Life Cycle," Int. J. Eng. Adv. Res. Technol., vol. 3, no. 2, pp. 1–8, 2017.

[10] L. Minku and X. Yao, "How to Make Best Use of Cross-Company Data for Web Effort Estimation?" Proc. Empirical Softw. Eng. Conf., pp. 446-455, 2013.

[11] S. Z. Iqbal, M. Idrees, A. B. Sana, and N. Khan, "Comparative Analysis of Common Software Cost Estimation Modelling Techniques," Math. Model. Appl., vol. 2, no. 3, pp. 33-39, 2017, doi: 10.11648/j.mma.20170203.12.

[12] S. Z. Iqbal and K. Saghar, "Improving Software Cost Estimation with Function Points Analysis Using Fuzzy Logic Method," LC Int. J. STEM, vol. 1, no. 1, pp. 10–19, 2020.

[13] J. Creswell, Qualitative Inquiry and Research Design, Sage, 2013.

[14] A. Trendowicz and J. Munch, Software Project Effort Estimation, Springer, 2009.

[15] S. McConnell, Software Estimation: Demystifying the Black Art, Microsoft Press, 2006.

[16] M. Jørgensen, "Failure of Software Projects Due to Poor Estimation Practices," Information Softw. Technol., vol. 59, pp. 132-151, 2015.

[17] B. Fitzgerald and K. Stol, "Continuous Software Engineering: A Roadmap," J. Syst. Softw., vol. 123, pp. 176-189, 2017.
[18] H. Erdogmus, "Effect of Uncertainty and Experience on Software Estimation," Empirical Softw. Eng., vol. 5, pp. 23-50, 2000.
[19] J. Keil, P. Cule, K. Lyytinen, and R. Schmidt, "A Framework for Identifying Software Project Risks," Commun. ACM, vol. 41, no. 11, pp. 76-83, 1998.
[20] K. Petersen, C. Wohlin, and D. Baca, "The Waterfall Model in Large-Scale Development," Empirical Softw. Eng., vol. 17, pp. 62-103, 2012.
[21] A. M. Davis, "The Art of Requirements Triage," Computer, vol. 36, no. 3, pp. 42-49, 2003.
[22] B. W. Boehm, "Software Cost Estimation with COCOMO II," Prentice Hall, 2000.
[23] A. J. Albrecht, "Measuring Application Development Productivity," Proc. IBM Application Development Symp., 1979, pp. 83-92.
[24] K. Beck et al., "Manifesto for Agile Software Development," Agile Alliance, 2001.
[25] M. M. Hericko, E. Z. Avdic, and K. Rozman, "Using Fuzzy Logic for Software Effort Estimation," Proc. 8th International Multi-Conference on Information Society, pp. 61-64, 2005.
[26] S. Z. Iqbal and M. Idrees, "Z-SDLC Model: A New Model for Software Development Life Cycle," Int. J. Eng. Adv. Res. Technol., vol. 3, no. 2, pp. 1–8, 2017.
[27] M. Jørgensen, "A Review of Studies on Expert Estimation of Software Development Effort," J. Syst. Softw., vol. 70, no. 1-2, pp. 37-60, 2004.
[28] S. McConnell, Software Estimation: Demystifying the Black Art, Microsoft Press, 2006.
[29] A. Trendowicz and J. Münch, Software Project Effort Estimation, Springer, 2009.
[30] S. Z. Iqbal et al., "Comparative Analysis of Common Software Cost Estimation Modelling Techniques," Math. Model. Appl., vol. 2, no. 3, pp. 33-39, 2017, doi: 10.11648/j.mma.20170203.12.
[31] S. Z. Iqbal and K. Saghar, "Improving Software Cost Estimation With Function Points Analysis Using Fuzzy Logic Method," LC Int. J. STEM, vol. 1, no. 1, pp. 10-19, 2020.
[32] K. Petersen and C. Wohlin, "The Effect of Moving From a Plan-Driven to an Incremental Software Development Approach," Empir. Softw. Eng., vol. 15, pp. 654-693, 2010.
[33] M. Jørgensen and D. Sjøberg, "An Effort Prediction Interval Approach Based on the Empirical Distribution of Previous Estimation Accuracy," Information and Software Technology, vol. 45, no. 3, pp. 123-136, 2003.
[34] M. Jørgensen, "Estimation of Software Development Work Effort: Evidence on Expert Estimation and Formal Models," Int. J. Forecast., vol. 23, no. 3, pp. 449-462, 2007.
[35] H. Erdogmus, "Impact of Agile Principles on Cost and Schedule Estimation," IEEE Softw., vol. 26, no. 6, pp. 86-90, 2009.
[36] B. Kitchenham et al., "Systematic Literature Reviews in Software Engineering," Information and Software Technology, vol. 51, no. 1, pp. 7-15, 2009.
[37] J. Keil et al., "A Framework for Identifying Software Project Risks," Commun. ACM, vol. 41, no. 11, pp. 76-83, 1998.
[39] Y. S. Lincoln and E. G. Guba, Naturalistic Inquiry, Sage Publications, 1985. [40] J. W. Creswell and C. N. Poth, Qualitative Inquiry and Research Design: Choosing Among Five Approaches, 4th ed., Sage, 2017.
[41] M. Patton, Qualitative Research and Evaluation Methods, 4th ed., Sage, 2015. [42] D. Silverman, Doing Qualitative Research, 5th ed., Sage, 2021.
[43] V. Braun and V. Clarke, "Using thematic analysis in psychology," Qualitative Research in Psychology, vol. 3, no. 2, pp. 77–101, 2006.
[44] G. Guest, K. M. MacQueen, and E. E. Namey, Applied Thematic Analysis, Sage, 2012.
[45] QSR International, "NVivo 14 Software," [Online]. Available: https://www.qsrinternational.com/
[46] M. Israel and I. Hay, Research Ethics for Social Scientists, Sage Publications, 2006.

[47] B. Birt, S. Scott, D. Cavers, K. Campbell, and F. Walter, "Member checking: A tool to enhance trustworthiness or merely a nod to validation?" Qualitative Health Research, vol. 26, no. 13, pp. 1802–1811, 2016.

[48] A. Shenton, "Strategies for ensuring trustworthiness in qualitative research projects," Education for Information, vol. 22, pp. 63–75, 2004.

[49] I. Korstjens and A. Moser, "Series: Practical guidance to qualitative research. Part 4: Trustworthiness and publishing," Eur. J. Gen. Pract., vol. 24, no. 1, pp. 120–124, 2018.

[50] J. Tracy, "Qualitative quality: Eight "big-tent" criteria for excellent qualitative research," Qualitative Inquiry, vol. 16, no. 10, pp. 837–851, 2010.

[51] N. Malterud, "Reflexivity and metapositions: Strategies for appraisal of clinical knowledge," Qualitative Health Research, vol. 1, no. 2, pp. 261–275, 2001.