JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 10. NO. 2 NOVEMBER 2024
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i2.5597

EXPLORING AGILE EFFORT ESTIMATION ISSUES: A SYSTEMATIC LITERATURE REVIEW

A Tetti Sinaga[1*]; Teguh Raharjo[2]; Ni Wayan Trisnawaty[3]

Faculty of Computer Science[1,2,3]
University of Indonesia, Jakarta, Indonesia[1,2,3]
www.ui.ac.id[1,2,3]
tetti.sinaga@ui.ac.id[1*], teguhr2000@gmail.com[2], ni.wayan05@ui.ac.id[3]

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract*—Effort estimation is crucial in software development, especially in Agile projects. The 2020 Standish Group survey found that only 31% of software projects success. The success of a software development project depends on the accuracy of effort estimation. This research aims to analyze studies related to effort estimation methods in Agile software development to identify related issues. A systematic literature review by Kitchenham was conducted across Emerald, Science Direct, Scopus, SpringerLink, and IEEE databases and identified 239 relevant studies from 2018 and 2023, ultimately focusing on 40 studies about effort estimation challenges in Agile software development. The research revealed 59 issues related to various estimation methods. The main challenge in effort estimation for Agile software development is team experience and limited knowledge about the domain, which results in inaccurate estimation result. Requirements' details, tasks complexity, and lack of data will complicate problem-solving and the prediction of the duration of completion. Reliance on expert judgment will increase the risk of bias and inaccuracy in estimates. These challenges increase the likelihood of project failure due to a mismatch between initial planning and reality as development progresses.*

Keywords*: agile, estimation effort, estimation effort issue, software development effort.*

Intisari—*Effort estimation sangat penting dalam pengembangan perangkat lunak, terutama dalam proyek Agile. Survei Standish Group tahun 2020 menemukan bahwa hanya 31% proyek perangkat lunak yang berhasil. Keberhasilan proyek pengembangan perangkat lunak bergantung pada keakuratan Effort estimation. Penelitian ini bertujuan untuk menganalisis studi yang terkait dengan metode Effort estimation dalam pengembangan perangkat lunak Agile untuk mengidentifikasi masalah dan tantangan dalam penentuan Effort estimation. Systematic literature review oleh Kitchenham dilakukan pada seluruh database Emerald, Science Direct, Scopus, SpringerLink, dan IEEE dan mengidentifikasi 239 studi relevan dari tahun 2018 dan 2023, dan mengidentifikasi 40 studi yang berfokus pada tantangan estimasi upaya dalam proyek pengembangan perangkat lunak Agile. Penelitian ini mengungkapkan 59 masalah yang terkait Effort estimation dari berbagai metode estimasi. Tantangan utama dalam penentuan Effort estimation untuk pengembangan perangkat lunak Agile diantaranya pengalaman tim dan pengetahuan terbatas tentang domain yang menghasilkan hasil estimasi yang tidak akurat. Rincian persyaratan, kompleksitas tugas, dan kurangnya data akan mempersulit pemecahan masalah dan prediksi durasi penyelesaian. Ketergantungan pada penilaian ahli akan meningkatkan risiko bias dan ketidakakuratan dalam estimasi. Tantangan-tantangan ini meningkatkan kemungkinan kegagalan proyek karena ketidaksesuaian antara perencanaan awal dan kenyataan saat pengembangan berlangsung.*

Kata Kunci*: agile, upaya estimasi, masalah upaya estimasi, upaya pengembangan perangkat lunak.*

INTRODUCTION

The Information Technology (IT) industry has become one of the leading sectors globally, contributing active hardware and software programs supporting fields such as medicine, business, education, and social networking [1]. Software development continues to evolve and has become a critical global technology. The Standish Group's 2020 report, analyzing 50,000 projects, reveals that only 31% of software projects meet their objectives, with over 50% of those failing to deliver value to companies[2].

The success and relevance of companies are determined by their ability to produce technology on a larger scale than, or equivalent to, the needs of their consumers, yet faster than their competitors. New software development models have emerged in response to changing customer needs and the increasing pressure to innovate continuously. New software development models have emerged to meet changing customer needs and the pressure for continuous innovation, leading to more complex organizational structures and increased collaboration across management, employees, and departments[3].

