

<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *2025 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2025, Honolulu, Oct 2-3, 2025*.

Citation for the original published paper:

Edison, H., Ali, N b. (2025)

Another Systematic Review? A Critical Analysis of Systematic Literature Reviews on Agile Effort and Cost Estimation

In: *International Symposium on Empirical Software Engineering and Measurement* (pp. 23-32). IEEE Computer Society

International Symposium on Empirical Software Engineering and Measurement

<https://doi.org/10.1109/ESEM64174.2025.00051>

N.B. When citing this work, cite the original published paper.

©2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:bth-29289>

Another Systematic Review? A Critical Analysis of Systematic Literature Reviews on Agile Effort and Cost Estimation

Henry Edison and Nauman bin Ali
Department of Software Engineering
Blekinge Institute of Technology, Sweden
 {henry.edison, nauman.ali}@bth.se

Abstract

Background: Systematic literature reviews (SLRs) have become prevalent in software engineering research. Several researchers may conduct SLRs on similar topics without a prospective register for SLR protocols. However, even ignoring these unavoidable duplications of effort in the simultaneous conduct of SLRs, the proliferation of overlapping and often repetitive SLRs indicates that researchers are not extensively checking for existing SLRs on a topic. Given how effort-intensive it is to design, conduct, and report an SLR, the situation is less than ideal for software engineering research. **Aim:** To understand how authors justify additional SLRs on a topic. **Method:** To illustrate the issue and develop suggestions for improvement to address this issue, we have intentionally picked a sufficiently narrow but well-researched topic, i.e., effort estimation in Agile software development. We identify common justification patterns through a qualitative content analysis of 18 published SLRs. We further consider the citation data, publication years, publication venues, and the quality of the SLRs when interpreting the results. **Results:** The common justification patterns include authors claiming gaps in coverage, methodological limitations in prior studies, temporal obsolescence of previous SLRs, or rapid technological and methodological advancements necessitating updated syntheses. **Conclusion:** Our in-depth analysis of SLRs on a fairly narrow topic provides insights into SLRs in software engineering in general. By emphasizing the need for identifying existing SLRs and for justifying the undertaking of further SLRs, both in design and review guidelines and as a policy of conferences and journals, we can reduce the likelihood of duplication of effort and increase the rate of progress in the field.

Index Terms

systematic reviews, need for a review, agile, effort estimation, cost estimation

I. INTRODUCTION

Systematic literature reviews (SLRs) were adopted from medicine into software engineering as a tool to establish evidence-based software engineering [1]. The purpose of SLRs is to identify, assess, and synthesize evidence related to narrow and specific research questions reported in empirical studies [2]. Since its introduction in the software engineering domain in 2004 [3], the number of SLRs published in top-tier conferences and journals has increased rapidly [4], [5]. Several studies have evaluated the methodological rigour and relevance of published SLRs in the field. Such studies have resulted in revised and improved SLR guidelines for designing and conducting [1], [6], reporting [4], [7] and evaluating [8]–[10].

However, when it comes to the quality assessment of SLRs in software engineering, mainly the DARE criteria¹ are used. Costal et al. [11] found that of the 35 studies they evaluated, only one study used an adaptation of the Assessing the Methodological Quality of Systematic (AMSTAR) framework, and four proposed their own assessment criteria. The overwhelming majority (that is, 30 of the 35 studies) used DARE or its variations. DARE criteria and its adaptations in previous studies have very rightly focused on assessing the methodological rigor and the relevance of SLRs. However, DARE has not questioned whether these SLRs were warranted in the first place or not.

Like all research, SLRs must be carried out diligently, following rigorous methodological guidelines, with results accurately reported and interpreted. SLRs typically require much greater effort than conventional literature reviews. Therefore, researchers in software engineering should carefully consider whether undertaking a systematic literature review is warranted before proceeding. One may assume that it is a given and would be handled by the authors or during the peer review process. We agree with this assumption but would like to empirically assess if that is the

¹DARE criteria used by NIHR Centre for Reviews and Dissemination

case in practice. Some of this unintended redundancy may be unavoidable, due to various reasons, lack of visibility for certain venues, simultaneous efforts undertaking literature reviews with similar or overlapping aims, but we want to ascertain the extent of the overlap and duplication of effort and suggest guidelines to avoid it in the future.

For this purpose, we have consciously chosen SLRs in the area of effort estimation in Agile, as an illustrative example case. Effort estimation has always been a topic of interest for secondary studies. For example, in one of the earliest tertiary studies identifying early adopters of SLRs in the software engineering community, Kitchenham et al. [3] identified that among the first 20 SLRs, seven were about cost estimation. Since the early 2000s, Agile has had a major impact on software development and on effort estimation. Given the extensive historical emphasis on this area and the continued interest, we decided to use this as a case in this study.

In this study, we pose and answer the following research questions:

- RQ1: Which systematic literature reviews are published on cost/effort estimation in the agile software development context?
- RQ2: What are the characteristics of these SLRs? We focus on salient features of these studies, including aims and objectives, years covered in search, quality scores, e.g., using DARE, publication venue ranking, etc.
- RQ3: Are the SLRs aware of previous reviews on the topic? We will use citations or discussions of existing reviews to indicate awareness.
- RQ4: What justifications are presented in the SLRs to justify undertaking another SLR on the topic?

The remainder of the paper is structured as follows. Section II describes the data collection process. Then, in Section III, we report the results and analysis, which are discussed in Section IV and concluded in Section V.

II. RESEARCH METHODOLOGY

We followed Kitchenham et al.'s [1] guidelines to build the sample SLRs and analyze them to answer the research questions. The procedures we followed are detailed in this section.