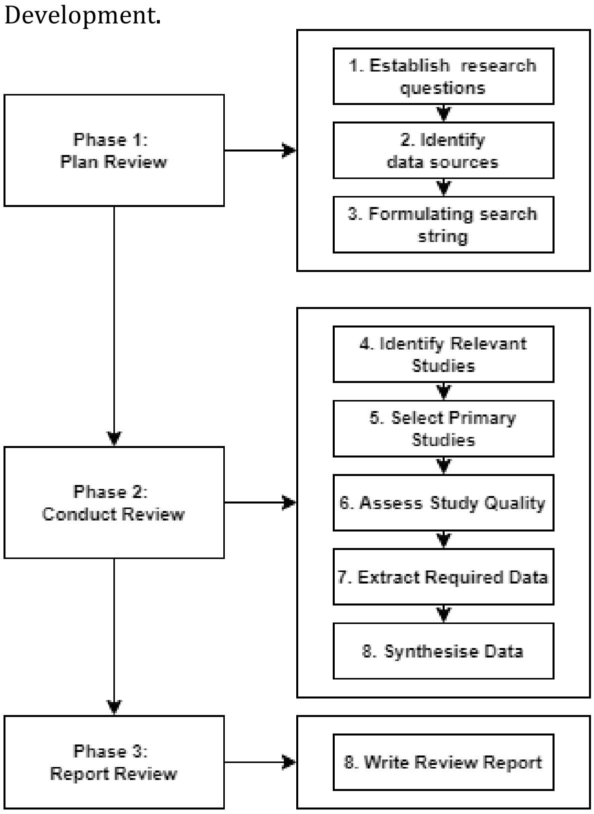
The Agile approach brings gradual changes, enables software projects to adapt flexibly to changing requirements and has gained wide acceptance in development. Since the Agile Manifesto and its 12 principles in 2001, the research community has shown significant interest in Agile methods. In recent years, the widespread adoption of Agile methodologies in software development has become evident [4]. Additionally, the application of Artificial Intelligence (AI)-based methods in Software Effort Estimation has increased [5].

However, the Standish Group report in 2020 shows that 50% of projects are challenged (meaning over budget, behind schedule, or missing features) [2]. This can occur due to errors in development planning, including determining the estimated effort for project work. Therefore, software developers require an effective effort estimation model to facilitate project planning [4]. With the evolution of software development methods and the corresponding effort estimation techniques, it becomes essential to analyze the existing methods to find issues in each method in order to maximize the accuracy of the effort estimation. This research aims to analyze studies on the methods used for estimating size or effort in Agile Software Development from 2018 to 2023 to identify the issue in effort estimation and ascertain the most prevalent issue in determining effort estimation in software development using Agile approaches. So, this study will answer the research question: What are the challenges faced in producing accurate effort estimates in Agile software development?

.

MATERIALS AND METHODS

Several research works have been conducted on predicting effort in Agile software development projects and have inspired various authors to undertake efforts to consolidate the existing knowledge on this subject. This article presents Systematic Literature Review (SLR) on effort estimation in Agile software issues. The review is based on a study conducted from 2018 to 2023, with the primary goal of analyzing studies related to the issues in estimating size or effort in Agile Software Development.



Source : (Research Results, 2024)

Figure 1. SLR process Exploring Agile Effort Estimation Issues

The SLR process is carried out in 3 phases: the initial phase, conducting and reporting the review as shown in figure 1. Each phase of the SRL process is explained as follows:

Phase 1: Planning Review In this initial phase, the research questions are established as the foundation for conducting the SLR to achieve the

VOL. 10. NO. 2 NOVEMBER 2024
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i2.5597

research objectives. The predetermined research question focuses on the conditions of effort estimation issues in software development using an Agile approach. The researchers then identify data sources for the SLR selecting Emerald, Science Direct, Scopus, SpringerLink, and IEEE for this study. Next, a search string is formulated to filter studies relevant to the research question, and criteria for inclusion and exclusion are determined, along with quality criteria.

The guidelines provided by [6] for effective literature search suggest breaking down the research question into its essential components, such as the subject matter being studied, the intervention being implemented, the effects being measured, and the research design being employed. Subsequently, a comprehensive list of synonyms, acronyms, and alternative spellings for these keywords should be compiled. With a comprehensive set of keywords, practical search strings can be constructed using Boolean AND and OR operators.

Thus, the research string used includes (Agile AND software AND development AND "effort estimation") with an inclusion range from 2018 to 2023, focusing on the subject area of computer science, restricting to articles, and using the English language. This initial phase yields 43 documents from Scopus, 47 from IEEE, 25 from Science Direct, 41 from SpringerLink, and 57 from ProQuest. The review identified 239 studies on effort estimation in software development with an Agile approach show in Table 1.

Phase 2: Conducting the Review Following the planning review, the researchers proceed to the second phase by selecting and extracting relevant studies based on the research question and title. Document extraction and synthesis are performed to identify the documents most aligned with the research question. After completing this phase, 40 documents relevant to effort estimation are identified show in Table 1.

Phase 3: Reporting the Review With the extraction and synthesis of previous research documents completed, the third phase analyzes effort estimation methodologies and issues employed by previous researchers. In the reporting phase, the researchers document the extraction process results, focusing on issues in effort estimation in software development with an Agile approach. The analysis aims to conclude the issues in effort estimation up to 2023. The total number of research papers/articles found, collected, and selected after a rigorous search is presented in the Table 1.

Table 1. Total research papers/articles found collected and selected after a rigorous search

	Emerald	Science Direct	Scopus	Springer Link	IEEE	ProQuest
Collected	26	25	43	41	47	57
Selected	5	2	8	5	14	6

Source: (Research Results, 2024)

RESULTS AND DISCUSSION

The data was analyzed to identify issues in Agile effort estimation. Then, a deeper analysis of the 40 selected studies was conducted to find the issues and challenges that arise when determining effort estimation. 59 issues and challenges were identified during the systematic literature review, as presented in Table 2. Tables 3 and 4 present the frequency of their appearance in previous studies.

Table 2. Estimation Effort Issue

Estimation Effort Issue	Code
Dynamic nature of software development requirements	DY
Requirements details	UN
The effort calculated may differ from the actual case	TH
Ignorance of requirements	IG
Uncertainties in requirements	CH
Non-functional requirements	NO
functional requirement	ZE
Task priority	SK
Incomprehensive and Unrealistic of the tasks	TH
Unclear Task information	UT
Underestimation of task	NU
Overestimation of task	OV
Determination/ calculation of transaction complexities of the task	PC
Project size/scope	PR
Story point factors	ST
Difficult attribute relation analysis	DI
Hardware/software requirements	HA
Organizational factor	OR
National cultures factor	NA
Time-consuming	TI
Requires a work breakdown structure	RE
Maintain efficiency	MA
Reliable as possible	RE
Biased towards certain criteria	BI
Accuracy of Effort Estimation Processes	AC
Estimation validation	VA
Project type	TY
Complexity of tasks in software production	CP
Absence of expert	AB
Relies on the subjective and judgment evaluation of experts	RI
Individual estimates	EX
Availability of historical data	UA
Dataset can be problematic / Quality data	DA
Ignorance of documentation	IO
Insufficient data	IS
Dependency on documented information	DP
Dataset characteristics	CS
The maturity level of teams	LE

Estimation Effort Issue	Code
Limited knowledge of the domain	LI
Limited knowledge of the risks	LM
Team member's experience	CL
Reliance on Subjective Assessment	RL
Unifying Team Perspectives	UI
Incorrect assumptions about the work involved	IC
Communication skills	CM
Managerial skills	MN
Technical ability	BT
Working environment	WO
Inexperienced Scrum masters	EX
The gap between reality and developers' perceptions	GA

Estimation Effort Issue	Code
dThe dynamic and complex nature of modern software development	DN
Techniques had not been empirically validated	TE
Lack of Techniques to Detect Missing or Changing Information	LA
Reliance on Subjective Methods	RA
The method might be complex a	ME
Reliability of external software deliveries	RY
Incorrect algorithm parameter	CY
The accuracy of models	CH
Accuracy for bug fixing and refactoring	BU

Source: (Research Results, 2024)

Table 3. Results of the analysis of 40 previous studies

Author	DY	UN	TH	CH	IG	NO	ZE	SK	TH	UT	NU	OV	PC	PR	ST	DI	HA	OR	NA	TI	RE	MA	RE	BI	AC	VA	ES	TY	CP	FU	UR	AB	RI
Hansen [1]	✓																																
Kumar [3]								✓							✓									✓	✓				✓				✓
Tawosi [4]	✓																																
Meiliana [7]											✓	✓																				✓	
Effendi [8]			✓										✓																				
Usman [9]														✓																		✓	✓
Sharma [10]		✓									✓	✓		✓		✓																	✓
Sielskaitė [11]				✓	✓												✓	✓	✓	✓													
Rahman [12]																								✓									
Abdullah [13]																				✓													
Mateen [14]													✓							✓	✓												
Choetkiertikul [15]																																	✓
Arachchi [16]															✓																		✓
Unlu [17]													✓																				
Sanchez [18]																																✓	
Madya [19]									✓											✓		✓	✓										
Tashtoush [20]													✓							✓													
Kaur [21]		✓		✓						✓																							
Linz [22]		✓											✓																			✓	✓
Iftint [23]		✓								✓																							
Ahmed [24]				✓																												✓	
Ritu [25]																								✓									