A. Search strategy

We used the following search string to identify related SLRs in four databases (see Table I):

("systematic review" OR "research review" OR "research synthesis" OR "research integration" OR "systematic overview" OR "systematic research synthesis" OR "integrative research review" OR "integrative review" OR "systematic literature review" OR "literature review") AND (estimat OR predict* OR forecast* OR calculat* OR assessment OR measur* OR sizing) AND (effort OR resource OR cost OR size OR metric OR user story OR velocity) AND (Agile OR "extreme programming" OR "Scrum" OR "feature driven development" OR "dynamic systems development method" OR "crystal software development" OR "crystal methodology" OR "adaptive software development" OR "lean software development")*

One could even argue that a more inclusive approach would have been not to have the restriction of the third set of keywords capturing the Agile context. Previous reviews on effort and cost estimation may have included literature from this context without making it the central theme, and thus, they may not have mentioned it in the title, abstract, or keywords of a paper. However, after skimming through the titles of the hits, we could see that the search has already been given a sizeable sample of relevant secondary studies. Thus, we decided to keep the rather narrow focus of the search string.

To complement the search, we also searched on Google Scholar using a simple search string: *("effort estimation" OR "cost estimation") AND agile AND ("systematic literature review" OR "systematic review")*. We reviewed the first 50 hits ranked by Google Scholar on their relevance to the search string and identified nine additional SLRs. We believe this additional step adds to the confidence that we have good coverage of SLRs published on cost/effort estimation in agile software development.

We ran the search string on March 11, 2025, and retrieved a total of 162 papers, as shown in Table I.

B. Selection criteria

For a paper to be included, it must meet all of the following:

- It reviews the literature on cost or effort estimation in the context of agile software development.
- It reports a systematic literature review study. We rely here on the author's description of the study.
- Peer-reviewed papers only.
- Written in English only.

TABLE I
SEARCH RESULTS

Database	Retrieved	Duplicates	Remaining
Scopus	23	10	13
IEEE Xplore	21	1	20
ACM Digital Library	14		14
ISI Web of Science	104	7	97
Total	162	18	144

After removing 18 duplicates, 144 papers were evaluated in two stages. In the first stage, the first author reviewed the papers based on the title and abstract using the selection criteria. The papers marked as excluded were then reviewed by the second author. When there was a disagreement between the reviewers, the papers were included for a second-stage review. In the second stage, we reviewed a paper for relevance by reading its full text. Data extraction was performed if the papers met the selection criteria after reading the full text. Otherwise, the papers are excluded from further analysis. Ultimately, we included 18 papers in the analysis.

C. Data extraction

To answer our research questions, we focused on the following facets:

- The research questions of the study. Based on the formulation of the research questions, we extracted the main idea of the question. For example, “*What methods have been used to estimate size or effort in ASD*” (RQ1, SLR9) and “*What are the different approaches/techniques used for software cost estimation?*” (RQ1, SLR 10) were coded into types of estimation techniques. “*Which factors most influence agile effort estimation?*” (RQ3, SLR3) and “*What are the barriers and difficulties in the application of software cost estimation models?*” (RQ2, SLR10) were coded into challenges of estimation.
- The justification for conducting the SLR: e.g., claims about lack of coverage, quality (methodological limitations), lack of existing reviews, rapid technological advancements, methodological updates warranting an additional review, different focus than existing reviews, or temporal update.
- Quality assessment of the SLRs, based on the five DARE criteria [12]. The assessment concerns the systematic review process and whether these activities have been performed. Thus, we adopted the three-point scale: yes (1), partly (0.5), and no (0).
- The awareness of existing SLRs in the topic. We checked if the study cited previous SLRs as part of their related work or in the discussion section. If the SLRs were part of the primary studies, we would not have considered their awareness.

The unit of analysis in our study is the SLR study itself. For an SLR to be included in the analysis, the authors should perform the actual review process. Thus, studies that performed analysis on the primary studies identified in other SLRs will be excluded, i.e., [13]). Moreover, to avoid double-counting [14], if one SLR study was reported in more than one paper, we included the latest publication only. The final version of the dataset is accessible through a publicly available [15].

III. RESULTS AND ANALYSIS

A. RQ1: Existing SLRs on cost/effort estimation in the agile software development context

Based on our search and selection criteria, we identified 18 SLRs on cost/effort estimation in the agile context that were published between 2014 and 2024. Fig. 1 shows the distribution of these SLRs over time. Each square represents an SLR. Thus, for a calendar year where two SLRS were published, it will have two squares above the year. The earliest SLR on agile estimation was published in a conference in 2014 (SLR1), and since then, at least one SLR on agile estimation has been published every year except 2015. The peak was reached in 2024 when four SLRS in the area were published.

In Fig. 1, another dimension is also depicted, which is the number of primary studies included in an SLR. Thus, the placement of a square on the y-axis represents the number of primary studies in that SLR. Only three of the 18 SLRs have more than 57 primary studies, which is an indicator that a research topic has accumulated enough studies that warrant an SLR [5]. A majority, i.e., eleven SLRs have between 20 and 40 primary studies. Surprisingly, four papers have an unusually small number of primary studies for an SLR, i.e., 8, 8, 11 and 12. Such a low number of

primary studies raises questions about the scope of the SLRS and whether the need and feasibility of the studies were assessed before undertaking the study.

The minimum number of the SLR's primary studies is eight, and the maximum is 86. The mean value is 33, and the median is 26.5, which means the number of reviewed papers is right-skewed. The Spearman rank correlation between the year of publication and the number of reviewed papers is 0.50 (p-value = 0.034), which indicates that the research on cost/effort estimation in agile software development tends to increase moderately over the year. However, we did not check the overlap of primary studies between SLRs.