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 2 NOVEMBER 2024
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i2.5597

Author	DY	UN	TH	CH	IG	NO	ZE	SK	TH	UT	NU	OV	PC	PR	ST	DI	HA	OR	NA	TI	RE	MA	RE	BI	AC	VA	ES	TY	CP	FU	UR	AB	RI
Predescu [26]															✓								✓		✓	✓			✓				
Cao [27]											✓														✓				✓				✓
Borade [28]		✓															✓			✓					✓			✓	✓				✓
Hamid [29]														✓															✓				
Sandeep [30]											✓	✓		✓											✓								
Karna [31]		✓												✓															✓				
Butt [32]												✓																					✓
Sousa [33]		✓		✓																													
Turic [34]						✓																											
Bhaskaran [35]		✓					✓							✓	✓														✓				
Sudarmaningtyas [36]	✓	✓															✓											✓	✓				
Stober [37]							✓							✓											✓								
Khuat [38]														✓	✓														✓				

Source: (Research Results, 2024)

Table 4. Results of the analysis of 40 previous studies (continue)

Author	EX	UA	DA	IO	IS	DP	CS	LE	LI	LM	CL	RL	UI	CI	IC	CM	MN	BT	WO	EX	GA	DN	TE	LA	RA	ME	RY	TO	CY	CH	BU
Hansen [1]		✓			✓					✓	✓		✓									✓					✓				
Kumar [3]								✓																							
Tawosi [4]																														✓	
Pasuksmit [5]			✓	✓	✓	✓				✓														✓							
Usman [9]											✓					✓														✓	
Sharma [10]																					✓										
Sielskaitė [11]				✓					✓		✓											✓									
Rahman [12]			✓	✓		✓			✓																						
Abdullah [13]									✓		✓												✓								
Mateen [14]					✓				✓																						
Choetkiertikul [15]									✓			✓																			
Unlu [17]																									✓						
Sanchez [18]			✓			✓																✓			✓					✓	
Madya [19]																														✓	
Tashtoush [20]									✓				✓	✓		✓						✓				✓					
Kaur [21]				✓																											
Linz [22]					✓										✓																
Iftint [23]					✓																										

Author	EX	UA	DA	IO	IS	DP	CS	LE	LI	LM	CL	RL	UI	CI	IC	CM	MN	BT	WO	EX	GA	DN	TE	LA	RA	ME	RY	TO	CY	CH	BU
Ahmed [24]		✓																													
Predescu [26]					✓	✓																									
Cao [27]			✓						✓																						
Borade [28]			✓								✓					✓	✓	✓	✓												
Hamid [29]	✓					✓					✓						✓		✓	✓											
Karna [31]						✓																							✓	✓	
Butt [32]	✓																				✓										
Sousa [33]					✓		✓															✓				✓					
Turic [34]																															✓
Bhaskaran [35]											✓							✓				✓								✓	
Sudarmaningtyas [36]									✓		✓					✓	✓	✓													✓
Stober [37]								✓			✓																				
Khuat [38]								✓			✓																			✓	
Usman [39]		✓						✓																							✓
Priya Varshini [40]			✓		✓																										

Source: (Research Results, 2024)

The more frequently an issue appears in earlier research, the more seriously it is considered. The most frequently occurring problem in previous studies was Team Member's Experience and Complexity of Tasks in Software Production, which appeared in 10 prior studies. This was followed by Requirements Details and Relies on Expert's Judgment, which appeared in 9 previous studies, and Limited Knowledge About the Domain, which appeared in 8 previous studies. More details can be seen in Tables 3 and 4 and visualized in Figure 1.
The research found that the most common issue in Agile effort estimation is "team member experience," followed by "limited domain knowledge," "requirements details," "task complexity," and "reliance on expert judgment." Other common issues include "insufficient data," "project size/scope," "model accuracy," and "time-consuming processes."

Various estimation methods are classified into four main categories: formal algorithmic estimation, expert estimation, analog estimation, and combination-based estimation. For instance, analog estimation requires as much data as possible about previously implemented projects to make accurate comparisons. Estimating software development efforts becomes increasingly challenging due to the rising complexity of software projects and more total projects. Specialists from various fields must collaborate to obtain realistic estimations, but there are still inherent risks when planning based on these estimates, which can lead to cost overruns.

Sudarmaningtyas [36] explains that various attributes are involved in the effort estimation process in Agile methodology. These attributes include complexity, story points, experience, size, user stories, effort, and time, with complexity, experience, size, effort, and time being the most frequently used attributes in the last three years. Effort estimation in software development is a critical process that determines the amount of work required to develop a system and plays a significant role in pricing and project planning.

Many studies have examined expert assessment, data-driven, and AI-supported estimation methods, but none can yet accurately measure effort in Agile software development. The following section analyzes the top 5 challenges (ranked 1-5) in Agile effort estimation, as shown in Figure 2.

A. Team Member's Experience

In software development using the Agile approach, especially in the Scrum methodology, the team's experience plays a crucial role in effort estimation. The development team typically does estimation in Agile development. New Agile teams

VOL. 10. NO. 2 NOVEMBER 2024
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i2.5597

often struggle to provide accurate story point estimates. This difficulty primarily arises due to the team's lack of experience or insufficient training in Agile methodology. Additionally, it has been observed that only a few team members actively participate in the estimation process.

In contrast, others act as observers and agree with the provided estimates without engaging in thorough discussions. The condition can result in a backlog or an increase in development time. Sudarmaningtyas [36] agreed in their research that team members' experiences influence effort estimation in Agile software development. This study found that factors related to individuals, such as team members' experience, knowledge, and technical skills, significantly impact effort estimation in Agile development. Together, these three factors contribute significantly to the estimation of effort by 31.1%. Regarding the magnitude of their effects, technical ability has the most significant influence, followed by knowledge and experience.

Usman [9] identifies potential factors that can influence the accuracy of project estimates, which involves considering project size, requirements prioritization, and the development team's maturity level. The maturity level of the development team has a significant impact on project estimation. The maturity of the development team plays a crucial role in improving the accuracy of effort estimation in large-scale Agile software development.

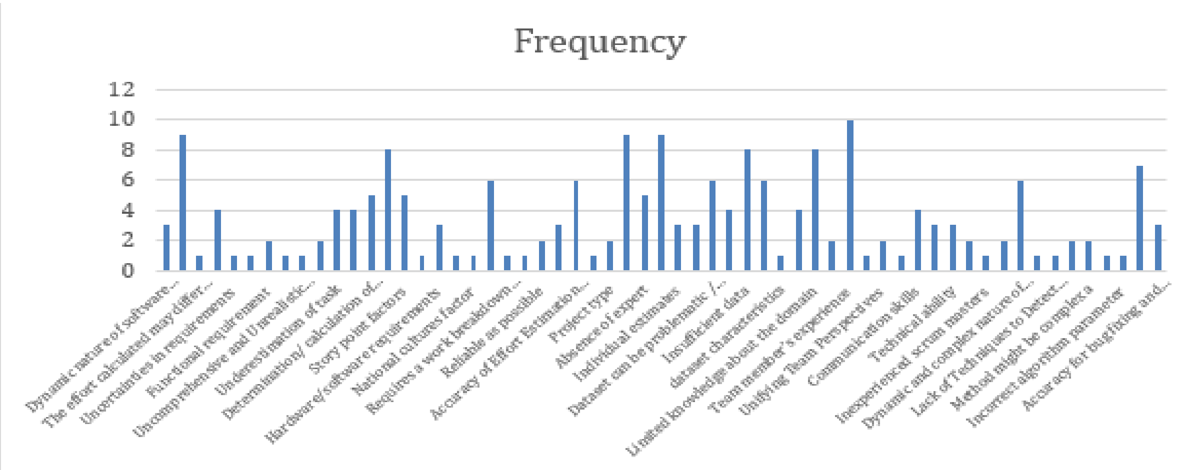
B. Requirements Details

Effort estimation in Agile involves factors related to the project and factors related to people. Project factors include quality, hardware and software requirements, ease of operation, complexity, data transactions, and multiple locations. In Agile methodologies like Scrum, effort estimation is typically done using 'story points' that express the size and features associated with a user story. These story points are the basis for upcoming releases and influence the project direction based on business value and initial developer estimates.

Choetkiertikul [15] acknowledges that story points estimated by human teams can contain biases and may not always be accurate. To address this, the researchers conducted experiments with both original and adjusted normalized story points. Karna [31], utilizing data mining techniques to determine effort estimation in Agile software development, found that the detailed requirements in Agile software development can be predicted more accurately using data mining techniques. In turn, it can assist in optimizing project management and reducing errors in effort estimation.