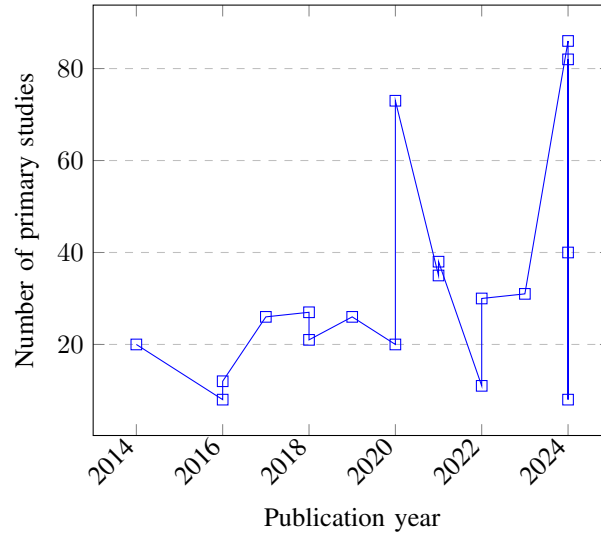


Fig. 1. Temporal distribution of the SLRs and the number of primary studies included in each of the SLR

The number of authors involved in the SLR studies varied between two and eight. The average number of SLR authors is three. One would argue that the reason for involving more persons in an SLR study is to reduce the complexity of reviewing the number of papers returned by the search string from all digital libraries. However, the Spearman rank correlation between the number of papers and the number of authors is 0.194 with a p-value of 0.471. This indicates no strong or significant relationship between the number of authors and the reviewed papers of the SLR.

B. RQ2: Characteristics of the SLRs

1) *Aims and objectives*: By coding the research questions of the identified SLRs, we identified eight themes that capture the aims of these SLRs, as shown in Table II (columns two to nine). Over the years, the main objective of secondary studies has been identifying existing effort/cost estimation techniques (17 reviews). This includes using machine learning models for effort estimation (3 reviews). The second main objective is to assess the performance of effort/cost estimation techniques in terms of their accuracy or effectiveness (12 reviews). In addition, five SLR studies reported the barriers or challenges of effort estimation in agile software development, and four discussed the solutions to address them.

Effort/cost estimation techniques have been used in different contexts, such as planning activities (SLR1), development activities (SLR1, SLR4, SLR9, SLR16) and mobile apps development (SLR6). Only one study investigates the types of empirical studies on effort estimation, and one study aggregates the benefits of estimation for companies.

2) *Years coverage*: Fig. 2 shows the years of coverage of the SLRs on cost/effort estimation in agile software development. The median and mean values of the number of primary studies are 26.5 and 33, with a standard deviation of 23.20. The longest review period covered by the reviewed papers in an SLR is 24 years (SLR13). The SLR was published in the International Journal of Computer Applications. However, only 30 papers were reviewed in the SLR. The SLR showed that the first study on cost/effort estimation in agile software development was published in 1997.

TABLE II
THE AIMS OF THE REVIEWS BASED ON THE RESEARCH QUESTIONS

SLR ID	Benefits of estimation for companies	Estimation metrics	Context where effort estimation techniques are applied	Types of estimation techniques	Assessment of the techniques	Challenges of estimation	Solution to the challenges	Empirical studies
SLR1 [16]		✓	✓	✓	✓			
SLR2 [17]				✓	✓			
SLR3 [18]				✓	✓	✓		
SLR4 [19]		✓	✓	✓	✓			
SLR5 [20]		✓		✓	✓			✓
SLR6 [21]		✓	✓	✓	✓			
SLR7 [22]		✓		✓	✓			
SLR8 [23]				✓	✓			
SLR9 [24]			✓	✓	✓			
SLR10 [25]				✓	✓	✓		
SLR11 [26]		✓		✓	✓			
SLR12 [27]				✓	✓			
SLR13 [28]				✓	✓	✓		
SLR14 [29]				✓	✓			
SLR15 [30]				✓	✓	✓	✓	
SLR16 [31]	✓		✓	✓	✓	✓		
SLR17 [32]				✓			✓	
SLR18 [33]							✓	

SLR15 and SRL7 are about user-story-based effort estimation. SLR15 covered 17 years but reviewed 86 papers that were published from 2006 to 2023. Whereas SLR7 covers the shortest time period from 2016 to 2018, and it includes 24 papers.

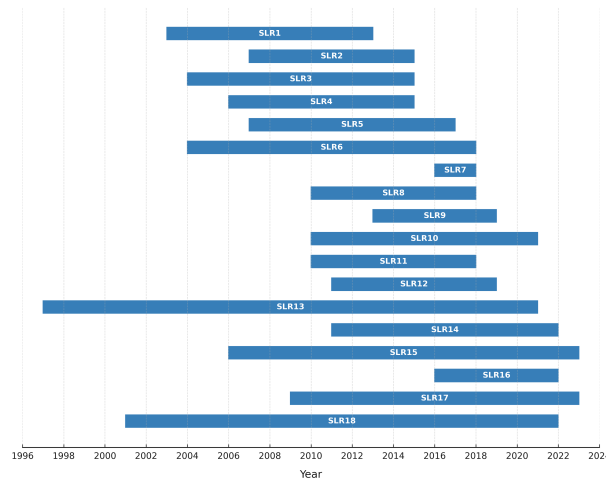


Fig. 2. Years coverage of the SLRs

3) *Quality score*: The average quality score of the SLRs is 4.1, while the minimum score is two, and the maximum is five. It can be seen in Fig. 3, among the five criteria to assess the quality of tertiary study [12], almost half of the SLRs did not report the quality assessment of their primary studies.

The majority of the SLRs have explicit inclusion/exclusion criteria and use at least four digital libraries such as IEEE Xplore, ACM Digital Library, Scopus, and Google Scholar. Three SLRs did not provide the complete list of the primary studies.

Since the DARE criteria do not question whether an SLR was warranted in the first place or not, the high DARE scores for these papers do not capture the concerns identified in this study about the overlap between the reviewed SLRS.

4) *Publication ranking*: In our dataset, 56% of the SLRs (10 reviews) are journal papers while 44% (8 reviews) are published in conferences. Of the 8 conference papers, only one study (SLR5) was published in a venue with an

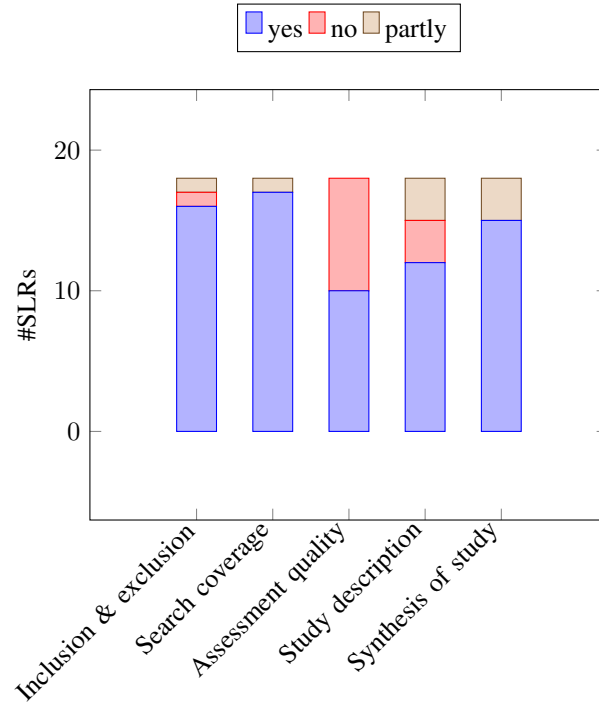


Fig. 3. SLRs' quality score using DARE criteria

A rating (Americas Conference on Information Systems) by the CORE Ranking. On the other hand, one SLR was published in an A-rated journal (SLR15), and two in C-rated journals (SLR12, SLR14). The remaining studies (15 studies) were published in non-rated venues.

Using the Scimago rating classification, four venues of the SLRs were rated Q1 (SLR9, SLR12, SLR14, and SLR17). Two venues were rated Q4 (SLR4 and SLR11). The remaining venues were not rated in Scimago.

C. RQ3: Awareness of the previous SLRs on the topic

In Table III, the SLRs are sorted by the year of publication. SLR1 is the first SLR in agile cost and effort estimation while SLR18 is the latest one. The SLRs in the rows cite the SLRs in the columns. For example, SLR1 did not cite any other SLRs, while SLR6 cited SLR1 and SLR4 (only two out of five SLRs that have been published in previous years).

From the 18 SLRs on cost/effort estimation in the agile context, more than 60% of them (11 studies) were not aware of previously published SLRs. No citation to previous SLRs was discussed or mentioned in the related work or in the discussion section. For example, three SLRs (SLR7, SLR12 and SLR15) conducted reviews on user story estimation in 2019, 2022, and 2024, respectively. We expect that newer reviews would make an extensive effort to become aware of similar or related work on the topic. Conference papers like SLR7 usually take a shorter period of time to be publicly available than journal papers. However, SLR12 did not cite SLR7, and SLR15 did not cite SLR7 or SLR12.

Some of the reviews treated previous SLRs as the primary studies instead, which makes them prone to the threat of double-counting [14]. For example, in SLR3, SLR1 is considered one of the primary studies.

Fig. 4 shows the SLRs citation network. The direction of the arrow shows which papers cite which paper. SLR1 and SLR9 are the most cited studies in all reviews, by six and five SLRs, respectively. The authors of SLR9 mentioned that their study was an update of SLR1.

D. RQ4: Justification for undertaking another SLR on the topic

The first SLR on the topic (SLR1) was published in 2014 and argued that there were no existing reviews that aggregate the evidence of cost/effort estimation in the agile context. Since then, 17 SLRs on the topic have been

TABLE III
CITATION OF SLRS ON COST/EFFORT ESTIMATION IN THE AGILE CONTEXT

	SLR1	SLR2	SLR3	SLR4	SLR5	SLR6	SLR7	SLR8	SLR9	SLR10	SLR11	SLR12	SLR13	SLR14	SLR15	SLR16	SLR17	SLR18	
SLR1	-																		
SLR2		-																	
SLR3			-																
SLR4				-															
SLR5					-														
SLR6	✓			✓		-													
SLR7							-												
SLR8								-											
SLR9	✓		✓	✓		✓			-										
SLR10										-									
SLR11											-								
SLR12	✓		✓					✓	✓			-							
SLR13													-						
SLR14	✓								✓					-					
SLR15	✓			✓					✓				✓		-				
SLR16									✓							-			
SLR17	✓								✓								-		
SLR18																			-

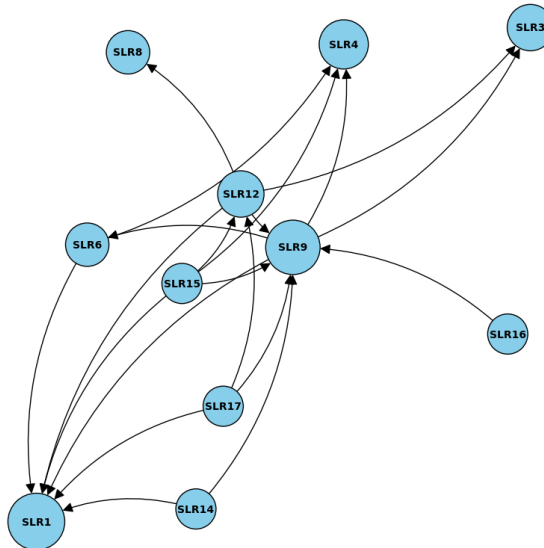


Fig. 4. SLRs citation network

published for various reasons. For example, three SLRs claimed that there is a lack of existing reviews in the area of agile estimation for mobile applications (SLR6), estimating the complexity in user stories (SLR7), and machine learning estimation models of Scrum-based projects (SLR8). However, even though SLR11 argued that there was a lack of existing reviews, their first research question is similar to those in SLR1. Yet SLR11 did not cite any previous SLRs, including SLR1.