C. Limited Knowledge About the Domain

Varshini [40] explores that a lack of domain knowledge, uncertainties in project deadlines, and reliance on traditional effort estimation methods often lead to unreliable estimates and Abdullah [13] highlights challenges faced in effort estimation due to the involvement of unspecialized team members in tasks they are not skilled at. One significant challenge is the "insufficient understanding of influencing factors and potential risks," which can result in inaccurate estimations. These issues can give rise to substantial problems, such as software products not being delivered on time or failing to meet expected non-functional requirements.



Source: (Research Results, 2024)

Figure 2. Frequency/occurrences of effort estimation issues

D. Complexity of Task in Software Production

Task complexity in Agile software production becomes problematic due to the inherent uncertainty and variability in software development tasks. Agile methods emphasize adaptability and iterative development, which can lead to changes in task requirements and scope. This complexity becomes a significant issue, primarily due to the subjective nature of the traditional estimation approach, which can be time-consuming, especially with numerous user stories and various influencing factors such as the development team's expertise, task size, and the working environment. Khuat [38] uses a hybrid method in effort estimation while Ünlü [17] explores effort estimation within the context of microservice architectures, expressing that the challenges of accurately estimating project effort due to the unpredictable nature of software requirements and the dynamic nature of Agile teams, and communications contribute to the uncertainty in expert judgment-based estimations, necessitating more objective and systematic approaches for accurate effort prediction.

E. Relies on Expert's Judgment

Subjective estimation methods like expert judgment and planning poker are commonly used, which can introduce biases and inconsistencies. The expert judgment-based effort estimation approach is one of the methods used in the software industry to predict how much human resources are required to complete a software project. This approach involves experts with experience in similar projects providing estimates of how long and how many workforce resources are needed to complete the project. The complexity of tasks and the inherent uncertainty in software projects make it challenging to estimate accurately using expert judgment alone, where team skills, prior experience, and task size as fundamental cost drivers can lead to variability in estimates due to differences in individual expert assessments, variations in expert experience and expertise, and the potential for cognitive biases. Usman [39] explores the complexities of relying on expert judgment for effort estimation in software development and identifies that reliance on expert judgment, such as planning poker, analogy, and expert judgment, can lead to subjective and inconsistent estimations.

CONCLUSION

The 40 studies on Agile software development effort estimation, identified through SLR, explored various methods like data-driven estimation, AI-supported systems, formal algorithmic estimation, expert estimation, analog estimation, and combination-based estimation. A total of 59 issues were identified across all approaches, with the most common challenge are team experience and limited knowledge about the domain, which results in inaccurate estimation result. Requirements' details, tasks complexity, and lack of data will complicate problem-solving and the prediction of the duration of completion. Reliance on expert judgment will increase the risk of bias and inaccuracy in estimates.

These challenges increase the risk of project failure due to mismatches between initial planning and reality as development progresses. Key issues must be addressed to reduce project failure risks. Practical solutions include enhancing team knowledge through training and mentoring, improving requirements quality via regular refinement sessions and visualization tools, and reducing bias in expert judgment with structured techniques like the Delphi method. Combining data-driven and expert-based estimates can also improve accuracy. For future research, exploring how AI and machine learning can enhance Agile estimation is recommended, focusing on intelligent systems that learn from project data. Additionally, studies should examine how team cohesion, collaboration, and communication impact estimation accuracy.

REFERENCE

[1] P. Hansen and H. Timinger, "Concept of a fuzzy expert system for story point estimations in agile projects," *in Proc. IEEE Int. Conf. Eng., Technol. Innov. (ICE/ITMC) & Int. Assoc. Manag. Technol. (IAMOT) Joint Conf.*, pp. 1–9, 2022, doi: 10.1109/ITMC-IAMOT55089.2022.10033271.
[2] Standish Group International, CHAOS Summary 2020. The Standish Group International, Inc., 2020.
[3] S. Kumar, M. Arora, Sakshi, and S. Chopra, "A review of effort estimation in agile software development using machine learning techniques," *in Proc. 4th Int. Conf. Inventive Res. Comput. Appl. (ICIRCA),* pp. 416–422, 2022, doi: 10.1109/ICIRCA54612.2022.9985542.
[4] V. Tawosi, A. Al-Subaihin, and F. Sarro, "Investigating the effectiveness of clustering for story point estimation," *in Proc. IEEE Int. Conf. Softw. Anal., Evol., Reengineering (SANER),* pp. 827–838, Mar, 2022, doi: 10.1109/SANER53432.2022.00101.
[5] J. Pasuksmit, P. Thongtanunam, and S. Karunasekera, "Towards reliable agile iterative planning via predicting documentation changes of work items," *in*

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 2 NOVEMBER 2024
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i2.5597

Proc. Mining Softw. Repositories Conf. (MSR), pp. 35–47, May, 2022, doi: 10.1145/3524842.3528445.
[6] B. Kitchenham et al., "Systematic literature reviews in software engineering – A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2019, doi: 10.1016/j.infsof.2008.09.009.
[7] Meiliana, Daniella, G. Wijaya, N. Putra, N. G. E., and R. Efata, "Agile software development effort estimation based on product backlog items," *Procedia Comput. Sci.*, vol. 227, pp. 186–193, 2023, doi: 10.1016/j.procs.2023.10.516.
[8] A. Effendi, R. Setiawan, and Z. E. Rasjid, "Adjustment factor for use case point software effort estimation (Study case: Student desk portal)," *Procedia Comput. Sci.,* vol. 157, pp. 691–698, 2019, doi: 10.1016/j.procs.2019.08.215.
[9] M. Usman, K. Petersen, J. Börstler, and P. Santos Neto, "Developing and using checklists to improve software effort estimation: A multi-case study," *J. Syst. Softw.*, vol. 146, pp. 286–309, 2018, doi: 10.1016/j.jss.2018.09.054.
[10] A. Sharma and N. Chaudhary, "Prediction of software effort by using non-linear power regression for heterogeneous projects based on use case points and lines of code," *Procedia Comput. Sci.,* vol. 218, pp. 1601–1611, 2022.
[11] G. Sielskaitė, "Analyzing software effort estimation by applying static, single & multivariable models," *in Proc. Int. Conf. Comput. Methodol. Commun. (ICCMC),* pp. 832–835, 2021, doi: 10.1109/ICCMC51019.2021.9418286.
[12] M. Rahman et al., "Software effort estimation in agile development using planning poker and Fibonacci scale: A comparative study," *in Proc. Int. Conf. Emerg. Technol. Comput. (iCETiC),* pp. 1–6, 2020, doi: 10.1109/iCETiC50022.2020.9384907.
[13] A. Abdullah, A. Hussein, and G. Andy, "A case study validation of the pair-estimation technique in effort estimation of mobile app development using agile processes," *in Proc. 10th Int. Conf. Adv. Comput. Inf. Technol. (ACIT)*, 2020, doi: 10.1109/ACIT2020.2020.9205983
[14] M. A. Mateen and A. A. Malik, "A comparative study of the accuracy and efficiency of Wideband Delphi and Planning Poker software effort estimation techniques," *Proc. Int. Conf. IT Ind. Technol. (ICIT),* pp. 1–5, 2023, doi: 10.1109/ICIT59216.2023.10335782.
[15] M. Choetkiertikul et al., "A deep learning model for estimating story points," *IEEE Trans. Softw. Eng.*, vol. 45, no. 7, pp. 637–656, 2019, doi: 10.1109/TSE.2018.2792473.
[16] K. J. W. Arachchi and C. R. J. Amalraj, "An agile project management supporting approach for estimating story points in user stories," *Proc. Int. Conf. Inf. Technol. Res. (ICITR),* pp. 1–6, 2023, doi: 10.1109/ICITR61062.2023.10382930.
[17] H. Unlu et al., "An exploratory case study on effort estimation in microservices," *Proc. 49th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, pp. 215–218, 2023, doi: 10.1109/SEAA60479.2023.00040.
[18] R. Sanchez and J. Carroll, Usability Engineering: Scenario-Based Development of Human-Computer Interaction, 2001. DOI: 10.1016/j.jksuci.2019.03.001.
[19] G. R. Madya, E. K. Budiardjo, and K. Mahatma, "PREP: A post-requirements effort estimation method in Scrum's sprint grooming," *Proc. Int. Conf. Data Softw. Eng. (ICoDSE),* pp. 132–137, 2022, doi: 10.1109/ICoDSE56892.2022.9972012.
[20] Y. M. Tashtoush et al., "Project management effort estimation using Agile Manager game platform," *Proc. Int. Conf. Inf. Commun. Syst. (ICICS)*, pp. 149–154, 2022, doi: 10.1109/ICICS55353.2022.9811211.
[21] A. Kaur and K. Kaur, "Function points based test effort estimation model for mobile applications," *J. King Saud Univ. - Comput. Inf. Sci.,* vol. 34, no. 3, pp. 946–963, 2022.
[22] T. Linz, Testing in Scrum: A Guide for Software Quality Assurance in the Agile World. 2014.
[23] D. Iftint, R. Catalin, and O. Oliver, "An NLP approach to estimating effort in a work environment," *Proc. Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM*), pp. 1–6, 2020, doi: 10.23919/SoftCOM50211.2020.9238219
[24] M. Ahmed et al., "Blockchain-based software effort estimation: An empirical study," *IEEE Access,* vol. 10, pp. 120412–120425, 2022, doi: 10.1109/ACCESS.2022.3216840.
[25] R. Tiwari and S. P. Vivekanandan, "Neural network-based agile software effort estimation in support vector machine classification model (ANN-ASVM)," *Neurocomputing,* vol. 437, pp. 171–182, 2021, doi: 10.1016/j.neucom.2021.01.088.
[26] E. Predescu, A. Stefan, and A. V. Zaharia, "Software effort estimation using multilayer perceptron and long short-term memory," *Informatica Economica*, vol. 23, no. 2, pp. 76–87, 2019, doi: 10.12948/issn14531305/23.2.2019.07.