“What kind of method is used for effort estimation in agile?” – RQ1, SLR11

“What techniques have been used for effort or size estimation in agile software development?” – RQ1, SLR1

As shown in Table IV, 44% of the SLRs (8 studies) did not provide any reasons or justification for why another SLR is needed, even though they were aware of the existing SLRs on the topic. For example, SLR15 discussed several previous SLRs in the area (e.g., SLR1, SLR4, SLR9, and SLR12), but the study did not provide any details

TABLE IV
SLRS ON COST/EFFORT ESTIMATION IN THE AGILE CONTEXT

SLR ID	Published Year	Claimed justification for another SLR	Cited SLRs	previous	Years coverage	Quality score	# Primary studies
SLR1	2014	Lack of existing reviews	-		10	5	20
SLR2	2016	No justification	-		8	4.5	8
SLR3	2016	No justification	-		11	5	12
SLR4	2017	No justification	-		9	3	26
SLR5	2018	Lack of existing reviews	-		10	2	27
SLR6	2018	Lack of existing reviews	SLR1, SLR4		14	5	21
SLR7	2019	Lack of existing reviews & Different focus than existing reviews	-		2	5	26
SLR8	2020	Lack of existing reviews	-		8	5	20
SLR9	2020	Temporal update	SLR1, SLR3, SLR4, SLR6,		6	4	73
SLR10	2021	No justification	-		11	5	35
SLR11	2021	Lack of existing reviews	-		8	4	38
SLR12	2022	Rapid technology advancement & Different focus than existing reviews	SLR1, SLR3, SLR8, SLR9,		8	2.5	11
SLR13	2022	No justification	-		24	4.5	30
SLR14	2023	Temporal update	SLR1, SLR9		11	5	31
SLR15	2024	No justification	SLR1, SLR4, SLR9, SLR12		17	3	86
SLR16	2024	No justification	SLR9		6	5	8
SLR17	2024	Different focus than existing reviews	SLR1, SLR9, SLR12		14	3	82
SLR18	2024	No justification	-		21	3.5	40

on why these SLRs are not sufficient to answer their research questions. Their first research question (“*What are the different methods that are used for effort estimation in ASD?*”) is similar to SLR1’s first research question. Similarly, SLR16 did not provide any reason for undertaking another SLR, even though they adopted the search string defined in SLR9 in their study.

IV. DISCUSSION

A. Redundancy and lack of clear justification of additional systematic reviews

Our findings underscore several critical issues prevalent in systematic literature reviews (SLRs) within the context of agile software estimation. Firstly, there is a noticeable lack of awareness among authors regarding existing literature reviews, with over 60% of the analyzed studies failing to cite or discuss prior relevant SLRs. This suggests a significant gap in the initial literature assessment process, indicating that authors either overlook or inadequately assess existing knowledge before initiating new reviews.

Moreover, justification for conducting new SLRs was frequently absent or insufficiently articulated. Half of the reviewed SLRs did not explicitly justify the necessity of their study, while those providing justifications often cited superficial or redundant reasons such as a “lack of existing reviews,” despite clear evidence to the contrary. This pattern reveals a critical need for a stronger emphasis on explicit justification and validation when proposing new secondary studies.

This situation is not unique for literature reviews on cost and effort estimation in Agile. Similar observations have been made by others in other topics such as regression testing [34], software analytics [35], [36] and DevOps [37], where a large number of secondary studies have been published. Cost and effort estimation in the context of Agile software development presents a fairly niche topic, which makes it an interesting case to study, with 18 published reviews on an apparently narrow topic.

Like any research, it is important for secondary studies to capitalize on existing research. However, the effort-intensive and time-consuming nature of SLRs [38] makes it even more important.

B. Guidelines for searching and analyzing related work

The guidelines for conducting secondary studies already emphasize the importance of establishing the need for undertaking a literature review [1]. Furthermore, a recent quality appraisal instrument for systematic literature reviews enables reviewers to assess whether a study has sufficiently justified the need [39]. In Figure 5, we further

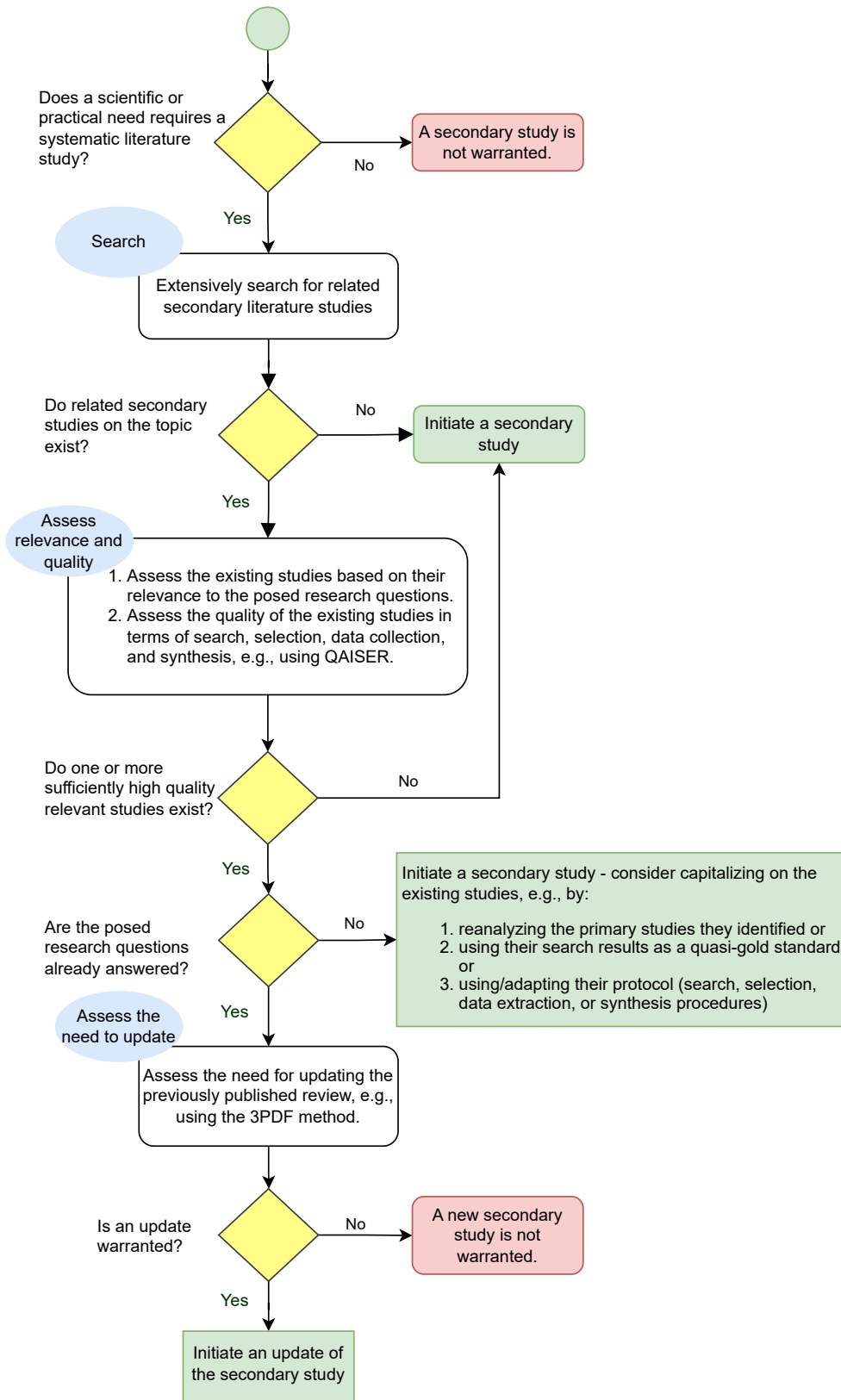


Fig. 5. Work flow to systematically identify and utilize existing systematic literature reviews.

strengthen the guidelines for researchers by presenting a workflow for systematically approaching and dealing with existing secondary studies.

In software engineering, there is no prospective register² for announcing and describing a literature review that a research group is about to undertake. So, there will always be some unintended duplication of effort. However, our proposed workflow (see Figure 5) intends to reduce the likelihood of overlooking or not capitalizing on existing secondary studies.

The three main activities in this workflow are 1) search for existing reviews, 2) assessment of existing reviews, and 3) assessment of the need for an update.

- 1) **Search:** For search, we propose a general search string that is composed of two blocks of keywords joined by a Boolean ‘and’.

Block 1: keywords to identify systematic literature studies including: “systematic review” OR “research review” OR “research synthesis” OR “research integration” OR “systematic overview” OR “systematic research synthesis” OR “integrative research review” OR “integrative review” OR “systematic literature review” OR “literature review” OR “scoping review” OR “systematic map” OR “systematic mapping” OR “mapping study” OR “tertiary study” OR “tertiary review”³

AND

Block 2: topic-specific keywords, e.g., in this study, we used effort estimation and agile-related keywords.

- 2) **Assess relevance and quality:** For the assessment of existing reviews, we recommend the use of Quality Assessment Instrument for Software Engineering systematic literature Reviews (QAISER) [39]. It is a specialized tool that adapts the healthcare-focused AMSTAR 2 assessment instrument, tailoring it specifically for software engineering contexts. QAISER incorporates 15 evaluation criteria that assess key elements of SLR quality, such as the clarity and relevance of research questions, the comprehensiveness of literature searches, the appropriateness of methods for study selection, data extraction, and synthesis procedures.

A structured evaluation process within QAISER allows reviewers to systematically identify strengths and weaknesses in the SLR’s need, methodology, and findings.

The development of QAISER included validation through input from four external software engineering experts (including leading guideline authors like Prof. Barbara Kitchenham and Prof. Kai Petersen). Its utility has been demonstrated through practical applications where the tool was successfully employed to appraise six systematic literature studies.

- 3) **Assess the need to update:** In the third activity in the workflow, when deciding whether a previous review should be updated, we suggest the use of the guidelines by Mendes et al. [40].

C. Threats to validity

Threats to validity related to the results from this study are discussed below.

1) *Search comprehensiveness:* We only constrained our study to SLR and did not look for mapping studies. Our search strings also yielded some mapping studies, and those were excluded. This means that we may have overlooked relevant papers. However, we think having already identified 18 relevant SLRs is not a major threat to the findings of this study.

2) *Researchers bias:* To reduce the researchers’ bias, two reviewers were involved in the selection of the SLRs and in judging the claimed justification reported in the papers. One reviewer performed the data extraction using an extraction form. Both reviewers then discussed and evaluated a pilot with three papers. Any ambiguities and confusion about the extraction process were resolved.

3) *Generalization:* In this study, we only focus on SLRs on one topic, which is cost/effort estimation in agile software development. Thus, the problem may not be generally true in other secondary studies in other topics within software engineering domain.

V. CONCLUSION

This study critically examined the landscape of systematic literature reviews (SLRs) on effort and cost estimation in agile software development. Our analysis of 18 SLRs revealed substantial redundancy, insufficient awareness

²<https://www.crd.york.ac.uk/prospero/>

³Although we are looking for existing systematic literature reviews, we recommend that the researchers make a more inclusive search for non-systematic literature reviews, mapping studies, and any tertiary studies.

of prior reviews, and a lack of clear justification for conducting new secondary studies. These findings highlight systemic issues in how SLRs are initiated and positioned within the empirical software engineering community.

To mitigate these challenges, we propose a structured workflow to approach related work comprising three key activities: a comprehensive search for related reviews, rigorous quality appraisal using tools such as QAISER, and a principled evaluation of the need for an updated SLR. This workflow aims to support researchers in making evidence-informed decisions about initiating secondary studies and to enhance the transparency and utility of their contributions.

Our findings contribute to the broader discourse on improving research practices in software engineering by emphasizing the importance of methodological justification, reuse of prior evidence, and editorial scrutiny. We call upon the community—researchers, reviewers, and venue organizers alike—to adopt and reinforce practices that prevent redundant research and promote cumulative, high-quality evidence synthesis.

Future work can be done by broadening our sample by collecting and analysis the systematic mapping studies or any other types of secondary studies in cost/effort estimation in agile software development. Thus, with a more comprehensive search, the findings will complement our study. Another venue for future work is to replicate our study in other topics within software engineering. By contrasting and comparing the secondary studies in various topics, we can improve the generalizability of the findings.

Future work could also further investigate the list of primary studies identified by the SLRs in our sample. We will explore the overlap between the primary studies of the SLRs that focus on the same research questions. Such a study could confirm or refute our findings.

ACKNOWLEDGEMENTS

This work was supported by ELLIIT, the Swedish government-funded Strategic Research Area within IT and Mobile Communications, and a research grant for the GIST project (reference number 20220235) from the Swedish Knowledge Foundation.

REFERENCES

- [1] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based software engineering and systematic reviews*. CRC press, 2015.
- [2] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering,” EBSE Technical Report, Tech. Rep., 2007.
- [3] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, “Systematic literature reviews in software engineering—a systematic literature review,” *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.
- [4] N. B. Ali and M. Usman, “Reliability of search in systematic reviews: Towards a quality assessment framework for the automated-search strategy,” *Information and Software Technology*, vol. 99, pp. 133–147, 2018.
- [5] X. Wang, H. Edison, D. Khanna, and U. Rafiq, “How many papers should you review? a research synthesis of systematic literature reviews in software engineering,” in *Proceedings of 17th International Symposium on Empirical Software Engineering and Measurement*, 2023.
- [6] K. Petersen, S. Vakkalanka, and L. Kuzniarz, “Guidelines for conducting systematic mapping studies in software engineering: An update,” *Information and software technology*, vol. 64, pp. 1–18, 2015.
- [7] B. Kitchenham, L. Madeyski, and D. Budgen, “Segress: Software engineering guidelines for reporting secondary studies,” *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 1273–1298, 2022.
- [8] N. B. Ali and M. Usman, “A critical appraisal tool for systematic literature reviews in software engineering,” *Inf. Softw. Technol.*, vol. 112, pp. 48–50, 2019. [Online]. Available: <https://doi.org/10.1016/j.infsof.2019.04.006>
- [9] M. Usman, N. B. Ali, and C. Wohlin, “A quality assessment instrument for systematic literature reviews in software engineering,” *e-Infomatica Software Engineering Journal*, vol. 17, no. 1, 2023.
- [10] P. Ralph, N. b. Ali, S. Baltes, D. Bianculli, J. Diaz, Y. Dittrich, N. Ernst, M. Felderer, R. Feldt, A. Filieri *et al.*, “Empirical standards for software engineering research,” *arXiv preprint arXiv:2010.03525*, 2020.
- [11] D. Costal Costa, C. Farré Tost, J. Franch Gutiérrez, and C. Quer, “How tertiary studies perform quality assessment of secondary studies in software engineering,” in *24th Iberoamerican Conference on Software Engineering (CIBSE 2021): San Jose, Costa Rica, 30 August-3 September 2021*. Curran Associates, Inc., 2021, pp. 56–69.
- [12] D. Budgen, P. Brereton, N. Williams, and S. Drummond, “What support do systematic reviews provide for evidence-informed teaching about software engineering practice?” *e-Infomatica Software Engineering Journal*, vol. 14, no. 1, pp. 7–60, 2020.
- [13] R. Britto, M. Usman, and E. Mendes, “Effort estimation in agile global software development context,” in *Proceedings of International Conference on Agile Software Development*, 2014, pp. 182–192.
- [14] J. Börstler, N. bin Ali, and K. Petersen, “Double-counting in software engineering tertiary studies—an overlooked threat to validity,” *Information and Software Technology*, vol. 158, p. 107174, 2023.
- [15] H. Edison and N. Ali, “The dataset for the paper Another Systematic Review? A Critical Analysis of Systematic Literature Reviews on Agile Effort and Cost Estimation,” 2025. [Online]. Available: https://figshare.com/articles/dataset/Another_Systematic_Review_A_Critical_Analysis_of_Systematic_Literature_Reviews_on_Agile_Effort_and_Cost_Estimation_-_Supplement_Materials/29542682
- [16] M. Usman, E. Mendes, F. Weidt, and R. Britto, “Effort estimation in agile software development: a systematic literature review,” in *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, 2014, p. 82–91.
- [17] S. Bilgaiyan, S. Mishra, and M. Das, “A review of software cost estimation in agile software development using soft computing techniques,” in *Proceedings of the 2nd International Conference on Computational Intelligence and Networks*, 2016, pp. 112–117.

- [18] T. Schweighofer, A. Kline, and L. Marjanherić, “How is effort estimated in agile software development projects?” in *Proceedings of the 5th Workshop of Software Quality, Analysis, Monitoring, Improvement, and Applications*, 2016.
- [19] S. Bilgaiyan, S. Sagnika, S. Mishra, and M. Das, “A systematic review on software cost estimation in agile software development,” *Journal of Engineering Science & Technology Review*, vol. 10, no. 4, pp. 51–64, 2017.
- [20] E. D. Canedo, D. P. Aranha, M. de Oliveira Cardoso, R. P. da Costa, and L. L. Leite, “Methods for estimating agile software projects: Systematic literature review,” in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering*, 2018, pp. 34–39.
- [21] A. R. Altaieb and A. M. Gravell, “Effort estimation across mobile app platforms using agile processes: A systematic literature review,” *Journal of Software*, vol. 13, no. 4, 2018.
- [22] M. Durán, R. Juárez-Ramírez, S. Jiménez, and C. Tona, “Taxonomy for complexity estimation in agile methodologies: A systematic literature review,” in *Proceedings of the 7th International Conference in Software Engineering Research and Innovation*, 2019, pp. 87–96.
- [23] M. Arora, S. Verma, Kavita, and S. Chopra, “A systematic literature review of machine learning estimation approaches in scrum projects,” in *Cognitive Informatics and Soft Computing*, ser. *Advances in Intelligent Systems and Computing*, P. K. Mallick, V. E. Balas, A. K. Bhoi, and G.-S. Chae, Eds. New York, NY: Example Publisher, 2020, vol. 1040, pp. 573–586.
- [24] M. Fernández-Diego, E. R. Méndez, F. González-Ladrón-De-Guevara, S. Abrahão, and E. Insfran, “An update on effort estimation in agile software development: A systematic literature review,” *IEEE Access*, vol. 8, pp. 166 768–166 800, 2020.
- [25] F. Aizaz, U. I. Janjua, H. Zafar, J. A. Khan, and I. Kazim, “An empirical investigation on software cost estimation techniques and barriers on agile software development in software industry of pakistan,” in *Proceedings of International Conference on Frontiers of Information Technology*, 2021, pp. 194–199.
- [26] P. Sudarmaningtyas and R. Mohamed, “A review article on software effort estimation in agile methodology,” *Journal Science & Technology*, vol. 29, no. 2, pp. 837–861, 2021.
- [27] B. Alsaadi and K. Saeedi, “Data-driven effort estimation techniques of agile user stories: a systematic literature review,” *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5485–5516, 2022.
- [28] P. Tandon, U. Suman, and M. Rathore, “Systematic literature review on effort estimation in agile software development using machine learning techniques,” *International Journal of Computer Applications*, vol. 184, no. 21, pp. 15–23, 2022.
- [29] S. Abusaeed, S. U. R. Khan, and A. Mashkoor, “A fuzzy ahp-based approach for prioritization of cost overhead factors in agile software development,” *Applied Soft Computing*, vol. 133, 2023.
- [30] M. Iqbal, M. Ijaz, T. Mazhar, T. Shahzad, Q. Abbas, Y. Ghadi, W. Ahmad, and H. Hamam, “Exploring issues of story-based effort estimation in agile software development (asd),” *Science of Computer Programming*, vol. 236, 2024.
- [31] H. R. Moechtar, S. Sulistyono, and R. Ferdiana, “A review on improving the accuracy of effort estimation in software development with agile method,” in *Proceedings of the 11th International Conference on Information Technology, Computer, and Electrical Engineering*, 2024, pp. 31–36.
- [32] S. K. Jirat Pasuksmit, Patanamon Thongtanunam, “A systematic literature review on reasons and approaches for accurate effort estimations in agile,” *ACM Computing Surveys*, vol. 56, 2024.
- [33] A. T. Sinaga, T. Raharjo, and N. W. Trisnawaty, “Exploring agile effort estimation issues: A systematic literature review,” *Jurnal Ilmu Pengetahuan and Teknologi Komputer*, vol. 10, no. 2, pp. 415–424, 2019.
- [34] N. B. Ali, E. Engström, M. Taromirad, M. R. Mousavi, N. M. Minhas, D. Helgesson, S. Kunze, and M. Varshosaz, “On the search for industry-relevant regression testing research,” *Empirical Software Engineering*, vol. 24, pp. 2020–2055, 2019.
- [35] M. Laiq, N. b. Ali, J. Börstler, and E. Engström, “Software analytics for software engineering: A tertiary review,” *arXiv preprint arXiv:2410.05796*, 2024.
- [36] —, “What do we know about software analytics research? a critical review of secondary studies,” in *51st Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2025, Salerno, Italy, September 10-12, 2025*, 2025.
- [37] A. Hrusto, “Paper v in the thesis titled: Enhancing DevOps with Autonomous Monitors: A Proactive Approach to Failure Detection,” Doctoral Thesis (compilation), Computer Science, Lund University, Lund, Oct. 2024, iSBN: 9789181042092.
- [38] H. Zhang and M. A. Babar, “Systematic reviews in software engineering: An empirical investigation,” *Information and software technology*, vol. 55, no. 7, pp. 1341–1354, 2013.
- [39] M. Usman, N. B. Ali, and C. Wohlin, “A quality assessment instrument for systematic literature reviews in software engineering,” *e-Infomatica Software Engineering Journal*, vol. 17, no. 1, p. 230105, Feb. 2023, available online: 09 Feb. 2023.
- [40] E. Mendes, C. Wohlin, K. Felizardo, and M. Kalinowski, “When to update systematic literature reviews in software engineering,” *Journal of Systems and Software*, vol. 167, p. 110607, Sep. 2020.