[27] L. Cao, "Estimating efforts for various activities in agile software development: An empirical study," *IEEE Access*, vol. 10, pp. 83311–83321, 2022, doi: 10.1109/ACCESS.2022.3196923.

[28] J. Borade and V. R. Khalkar, "Software project effort and cost estimation techniques," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 8, 2013.

[29] M. Hamid, F. Zeshan, and A. Ahmad, "Fuzzy logic-based expert system for effort estimation in Scrum projects," *Int. J. Inf. Eng.*, vol. 7, no. 3, 2021.

[30] R. C. Sandeep, M. Sánchez-Gordón, R. Colomo-Palacios, and M. Kristiansen, "Effort estimation in agile software development: An exploratory study of practitioners' perspectives," *Lean and Agile Software Development,* vol. 438, pp. 136–149, 2022, doi: 10.1007/978-3-030-94238-0_8.

[31] H. Karna, S. Gotovac, and L. Vicković, "Data mining approach to effort modeling on agile software projects," *Informatica*, vol. 44, 2020, doi: 10.31449/inf.v44i2.2759.

[32] S. A. Butt et al., "A cost estimating method for agile software development," *Computational Science and Its Applications-ICCSA*, vol. 12955, pp. 231–245, 2021, doi: 10.1007/978-3-030-87007-2_17.

[33] A. O. Sousa et al., "Applying machine learning to estimate the effort and duration of individual tasks in software projects," *IEEE Access,* vol. 11, pp. 89933–89946, 2023, doi: 10.1109/ACCESS.2023.3307310

[34] M. Turic, S. Celar, S. Dragicevic, and L. Vickovic, "Advanced Bayesian network for task effort estimation in agile software development," *Appl. Sci.,* vol. 13, no. 16, p. 9465, 2023, doi: 10.3390/app13169465.

[35] N. A. Bhaskaran and V. Jayaraj, "A hybrid effort estimation technique for agile software development (HEETAD)," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 1, pp. 1078–1087, 2019, doi: 10.35940/ijeat.A9480.109119.

[36] P. Sudarmaningtyas and R. Mohamed, "Significant factors in agile software development of effort estimation," *Pertanika J. Sci. Technol.*, vol. 30, no. 4, pp. 2851–2878, 2022, doi: 10.47836/pjst.30.4.30.

[37] I. T. Stober, "Agile software development," *Conf. Decision Aid Sci. Appl. (DASA),* pp. 761–765, 2007, doi: 10.1109/DASA53625.2021.9682239.

[38] T. T. Khuat and M. H. Le, "A novel hybrid ABC-PSO algorithm for effort estimation of software projects using agile methodologies," *J. Intell. Syst.*, vol. 27, no. 3, pp. 489–506, 2018, doi: 10.1515/jisys-2016-0294.

[39] M. Usman, J. Börstler, and K. Petersen, "An effort estimation taxonomy for agile software development," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 27, no. 4, pp. 641–674, 2017, doi: 10.1142/S0218194017500243

[40] A. G. Priya Varshini et al., "Machine learning approach for software effort estimation using combination of principal component regression and neural network," *J. Phys.: Conf. Ser.,* vol. 2325, no. 1, 2022, doi: 10.1088/1742-6596/2325/1/012049

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri