IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 25, 2020, accepted August 14, 2020, date of publication September 3, 2020, date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021664

An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review

MARTA FERNÁNDEZ-DIEGO[1], ERWIN R. MÉNDEZ[2], FERNANDO GONZÁLEZ-LADRÓN-DE-GUEVARA[1], SILVIA ABRAHÃO[3], (Member, IEEE), AND EMILIO INSFRAN[3], (Member, IEEE)

[1]Department of Business Organization, Universitat Politècnica de València (UPV), 46022 Valencia, Spain
[2]IMF Business School, 28012 Madrid, Spain
[3]Department of Computer Science, Universitat Politècnica de València (UPV), 46022 Valencia, Spain

Corresponding author: Fernando González-Ladrón-de-Guevara (fgonzal@omp.upv.es)

This work was supported by the Spanish Ministry of Science, Innovation and Universities through the Adapt@Cloud Project under Grant TIN2017-84550-R.

ABSTRACT Software developers require effective effort estimation models to facilitate project planning. Although Usman *et al.* systematically reviewed and synthesized the effort estimation models and practices for Agile Software Development (ASD) in 2014, new evidence may provide new perspectives for researchers and practitioners. This article presents a systematic literature review that updates the Usman *et al.* study from 2014 to 2020 by analyzing the data extracted from 73 new papers. This analysis allowed us to identify six agile methods: Scrum, Xtreme Programming and four others, in all of which expert-based estimation methods continue to play an important role. This is particularly the case of Planning Poker, which is very closely related to the most frequently used size metric (story points) and the way in which software requirements are specified in ASD. There is also a remarkable trend toward studying techniques based on the intensive use of data. In this respect, although most of the data originate from single-company datasets, there is a significant increase in the use of cross-company data. With regard to cost factors, we applied the thematic analysis method. The use of team and project factors appears to be more frequent than the consideration of more technical factors, in accordance with agile principles. Finally, although accuracy is still a challenge, we identified that improvements have been made. On the one hand, an increasing number of papers showed acceptable accuracy values, although many continued to report inadequate results. On the other, almost 29% of the papers that reported the accuracy metric used reflected aspects concerning the validation of the models and 18% reported the effect size when comparing models.

INDEX TERMS Effort estimation, agile methods, agile software development (ASD), systematic literature review (SLR).

I. INTRODUCTION

The research community has devoted a great deal of attention to agile software development ever since the publication in 2001 of the Agile Manifesto and its 12 principles.[1] However, the methods and practices that led to what is now widely known as Agile were being used and studied long before. The idea behind the formulation of the Manifesto was to find some common ground and propose an alternative approach to the software development processes applied in the previous 40 years [1]. The term Agile is basically used to refer to a variety of approaches, techniques, methods and practices that fulfill the values and principles expressed in that Manifesto [2].

Agile Software Development (ASD) is often presented as an alternative to more traditional approaches, such as waterfall, incremental, or evolutionary, in which predictability, extensive planning, codified processes and rigorous reuse are the key elements for the efficient development of software [3]. ASD is, however, based on iterative and incremental development models [4], promotes a rapid response to changes and focuses on customer satisfaction, timely and continuous delivery, informal methods and minimal planning [5].

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

[1]agilemanifesto.org

Some of the best known development methods in ASD are Scrum [6], [7], Xtreme Programming [8], Feature Driven Development [9], Lean Software Development [10], Adaptive Software Development [11], Crystal Methodologies [12] and the Dynamic Systems Development Method [13].

Effort estimation plays an important and critical role in any software development project. Effort estimation can be defined as the process by which effort is evaluated, and estimation is carried out in terms of the amount of resources required to end project activity in order to deliver a product or service that meets the given functional and non-functional requirements to a customer [14].

Software developers require effective effort estimation models to facilitate project planning and for the eventual successful implementation of the project [15]. If the effort estimations are accurate, they can contribute to the success of software development projects, while incorrect estimations can negatively affect companies' marketing and sales, leading to monetary losses [16]. Software project estimation involves the estimation of the effort, size, staffing, schedule (time) and cost involved in creating a unit of the software product [17], [18].

A number of studies with which to estimate effort in agile software development have been performed in recent decades. The estimation techniques can be classified into two major types, namely algorithmic and non-algorithmic techniques [19]. The former are based on equations and mathematics, which are used to process the software estimation, while the latter are based on analogy and deduction.

In spite of the vast number of approaches, the accuracy of software effort estimation models for agile development still remains inconsistent. There is consequently a need to gather, synthesize and validate evidence from existing studies in order to build a corpus of knowledge so as to provide direction to researchers and practitioners [18]. One strategy employed to summarize existing literature for a particular research domain is that of conducting a Systematic Literature Review (SLR) [20]. Furthermore, when estimating effort in agile development projects various challenges appear, the most important of which is how to properly integrate effort estimation with good agile development practices.

Several SLRs have been performed [16], [21]–[24] in order to address these challenges. These studies synthesize relevant research on effort estimation in agile software development (for a more comprehensive discussion of these studies, please refer to Section II). However, these studies provide a snapshot of only the knowledge available at that time. Newly identified studies could alter the conclusion of a review, and if they have not been included, this threatens the validity of the review, and, at worst, means the review could mislead.

This article, therefore, presents an updated SLR on effort estimation in agile software development which is based on the study carried out by Usman *et al.* [21] in 2014, hereafter denominated as the Original Study. Our objective is to expand the timeframe for the publication of the primary studies from December 2013 to January 2020, in an effort to attain answers to the same research questions stated in the Original Study:

- RQ1: What methods have been used to estimate size or effort in ASD?
- RQ2: What effort predictors have been used for effort estimation in ASD?
- RQ3: What are the characteristics of the dataset used for size or effort estimation in ASD?
- RQ4: What agile methods in ASD have been investigated for size or effort estimation?

The temporal update was performed following the original review protocol, with some minor changes that will be justified in Section IV. This work compares our results with those of the Original Study without integrating them; in other words, without merging both subsets of primary studies. The idea is to reinforce some of the results of the Original Study and to highlight the changes in the practices and research on effort estimation in agile software development produced in the subsequent four years.

The remainder of this article is organized as follows. In Section II, we present the related studies and a summary of the main findings of the Original Study. In Section III, we then go on to justify the need for this update. In Section IV, we describe the SLR process followed: research questions, search strategy for primary studies, inclusion and exclusion criteria and the selection of primary studies, taking the quality assessment into consideration. In Section V, we present the results of the primary studies, including a bibliometric analysis and the answers to the research questions. In Section VI, we discuss the main findings of this study, the study limitations and the implications of our results for research and practice. Finally, in Section VII we outline the conclusions obtained and the future lines of research.

II. RELATED WORK

There are numerous studies on effort estimation in agile software development projects. This has motivated several authors to carry out work aimed at summarizing the current knowledge regarding this topic. Some of these reviews (e.g., [25], [26]) are rather informal and present only an overview of the strengths and limitations of the effort estimation approaches used in agile software development. In this section, we focus on discussing the existing SLRs that deal with this topic.

In 2017, Hoda *et al.* [27] presented a tertiary study with the objective of reviewing all the SLRs published on agile software development research topics. Of these studies, only one SLR on effort estimation in agile development was found, and this corresponded to the study conducted by Usman *et al.* [21] (the Original Study) in 2014. It is based on 20 papers published from 2001 (when the Agile Manifesto was launched) to November 2013. Of these studies, only seven were journal papers. Since some of the papers reported more than one study, a total of 25 primary studies were finally selected. The objective of the study was to investigate which techniques

had been used for effort or size estimation in ASD, which effort predictors had been used during effort estimation for ASD, what the characteristics of the dataset/knowledge used were, and which agile methods for effort estimation had been investigated.

Overall, Usman *et al.* [21] found that although a variety of estimation techniques had been applied in an ASD context, those used most were the techniques based on some form of expert-based subjective assessment. In addition, most of the techniques had not attained acceptable prediction accuracy values in regards to how close the estimated values are to the actual values. Namely, most cases did not turn out to meet the 25% threshold suggested by Conte *et al.* [28] to assess the effort prediction accuracy, i.e., acceptable MMRE if calculated value $<= 25\%$ and acceptable Pred(25%) if calculated value $>= 75\%$. These threshold values are used frequently to specify the acceptable accuracy of effort estimation methods in literature [29]. Besides, there was little agreement on suitable cost drivers for ASD projects, and most of the estimation studies used within-company industrial datasets. More importantly, the authors concluded that practitioners would have little guidance from the current literature on effort estimation in ASD because the techniques had a low level of accuracy and there was little consensus as to what the appropriate cost drivers were for different agile contexts. This motivated the appearance of other SLRs on the topic.

A second study was conducted by Schweighofer *et al.* [22] in 2016. This SLR was based on 12 studies retrieved from 2000 to 2015. The objective of the study was to investigate which methods for the performance of effort estimation in ASD were available, how objective the estimation was, what influenced the estimation, and how accurate those methods and approaches were. The results showed that the majority of the studies employed methods based on subjective expert effort estimation, including techniques such as Planning Poker, Expert Judgment and Story Points. Estimation by analogy was also frequently used.

However, other estimation methods that are not based on expert judgment (e.g., COCOMO, SLIM, regression analysis) are not so frequently used in ASD. With regard to the factors that influence the estimation, the results show that personnel factors come before project factors in the ASD effort estimation process. This means that the group of experts' levels of knowledge and skill are essential for the estimation, as is the ability to form proper working teams. Although the study was target toward analyzing the accuracy of the approaches employed, the results did not provide a comprehensive overview of this. The authors concluded that although agile software development methods emerged in Europe as early as the year 2000, relatively few papers provide empirical knowledge on effort estimation.

An SLR on software cost estimation in ASD was conducted by Bilgaiyan *et al.* [23] in 2017. It was based on 26 primary studies published from 2006 to 2015. The objective of the study was to investigate different questions related to the estimation mechanisms used for agile software development methods, the parameters that define their accuracy, the comparative accuracies achieved by different estimation techniques, the suitable circumstances in which the estimation techniques can be applied, what problems that can be confronted when applying the techniques, etc.

The authors found that the estimation techniques applied in ASD and other development environments were Neural Networks, Expert Judgment, Planning Poker/Disaggregation, Use Case Point, Modified Use Case Point, Linear Regression, Wideband Delphi and Bottom-up/Top-down. They also found that neural networks and expert estimation are the most popular of the existing conventional estimation methods for ASD. As future findings, they anticipated the potentiality of using soft computing techniques, especially swarm and colony optimization algorithms, and the need to optimize the existing estimation techniques with more empirical outcomes in different test environments.

An SLR whose objective was to identify the metrics and methods most frequently used in ASD and the size metrics most frequently employed as regards effort estimates, deadlines and costs was perfomed in [30]. The authors selected 27 articles from between 2007 and 2018 for data extraction.

On the one hand, some of the most frequently used techniques are Planning Poker, Expert Opinion and Function Point Analysis. On the other, the metrics most frequently employed for estimates are Story Points and Function Points. The authors showed that the methods and the metrics for estimates are mostly applied in a given context of agile development, with adaptations in order to fit the context of project.

Another study on effort and size estimation in ASD was published by Altaleb and Gravell [16] in 2018. Unlike the previous studies, this study was specific to the domain of mobile application development. The objective of the study was to investigate: i) which methods had been used to estimate effort in mobile application development using agile methods; ii) which effort predictors had been used; iii) what the characteristics of the datasets that had been used were; and iv) how accurate and efficient the estimation methods were. The authors summarized the estimation techniques used for mobile apps, focusing particularly on the software estimation techniques that are applicable to the ASD process. The SLR was based on 21 studies published from 2008, the year in which the Apple App Store and the Android market were launched, to 2018. Of these studies, 13 concerned software estimation techniques in mobile apps and the other eight concerned agile software development in mobile apps.

The results showed that the techniques most commonly used for mobile apps were Function Size Measurement and Expert Judgment. The planning and development of mobile applications differs from other traditional software applications owing to the characteristics of the mobile environment, such as high autonomy requirements, market competition, and many other constraints. With regard to the size metrics and cost drivers, the results showed that the number of screens and type of supported platform for smartphones were the

most common factors used to measure the estimation prediction.

An analysis of the SLRs discussed above indicates that they pursued very similar research questions, but that the knowledge of effort estimation in agile software development is still dispersed. The work of Usman *et al.* [21] was, therefore, selected as the Original Study. On the one hand, [16] is domain-specific within the context of ASD, focusing only on mobile application development. On the other, we unfortunately identified several problems related to the design and reporting of the results in [22], [23] and [30]. In [22], the inclusion and exclusion criteria were not properly defined and the analysis was based only on 12 studies, one of which was [21], thus compromising the generalization of the results. [21] is a secondary study and SLRs are supposed to include only primary studies. In [23], the research questions were not well formulated as they covered more than one concept. In [30], the digital libraries consulted in the automatic search were not specified. Only 27 studies were retrieved, although books and pioneering articles were also considered. Similarly to that which occurred in [22], [21] was one of the 27 primary studies. Furthermore, none of these SLRs performed a quality assessment of the papers reviewed. Finally, it is unclear how the data extracted from the papers was used to answer the research questions, as a consequence of which, the results reported in [22], [23] and [30] may not be reliable.

Finally, it is worth mentioning that we recently found another update of the SLR conducted by Usman *et al.* [21] (the Original Study). This study was conducted by Dantas *et al.* [24] and followed a forward snowballing approach to select 24 new papers published from 2014 to 2017. It is for this reason that particular attention will, throughout this article, be paid to discussing the differences observed in [24] with respect to our study, which includes 73 new papers, and to integrating the results of their work when appropriate.

Moreover, in 2019, two research questions related to effort estimation in ASD were addressed in [31]: What are the existing methods used and what are the existing metrics for size? The authors selected 38 primary studies for this SLR from searches conducted separately in IEEE Xplore, ACM Digital Library, SCOPUS and ScienceDirect, from 2008 until August 2018. The research concerns only the mapping of the primary studies onto the estimation methods and size metrics used in ASD.

III. THE NEED FOR AN UPDATE

Systematic Literature Reviews (SLRs) are usually proposed as a framework in which to consolidate all reliable empirical research on a specific topic so as to reduce reviewer bias and follow a structured process to gain repeatable results [20]. In contrast to an ad hoc literature review, an SLR is a methodologically rigorous review of research results. Keeping SLRs updated is essential as regards prolonging their lifespan, because SLRs may lose their impact over the years when they become out of date or misleading [32]. In fact, the main purpose of updating SLRs, Systematic Mapping Studies and Tertiary Studies is to keep evidence as up-to-date as possible.

While updates of SLRs are very common in other fields, few have been found in software engineering to date [33]. Da Silva *et al.* [34] analyzed the quality, coverage of topics and potential impact of published SLRs as regards software engineering. In 2009, the authors did not find any updates of previous SLRs among their 120 secondary studies. In the field of healthcare, however, it is even possible to find methods with which to evaluate when an SLR should be updated [35], [36].

Kitchenham, Brereton and Budgen were the first software engineering researchers to elaborate a protocol with which to extend an existing tertiary study on SLRs [37]. There are three basic complementary means of extending and updating an SLR: a temporal update, a search extension, or a combination of both. While a temporal update expands the timeframe for the publication of the primary studies, without major changes to the original review protocol, a search extension expands the number of sources and the search strategies within the same timeframe as the original review in order to increase the coverage of the original study [34].

The authors of [38], [39] concluded that snowballing is a suitable technique by which to perform SLR updates. The use of this kind of tools to support updates, along with the collaboration between the members who participated in the previous study team, can facilitate the update process itself [33]. Both points had already been raised in [38], in which the inclusion of information about the previous study and the reuse of the protocol from the previous study were also considered as important when conducting an update.

The objective of this article is to update the Original Study on effort estimation methods in ASD. The practice of agile methods has increased considerably in recent years. However, since introducing agility by eliminating estimation is not the solution and a compromise has to be reached, the research questions stated in the previous SLR are still present, as suggested in [24]. In other words, new accumulated evidence is available and needs to be accounted for in order to update the findings of the Original Study by considering the issues that have recently appeared. This intended temporal update, which expands the timeframe for the publication of the primary studies, may, therefore, have new and relevant implications for stakeholders. To this end, we followed a qualitative synthesis approach that allowed us to better aggregate the evidence required to answer the various research questions and analyze the factors affecting the use of effort estimation techniques in agile software development in a more holistic manner.

Table 1 presents a comparison of the search strategy results in the Original Study, that of Dantas *et al.* [24] and the present study. In this regard, features such as the timeframe, the search approach, digital libraries used, the number of papers inspected, the quality threshold and the number of papers excluded owing to their low quality, the number

of primary papers and primary studies finally collected and the percentage of journal papers.

The timeframe of the Original Study covered the 12 years following the publication of the Agile Manifesto, and used eight digital libraries. The present study uses only four digital libraries to carry out a search for papers published in December 2013, when the previous study ends, and covers a timeframe of six years. Despite considering a smaller range of years and using fewer digital libraries, the number of papers retrieved in the present study is considerably greater than those of the Original Study, with approximately 122%, 176% and 192% more papers in the initial search (with duplicates), the subset after removing duplicates, i.e., the papers inspected, and the final subset of primary studies, respectively. This indicates that researchers' interest in this topic has increased significantly in recent years. At this point, it should be highlighted that, in the present study, the quality threshold was raised to 6.5.

As shown in Table 1, Dantas *et al.* [24] followed a Forward Snowballing approach [38] to find the primary studies published in the interval between 2014 and December 2017. Forward Snowballing involves searching for studies that cite the studies contained in the seed set. In the context of an SLR update, this seed set is formed of the primary studies of the previous SLR and the SLR itself that is to be updated. In fact, a relevant paper published in 2014 or later should refer to any of the primary studies from the Original Study. The authors used only Google Scholar and Scopus to review the citations of the seed set, despite the fact that the extraction of citations with the help of digital libraries, such as ACM Digital Library and IEEE Xplore would have minimized the risk of missing important papers [38]. They eventually retrieved 24 papers. Only six papers from this set of 24 have not been considered in the present study. Four papers did not satisfy the inclusion criteria, and were all rejected because we did not consider thay they reported effort or size estimation methods or techniques. The remaining two were not retrieved during our selection process. However, in [24], both were related to other studies that are part of our final set. In summary, 18 papers appear in both studies and we, therefore, consider that the extended set of 73 papers obtained in our study contributes to better fulfilling the purposes of this updated SLR.

Moreover, the Original Study took into account the fact that some of the papers could report more than one study, signifying that although the authors retrieved 20 papers, they accounted for 25 primary studies. However, Dantas *et al.* [24] followed a different criterion. They considered that, since some of the papers were by the same authors, they might represent a single study, as a result of which they retrieved a total of 24 papers mapped onto 15 primary studies. In our study, during the selection of primary studies, we verified that none of the journal papers were extensions of conference papers by the same authors. When considering the criteria of the Original Study while extracting the data from the primary studies analyzed, we were unable to identify any independent studies within one paper.

IV. METHOD

An extended revision of the guidelines for undertaking evidence-based software engineering and SLRs was published in 2015 [40], and served as the model with which to carry out this research, together with the structure of the Original Study in order to maintain the comparability. The research questions and the inclusion and exclusion criteria are the same as those of the Original Study. The process followed and the results obtained are detailed in the following sections.

Although we attempted to follow exactly the same protocol as that employed in the Original Study, some adaptations were inevitable, and are described in the text whenever appropriate. We do not believe that these adaptations have introduced bias into the review, and consequently consider that our results can be compared to those of the Original Study.

The PICOC strategy, which was suggested by Petticrew and Roberts [41] and is used to frame the research question elements in order to develop the review protocol, has been employed herein. The PICOC elements utilized in this study are described below:

- Population (P): ASD projects.
- Intervention (I): Effort/Size Estimation methods in ASD.
- Comparison (C): The findings of this updated review will be compared with the findings of the Original Study.
- Outcome (O): The accuracy of the estimation methods in ASD, along with their effort predictors and the characteristics of the dataset used.
- Context (C): Agile methods.

A. RESEARCH QUESTIONS

The Research Questions (RQ) of this update, which were obtained from the Original Study [21], are as follows:

- RQ1: What methods have been used to estimate size or effort in ASD?
  - RQ1a: What metrics have been used to determine the accuracy of effort estimation methods in ASD?
  - RQ1b: What is the level of accuracy of effort estimation methods in ASD?
- RQ2: What effort predictors have been used for effort estimation in ASD?
- RQ3: What are the characteristics of the dataset used for size or effort estimation in ASD?
- RQ4: What agile methods in ASD have been investigated for size or effort estimation?
  - RQ4a: What development activities have been investigated?
  - RQ4b: What planning levels have been investigated?

B. SEARCH PROCESS

In the Original Study, a total of eight digital libraries were used, i.e., ACM Digital Library, IEEE Xplore, ScienceDirect, SCOPUS, WOS, EI Compendex, INSPEC and SpringerLink. However, we could not access the latter three when

TABLE 1. Compararison of search strategy results.

	Usman *et al.* (2014)	Dantas *et al.* (2018)	Our study
Timeframe	2001 - Nov. 2013	2014 - 2017	Dec. 2013 - Jan. 2020
Search approach	Database search and Backward Snowballing	Forward Snowballing	Database search and Backward Snowballing
Digital libraries	ACM Digital Library, IEEE Xplore, ScienceDirect, SCOPUS, WOS, EI Compendex, INSPEC and SpringerLink	Google Scholar and SCOPUS	ACM Digital Library, IEEE Xplore, SCOPUS and WOS
Number of papers inspected	443	312	1222
Quality threshold	3.25	3.25	6.5
Number of papers excluded (low quality)	2	6	8
Number of primary papers	20	24	73
Percentage of journal papers	35%	37.5%	45.2%
Number of primary studies	25	15	73

performing this research. We also discarded ScienceDirect which permits access to only journals and books. The fact that the search does not support more than eight Boolean connectors per field and wildcards are not yet supported makes it impossible to ensure a similar search to that carried out in the other digital libraries. Furthermore, it is worth noting that Google Scholar was not selected as a data source. This SLR includes only those studies that appeared in the form of scientific peer-reviewed papers, while Google Scholar even contains unpublished papers and can be useful in finding grey literature in the broad sense of information that is outside recognised databases. The peer review process guarantees a higher level of quality. Anyway, the four digital libraries consulted are, according to [40], listed among the top sources of bibliographic references for journal articles and conference papers.

The search string from the Original Study was used in the automatic search:

(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'') 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 (software)

In the Original Study, the authors mentioned that it was necessary to customize the search string, although the specific changes are not indicated. We adjusted the search string according to the parameters required for each digital library consulted, as shown in Appendix B. The modifications were mainly intended to limit the search to the metadata of the publications (title, abstract, keywords), thus avoiding a full-text screening at this point of the search and selection process. Moreover, this served as a means to obtain more accurate and relevant results and, therefore, reduce the number of papers to be analyzed in the following phases.

Since the search process carried out in the Original Study took place in the first week of December 2013 and most of the studies indexed that month were missed, the timeframe employed for this SLR was from December 2013 to January 2020. It should be noted that only ACM Digital Library makes it possible to filter by month of publication and not only by year. The searches in the digital libraries were completed in February 2020. After performing the search, the results were stored in a piece of reference management software (Zotero[2]). Duplicates were eliminated by sorting the references in alphabetical order by digital library and title and maintaining only the first occurrence of each publication.

In order to analyze the performance of the keywords used in the search string, Table 2 associates each search term with the number of publications obtained and the percentage in relation to the total number of studies obtained after the elimination of duplicates. The search terms are grouped by topic in order to indicate which aspect of the search is being considered in each section of the search string, i.e., ASD, estimation, effort/size, field. It is important to highlight that these results are based on the data automatically imported into Zotero. In fact, 105 publications were imported without the corresponding keywords. They were completed manually and only 14 were left empty. The possible loss of data during the importation (e.g., fields considered by the digital libraries that might not have a parallel field in Zotero) could, therefore, explain why the term ''Software'' is not shown in 100% of the results. Furthermore, some keywords were not matched with any of the resulting studies, as is the case of ''Crystal Methodology'', ''Crystal Software Development'' and ''Dynamic Systems Development Method''. Moreover, the main search terms were ''Software'' (92.55%) and ''Agile'' (71.03%). Some other keywords with a considerable number of related publications were "Measur*" (Measure, Measuring,

[2]www.zotero.org

TABLE 2. Search terms performance.

Search terms	Number of papers	%
ASD:		
Agile	868	71.03%
Scrum	201	16.45%
Extreme Programming	28	2.29%
Lean Software Development	15	1.23%
Adaptive Software Development	4	0.33%
Feature Driven Development	3	0.25%
Crystal Methodology	-	-
Crystal Software Development	-	-
Dynamic Systems Development Method	-	-
Estimation:		
Measur	511	41.82%
Estimat	320	26.19%
Predict	245	20.05%
Assessment	200	16.37%
Calculat	78	6.38%
Forecast	27	2.21%
Effort/Size:		
Cost	376	30.77%
Size	347	28.40%
Effort	308	25.20%
Resource	298	24.39%
Metric	295	24.14%
Velocity	56	4.58%
User Story	27	2.21%
Sizing	18	1.47%
Field:		
Software	1131	92.55%

Measurement), "Cost" and "Size", with 41.82%, 30.77% and 28.40%, respectively.

C. INCLUSION AND EXCLUSION CRITERIA

Inclusion and exclusion criteria are used to determine which studies contain relevant and useful information and which ones will be discarded after the search process has concluded. The inclusion criteria in the Original Study were the following: a) papers related to effort or size estimation methods or techniques; b) papers based on any of the ASD methods; c) papers written in English; d) peer-reviewed conference or journal papers; e) papers reporting empirical studies on effort estimation in ASD. The following criteria were employed to exclude papers from our review: a) papers that focus only on the software maintenance phase, and b) papers that deal only with performance measurement (i.e., velocity). All of the aforementioned inclusion and exclusion criteria were applied in phases 1 and 2 of the selection process described in the following section.

D. SELECTION OF PRIMARY STUDIES

After removing the duplicated publications, we obtained a total of 1222 studies from the four digital libraries consulted.

TABLE 3. Results after full-text screening.

Reason for exclusion	Number of papers
Total from Phase 1	**175**
Not related to effort/size estimation	53
Not based on ASD	9
Not written in English	2
Not published in a conference or journal	6
No empirical evidence	29
Focuses only on the software maintenance phase	1
Deals only with performance measurement	-
No full-text available	-
Subtotal of papers eliminated	**100**
Remaining total	**75**

Only journal and conferences papers were considered in the search. This set of 1222 publications was submitted to several filtering and depuration phases, all of which are described in the following subsections. The goal of this selection process was to obtain a subset of primary studies and analyze them according to the Research Questions defined in Section III.A. The selection process was performed by a team of three researchers. In each of the phases, the number of papers was divided into three for processing. Once the processing had been completed, the researchers exchanged the papers, signifying that each paper was always reviewed at least twice. At the end of each phase, the same researcher ensured the consistency of the decisions made, particularly in controversial cases.

1) PHASE 1: FILTERING BY TITLE AND ABSTRACT

In this stage, the 1222 studies were analyzed by applying the inclusion and exclusion criteria to the title and abstract. The keywords were also considered as a third element to support the decision making process as regards whether a study was to be selected. A total of 175 studies remained, representing 14.32% of the publications reviewed in this phase.

2) PHASE 2: FULL-TEXT SCREENING

This phase consisted of reviewing the content of the studies resulting from phase 1 in order to confirm whether the inclusion and exclusion criteria were met. In the case of those papers not available in digital format, a direct request was sent to the authors via email. The university's interlibrary loan service was also used as a resource with which to obtain the full-text. Table 3 provides a summary of the results of this stage.

75 publications remained, representing 42.86% of the studies obtained after phase 1. Of those eliminated, the largest numbers corresponded to studies not related to effort or size estimation (53 studies), followed by studies that contained no evidence of the empirical application of the findings presented (29 studies). Furthermore, two studies were written in

Portuguese and six were not journal or conference papers and were, therefore, also ruled out.

3) PHASE 3: QUALITY ASSESSMENT

The quality checklist follows the model proposed by Kitchenham and Charters in [20]. At this point, all of the studies were scored on the basis of how well they satisfied the quality criteria described in Table 4. Each criterion was valued using a predefined scale (Y, N, P) to indicate: whether the study complied (Y = 1), did not comply (N = 0) or partially complied (P = 0.5) with them. The 13 quality questions in Table 4 were proposed in the Original Study, and the highest score was, therefore, 13, corresponding to the total number of quality criteria. In the Original Study, those studies that attained a score under the first quartile (3.25) were eliminated. In the present study, however, the quality threshold has been set at 6.5 in order to minimize bias and maximize internal and external validity. We consequently used the mean as the cutoff point in such a way that those studies that scored less than 6.5 were removed from the final set of primary studies.

Table 4 provides a summary of the quality assessment results, and for each criterion shows the number of papers that comply, do not comply or partially comply. This table also includes the quality assessment results of the 83 studies resulting from the snowballing phase described in Section IV-D4.

Most of the publications meet (Y) the quality criteria. In fact, the mean of the "Y" column is 58.54 of a total of 83 studies for which quality has been assessed. The most significant contrast will be noted in the case of criteria 9, 10 and 13, for which there are a greater number of papers that do not comply (N) with the quality criteria (29, 32 and 28 papers, respectively), when compared to the lower number of papers that do comply (Y) (41, 42 and 51 papers, respectively). Criterion 10 is particularly important when analyzing the reliability of the results, in addition to helping determine to what extent these results are biased by the subjective point of view of the researcher [42]. With regard to the use of one or several projects in the implementation and testing of the proposed models (criterion 13), this will be analyzed later in Section V-B3 as part of the answer to RQ3. Finally, the greatest number of papers with a partial assessment score (P) correspond with criterion 11, indicating that 32 papers did not respond appropriately to all the research questions. After completing this process, eight studies scored less than 6.5 and were discarded: seven from the 76 papers that resulted from phase 2, and one from the seven papers obtained from the snowballing phase after screening the full-text.

4) PHASE 4: SNOWBALLING

This phase consists of using the bibliographic references found in the papers to identify additional studies related to the objective of the search [43]. In the present study, this Backward Snowballing was performed after the quality assessment. The process was carried out recursively, that is, the references identified were again submitted to all the phases of the selection process and a snowballing of the remaining studies was, in turn, performed until no new references were identified. The first run of 69 primary studies enabled us to retrieve seven papers but one of them (from S33) did not attain the quality threshold. We, therefore, added six new studies [S10], [S13], [S15], [S27], [S35], [S38] and consequently obtained a total set of 75 studies, since these studies did not provide more in the second snowballing run. They were retrieved from S63, S70, S9, S57, S2 and S67, respectively. The final set of primary studies is listed in Appendix A.

Figure 1 shows a summary of the process followed to select the primary studies, along with the results obtained after each phase, as described in this section. At this point, however, we verified that none of the journal papers was an extension of a conference paper by the same authors. Two conference papers were eventually eliminated, leading to a final set of 73 primary studies.

TABLE 4. Quality assessment checklist (adopted from [20], [21]).

		Score		
#	Criteria	Y	N	P
1	Aims clearly specified?	76	0	7
2	Study designed to achieve these aims?	68	1	14
3	Estimation techniques used clearly described and their selection justified?	67	4	12
4	Variables considered by the study suitably measured?	67	1	15
5	Data collection methods adequately described?	65	4	14
6	Data collected adequately described?	58	5	20
7	Purpose of the data analysis clear?	66	3	14
8	Statistical techniques used to analyze data adequately described and their use justified?	59	11	13
9	Negative results (if any) presented?	41	29	13
10	Do the researchers discuss any problems with the validity/reliability of their results?	42	32	9
11	All research questions answered adequately?	45	6	32
12	How clear are the links between data, interpretation and conclusions?	56	3	24
13	Are the findings based on multiple projects?	51	28	4
Mean		**58.54**	**9.77**	**14.69**

V. ANALYSIS OF THE PRIMARY STUDIES

The analysis of the 73 primary studies is presented below in two sections. First, the bibliometric features of the final subset of studies are described, with a focus on the year of publication, the type of paper and the geographical distribution. The second section deals with the research questions stated in Section IV-A.

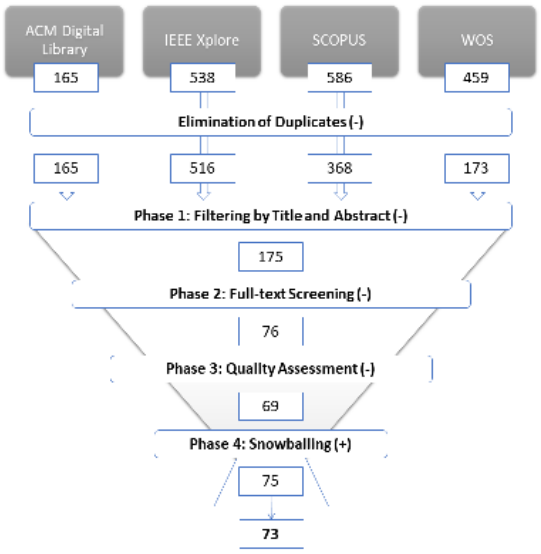

FIGURE 1. Primary study selection process.

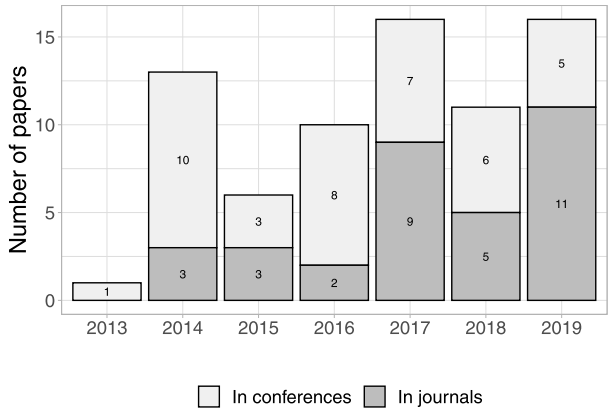

FIGURE 2. Number of papers published in journals and conferences per year.

A. BIBLIOMETRIC ANALYSIS

Figure 2 shows the distribution of the primary studies according to the type of document (journal papers or conference papers) and the year of publication. 40 out of the 73 primary studies were published in conferences, while the remaining 33 were published in journals.

The primary studies were found in a wide range of conferences and journals: as many as 29 conferences and 24 different journals. There were only four conferences at which more than one paper was presented: the Euromicro Conference on Software Engineering and Advanced Applications (SEAA), with four papers; the Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM/Mensura), with three papers, and the International Conference on Software Technologies (ICSOFT) and the International Conference on Software Engineering and Knowledge Engineering (SEKE), both with two papers. With regard to journals, there were also only four journals in which more than one paper was published: Journal of Systems and Software, with three papers, and another three (Journal of Software: Evolution and Process; International Journal of Intelligent Engineering and Systems, and International Journal of Advanced Computer Science and Applications), with two papers each.

With regard to the geographical distribution, the primary studies have been authored by researchers from a reduced set of countries, 30 in total, mainly in Asia and Europe. This analysis is restricted to the institutional affiliation of the first author. The country with the most publications is India (18), followed by Italy (6) and Germany (5). Other countries with more than two studies are Brazil, Pakistan and Turkey.

B. ANSWERS TO THE RESEARCH QUESTIONS

This section includes the results derived from the data extracted from the primary studies. These results allowed us to answer the research questions (RQ) proposed in this study. Data extraction was performed by the same team of three researchers and following the same procedure as that employed to select the primary studies. The percentages (shown in Tables 5 to 15) are calculated as the number of papers (column "#") divided by the total number of primary papers (73).

1) RQ1: WHAT METHODS HAVE BEEN USED TO ESTIMATE SIZE OR EFFORT IN ASD?

Table 5 shows the different methods used in the selected papers, as regards estimating both the effort and the size of a software project. Only three out of the 73 papers do not specify the method used to estimate software effort or size.

The estimation methods used in the primary studies were identified and categorized by following the classification proposed in [44], [45], in combination with the graphical representation proposed in [14].

Some of the estimation methods that rely on experts (Expert-based) are Planning Poker (18), Expert Judgment (8) and Wideband Delphi (4), which represents 34.25% of the total number of studies. ASD project managers prefer methods that facilitate collaboration and consensus for the estimation of the effort required and/or the size of the software to be developed. Planning Poker is, according to this SLR, the most widely studied method, and is usually used in combination with other methods such as Expert Judgment [S1], [S36], Wideband Delphi [S1], [S55], [S56], Machine Learning based estimation techniques [S21], [S26], [S31], [S43], Artificial Neural Network [S30], Functional Size Measurement [S30], [S38], [S48] and regression [S30], principally in order to make comparisons in relation to their accuracy.

Of the data-based estimation methods, those most frequently used are Machine Learning (Random Forest, Decision Tree, SVM, kNN, Stochastic Gradient Boosting, Naive Bayes and Random Forest) and Neural Networks (such as

TABLE 5. Estimation methods used.

Estimation method	Papers	#	%
Expert-based:			
Planning Poker	S1, S21, S26, S30, S31, S36, S38, S43, S45, S48, S50, S53, S55, S56, S57, S3 (Team Estimation Game), S59 (Average), S68 (Interaction Room Annotations)	18	24.66%
Expert Judgment	S1, S11, S16, S17, S36, S60, S62, S32 (Checklists)	8	10.96%
Wideband Delphi	S1, S55, S56, S64 (Program Evaluation Review Technique)	4	5.48%
Data-based:			
Machine Learning	S4 (Random Forest), S16 (Decision Tree), S19 (Decision Tree, SVM, kNN, Ensemble-based method), S21 (Stochastic Gradient Boosting), S24 (HKO method), S26, S29, S31 (Naive Bayes, J48, Random Forest, Logistic Model Tree), S37 (Text Classification), S41 (Decision Tree, Stochastic Gradient Boosting, Random Forest), S43 (SVM, kNN, Decision Tree, Naive Bayes), S44 (Document Fingerprints), S66, S72 (SVM), S67 (SVM optimized by Grid Search)	15	20.55%
Neural Network	S2 (Deep Belief Network, Deep Belief Network along with Antlion Optimization), S4 (Long Deep-Recurrent Neural Network), S12, S42 (General Regression Neural Network), S15 (Multilayer Perceptrons, ANN optimized by Fireworks Algorithm), S19, S30 (ANN), S34 (Evolutionary Cost-Sensitive Deep Belief Network), S35 (Feed-forward, Elman), S51 (Fuzzy Neural Network, kNN Regression, Independent Component Regression, ANN with a Principal Component Step, Multilayer Perceptrons), S52 (Feed-forward), S65, S71 (Multilayer Perceptrons)	13	17.81%
Functional Size Measurement	S7, S24, S30, S33, S38, S54, S69 (COSMIC), S11 (IFPUG), S20, S73 (FPA), S27, S48 (Simplified Function Points)	12	16.44%
Regression	S8 (Multiple Linear and Non-Linear Regression), S19 (Ridge Regression), S30 (Simple and Multiple Regression, Curve Fit.), S33 (Simple and Multiple Regression), S51 (Linear Regression), S62	6	8.22%
Algorithmic Methods	S13, S46 (Estimated Story Points), S20 (Estimation Process at Initial Stage), S40, S63 (Risk Aware and Quality Enriched Estimation Model)	5	6.85%
Fuzzy Logic	S5, S18, S23	3	4.11%
Swarm Intelligence	S9 (Particle Swarm Optimization - Artificial Bee Colony), S10 (Particle Swarm Optimization), S22 (Harmony Search Algorithm)	3	4.11%
Bayesian Network	S19, S25, S39	3	4.11%
Monte Carlo	S52	1	1.37%
Statistical Combination	S53	1	1.37%
Principal Component Analysis	S58	1	1.37%
COCOMO II	S60	1	1.37%
Combination-based:			
Use Case Point	S6, S70, S71 (UCP + Quick Function Point)	3	4.11%
Change Effort Prediction	S14, S60	2	2.74%
Ontology Model	S55, S56	2	2.74%
Experience Factory	S57	1	1.37%
Prioritization of Stories	S61	1	1.37%
Not reported	S28, S47, S49	3	4.11%

Deep Belief Network, Multilayer Perceptrons and Feed-forward), with 15 and 13 papers respectively.

We have identified 12 papers that use estimation methods based on Functional Size Measurement (FSM). These employ metrics such as COSMIC, FPA, IFPUG-FPA and Simplified Function Points (SiFP). As pointed out in the Original Study, the effort estimate is usually derived from the relative size of the project. These metrics will be discussed in detail in the subsection concerning RQ2.

Several regression techniques other than the aforementioned frequently used estimation methods have also been identified in six papers. Apart from regressions, algorithmic methods are used in five papers, as also shown in Table 5. It should be highlighted that this subset shares the following aspects:

- These estimation methods have mostly been proposed by the same authors [S13], [S46] or are referred to as having appeared in their previous work [S20], [S40], [S63].
- They all employ a common effort unit, called Estimated Story Points (ESP), with the exception of S20, which does not explicitly mention it.
- They define a process involving a series of steps or algorithms to determine the project effort.

Other less representative data-based methods include Fuzzy Logic, Swarm Intelligence, Bayesian Network, Monte Carlo, Statistical Combination, Principal Component Analysis and COCOMO II.

Some of the papers stand out owing to the fact that more than one method is used, thus making it possible to benchmark their results. For example:

- S31 and S43 investigate the use of classification algorithms [46] to determine a model with which to generate estimation values. S31 uses data obtained from user stories, whereas S43 focuses on estimating the effort required to solve system performance issues. These classification algorithms, such as Support Vector Machine (SVM), K-Nearest Neighbor (kNN), Naive Bayes, Decision Tree and Random Forest, are among the best known and most frequently used techniques in data mining [47] and Supervised Machine Learning [48].
- S41 and S42, which are by the same authors, use different methods for the assessment of an estimation model based on Story Points. S41 uses Machine Learning, whereas S42 evaluates different models based on Neural Networks.
- S12 compares three Machine Learning techniques: Adaptive Neuro-Fuzzy Modeling, Generalized Regression Neural Network and Radial Basis Function Networks. Story Points and Project Velocity are taken as inputs in the proposed approach.
- S19 proposes an ensemble-based model for the prediction of ASD effort and compares it with various individual predictive algorithms, such as Bayesian Networks, Ridge Regression, Neural Networks, SVM, Decision Trees and kNN.
- In S30, the authors applied different kinds of regression analyses and Artificial Neural Networks (ANN) to build the models. Simple, multiple, polynomial, power, exponential and logarithmic regressions were applied to Function Points, Story Points and effort data.
- S15 proposes a new method by which to ameliorate the accuracy of the agile software effort prediction process using the ANN optimized by Fireworks Algorithm. The performance of the Fireworks Algorithm is compared with other algorithms such as Directed Artificial Bee Colony, which is an improved version of the Artificial Bee Colony (ABC) algorithm, Teaching-learning Based Optimization and Teaching-learning Based ABC.
- S51 proposes the use of a Fuzzy Neural Network which is compared with models commonly used in the literature, such as kNN Regression, Independent Component Regression, ANN with a Principal Component Step and Multilayer Perceptrons.

The results of the comparison of these models in terms of accuracy will be discussed in Section V-B1.b, after introducing the different accuracy metrics.

Moreover, models based on data and expert opinion have been implemented in 12 of the studies [S11], [S16], [S21], [S26], [S30], [S31], [S38], [S43], [S48], [S53], [S60], [S62]. This evidences that there has been an interest in the combined use of these groups of methods, mainly to compare the accuracy and convenience of different estimation methods or to obtain better results. Indeed, 12.33% of the total number of papers were classified as combination-based effort estimation methods.

Figure 3 compares the results obtained in the two studies, considering only the main estimation methods identified in both cases. Planning Poker attains triple the amount of uses obtained in the Original Study, and the use of Neural Networks has also undergone a significant increase, whereas Expert Judgment, regression, algorithmic methods, COSMIC and Use Case Points (UCP) have decreased in relevance as regards their appearance in papers published in recent years. It is important to point out that, since the Original Study did not identify any papers based on Machine Learning, this category is not represented here.

Planning Poker was cited as the most frequently studied method (37.5%) by Dantas *et al.* [24]. These authors have also detected an increase in the use of intelligence techniques to support effort estimation (50%, compared to 27.5% presented in the Original Study). This trend has been reinforced throughout our study. In fact, 72.6% of the primary studies use models based on data, while 41 out of 73 studies (56.16%) use only data-based models.

a: RQ1a: WHAT METRICS HAVE BEEN USED TO DETERMINE THE ACCURACY OF EFFORT ESTIMATION METHODS IN ASD?

Estimation models cannot be evaluated without employing appropriate metrics to measure their accuracy. According to Table 6, the accuracy metrics most frequently used are those

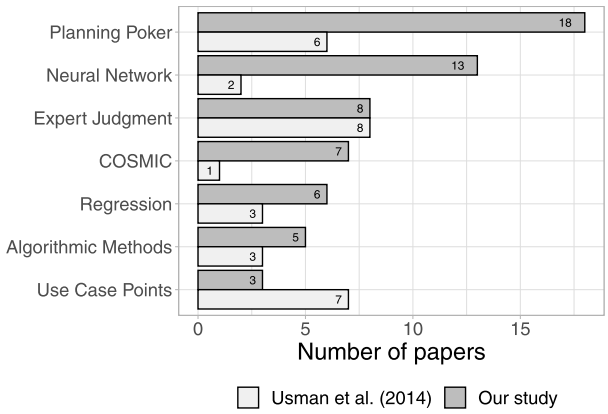

FIGURE 3. Estimation methods: Comparison of our results with those of Usman *et al.* (2014).

TABLE 6. Accuracy metrics used.

Accuracy metric	Papers	#	%
MRE	S1, S2, S5, S7, S8, S9, S10, S12, S14, S15, S16, S25, S26, S27, S30, S31, S33, S34, S35, S38, S42, S43, S44, S48, S55, S56, S57, S58, S59, S60, S63, S65, S66, S67, S71	35	47.95%
PRED(x)	S2, S5, S7, S8, S12, S14, S18, S21, S25, S26, S30, S33, S34, S35, S41, S55, S56, S63, S67	19	26.03%
R2	S8, S9, S15, S29, S38, S42, S62, S66	8	10.96%
% of accuracy	S24, S25, S34, S42, S43, S66, S72	7	9.59%
MAE	S4, S9, S19, S21, S25, S34, S72	7	9.59%
BRE	S3, S19, S24, S53, S63, S68	6	8.22%
MER	S12, S18, S21, S26, S41	5	6.85%
Other	S4 (Standardize Accuracy), S15 (MSE), S19 (RMSE), S25 (RMSE, RAE, RRSE), S32 (Mean(BREbias), Median(BREbias)), S34 (RMSE, MSE, RAE, RRSE), S35 (MSE), S42 (MSE), S51 (RMSE), S66 (MSE), S68 (Mean(BREbias), Median(BREbias)), S72 (Standardize Accuracy)	12	16.44%
Not reported	S6, S11, S13, S17, S20, S22, S23, S28, S36, S37, S39, S40, S45, S46, S47, S49, S50, S52, S54, S61, S64, S69, S70, S73	24	32.88%

that employ the Magnitude of Relative Error (MRE), either the mean or the median, with 35 papers out of 73. MRE is a basic unit-less value which is defined as:

$$MRE_i = \frac{|ActualEffort_i - PredictedEffort_i|}{ActualEffort_i}$$

MMRE (Mean MRE) is defined as the sample average of the MRE values over N projects. One of the disadvantages of the MMRE is that it is sensitive to outliers. MdMRE (Median MRE) has been used as another criterion because it is less sensitive to outliers. MdMRE (Median MRE) is specifically used in S9, S27, S41, S48 and S67.

The prediction level PRED(x) calculates the ratio of MRE values that fall into the selected range (x) out of the total of the N projects. PRED(x) is usually used as a complimentary criterion to MMRE, which is the case of most of the 19 papers, with the exception of S18, S21 and S41. As Shepperd and MacDonell state in [49], certain common measures for accuracy estimation, such as MMRE or PRED(x) are not the most appropriate. Measures based on residuals should, therefore, also be used.

The squared correlation coefficient (R2), denominated as the coefficient of determination, is used in eight papers. S15, S42 and S66 also used the Mean Square Error (MSE) along with R2.

Another criterion employed to evaluate and compare the accuracy of estimation models is the % of accuracy, defined as the number of correct predictions made divided by the total number of predictions made. Seven papers adopted this accuracy metric along, with others.

Mean Absolute Error (MAE) is the average number of absolute values of prediction errors. MAE was also used in seven papers too. S4 and S72 also used Standardized Accuracy (SA) which is based on MAE, while S19, S25 and S34 used the Root Mean Squared Error (RMSE). MAE and RMSE are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|ActualEffort_i - PredictedEffort_i|$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(ActualEffort_i - PredictedEffort_i)^2}{N}}$$

Both MAE and RMSE are useful for the comparison of the prediction errors of different models for a particular variable. They show errors in the same unit and scale as the variable itself. However, Relative Absolute Error (RAE) and Root Relative Squared Error (RRSE) can be used to compare models whose errors are measured in different units. S25 and S34 also used RAE and RRSE.

Another six papers that use Balanced Relative Error (BRE) have also been identified. BRE is a balanced measure that evenly balances overestimation and underestimation, and is defined as:

$$BRE_i = \frac{|ActualEffort_i - PredictedEffort_i|}{min(ActualEffort_i, PredictedEffort_i)}$$

S19 and S24 used mean(BRE), S68 used median(BRE), while S3, S53 and S63 used both. S68 also used BREbias, which measures not only the size of the estimation error, but also its direction.

An alternative criterion used to evaluate the accuracy of estimation models is the Magnitude of Error Relative (MER) to the estimate, which can provide higher accuracy than the

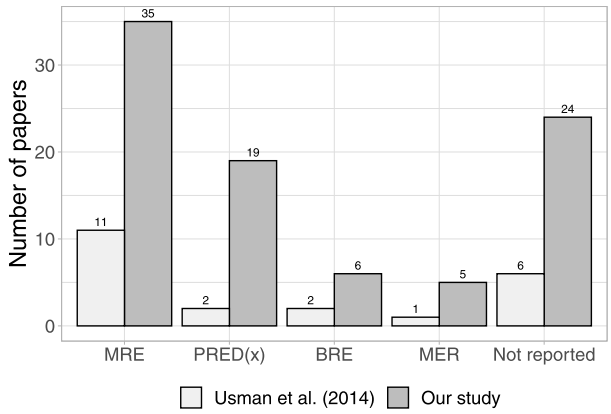

FIGURE 4. Accuracy metrics: Comparison of our results with those of Usman *et al.* (2014).

MRE. The mean of MER was used in five papers, one of which [S41] also used the median of MER. MER is defined as:

$$MER_i = \frac{|ActualEffort_i - PredictedEffort_i|}{PredictedEffort_i}$$

Finally almost 33% (24 papers) did not report the accuracy metrics used to validate the models. Surprisingly, all the papers previously classified in the category of algorithmic methods but one [S63] can be found here.

Figure 4 relates the results obtained in the two studies, considering only the main accuracy metrics identified in both cases. Accuracy metrics based on MRE (MMRE and MdMRE) are those most frequently used, and are present in 47.95% of the primary studies. This is consistent with the results of the Original Study and was also observed by Dantas *et al.* [24]. PRED(x) is the second most used accuracy metric (26.03%). Some studies implementing BRE (8.22%) have also been detected (probably owing to the criticism regarding the lack of balance of the metrics based on MRE), as well as studies implementing metrics based on MER (6.85%). In the Original Study, MdMRE was used in only one study, while balanced metrics such as BRE and BREbias were also used in two studies. This also occurs in [24].

b: RQ1b: WHAT IS THE LEVEL OF ACCURACY OF EFFORT ESTIMATION METHODS IN ASD?

Table 7 shows the accuracy values of each accuracy metric for those estimations methods that were reported in at least four of the papers in Table 5. Note that, in addition to "% of accuracy", the following metrics are also expressed as a percentage: PRED(x) and MAE. That typically employed is PRED(25%), but some papers [S30], [S33] also studied PRED(30%) and S7 reported the use of PRED(20%). However, S34 reported the use of PRED(10%) in addition to PRED(25%) and S41 also reported the use of PRED(50%), PRED(75%) and PRED(100%). These values were, of course, not included in this table.

However, PRED(25%) >= 75% is generally considered an acceptable model accuracy [29]. This is evident in Table 7 with the exception of six papers:

- In S26, BlackSheep learns from other user stories as soon as they are closed in the project, reducing its dependency on historical data. This Machine Learning proposed method improves on Planning Poker by a significant margin, from PRED(25%) = 73% to 81% when the accuracy metrics are computed throughout the entire development process of the projects. However, Planning Poker achieved only PRED(25%) = 33% at the beginning of the development time frame.
- S41 reported PRED(25%) = 38.10% for a Decision Tree model (and PRED(25%) = 66.67% for a Random Forest model), while the Stochastic Gradient Boosting effort estimation model outperformed the other two Machine Learning models.
- S21 proposed a hybrid model that incorporates expert knowledge and change impact analysis information in order to improve the effectiveness of effort estimates. The results indicate that HyEEASe is useful as regards supporting experts when making estimations, PRED(25%) = 75%. It performs better than purely expert-based judgment (Planning Poker). Moreover, a purely data-based model (Gradient Boosted Tree) did not perform better than an expert-based estimation supported by HyEEASe. The authors reported PRED(25%) = 50% in both cases.
- In S56, the proposed ontology approach provided an estimation accuracy of PRED(25) = 85%, while the Planning Poker and Delphi methods attained accuracies of 75% and 55%.
- S12 implemented three Machine Learning techniques to evaluate software effort in terms of cost. The authors reported PRED(25%) = 57.14% for Adaptive Neuro-Fuzzy Modeling, while PRED(25%) = 76.19% was attained for both the Generalized Regression Neural Network and the Radial Basis Function Network.
- In S30, PRED(30%) = 66.67% when the power exponential regression model was applied to Story Points, while when applied to COSMIC FP, PRED(30%) = 77.78%.

However, in most of the papers, these low PRED(x) values are reported for baseline models used for comparison purposes.

Conte *et al.* [28] considered MMRE <= 0.25 to be an acceptable level of performance for effort prediction models. Most of the papers in Table 7 also presented acceptable values of MMRE, but four papers reported excessively high values:

- S31 reported poor results (greater than 0.92) for all the Machine Learning models (Naive Bayes, J48, Random Forest, Logistic Model Tree).
- S44 proposed a user Story Points estimation tool that employs the document fingerprints technique. The

TABLE 7. Accuracy values of the estimation methods used.

Estimation method	Accuracy metric	Accuracy values	Papers
Expert-based:			
Planning Poker	MRE	0.03, 0.09	S59
	MMRE	0.07, 0.12, 0.13, 0.18, 0.29, 0.39, 0.41, 0.52, 0.58, 0.7, 1.07	S1, S26, S31, S38, S48, S55, S56, S57
	MdMRE	0.25, 0.58	S48
	PRED(x)	33, 50, 73, 75, 83	S21, S26, S55, S56
	R2	0.33	S38
	MAE	0.72	S21
	Mean(BRE)	0.03, 0.37, 0.45, 0.65, 0.82, 0.96	S3, S53, S68
	Median(BRE)	0.01, 0.2, 0.28, 0.42, 0.5	S3, S53, S68
	MMER	0.26, 0.47, 1.55	S21, S26
Expert Judgment	MMRE	0.12, 0.15, 0.29, 0.33	S1, S16, S60
Wideband Delphi	MMRE	0.08, 0.15, 0.23	S1, S55, S56
	PRED(x)	55, 75	S55, S56
Data-based:			
Machine Learning	MMRE	0.06, 0.07, 0.13, 0.15, 0.16, 0.19, 0.43, 0.5, 0.84, 0.92, 0.98, 1.13, 2.04	S16, S26, S31, S43, S44, S66, S67
	MdMRE	0.04, 0.09	S67
	PRED(x)	38.1, 50, 78, 80.95, 81, 85.71, 100	S21, S26, S41, S67
	R2	0.37, 0.45, 0.51, 0.76, 0.87, 0.98	S29, S66
	% of accuracy	23, 35.8, 38.4, 59, 61.5, 68.74, 73.1, 95.91	S24, S43, S66, S72
	MAE	0.88, 1.7, 1.9, 2.61, 8	S4, S19, S21, S72
	Mean(BRE)	0.03, 0.35, 0.67	S19, S24
	MMER	0.13, 0.16, 0.17, 0.38, 1.03	S21, S26, S41
	MdMER	0.12, 0.29	S41
Neural Network	MMRE	0.03, 0.04, 0.05, 0.06, 0.08, 0.1, 0.12, 0.13, 0.14, 0.15, 0.17, 0.35, 1.58	S2, S12, S15, S30, S34, S35, S42, S65, S71
	MdMRE	0.05, 0.06, 0.12, 0.13, 0.16	S2
	PRED(x)	57.14, 76.19, 91, 92, 94.87, 95.23, 96, 97, 100	S2, S12, S30, S34, S35
	R2	0.62, 0.91, 0.93, 0.99	S15, S42
	% of accuracy	85.92, 94.76, 99.49	S34, S42
	MAE	0.02, 2.09, 13.21	S4, S19, S34
	Mean(BRE)	0.05	S19
	MMER	0.04, 0.06, 0.16	S12
FSM	MMRE	0.06, 0.13, 0.22, 0.28, 0.3, 0.66, 1.2	S7, S27, S33, S38, S48
	MdMRE	0.09, 0.66, 0.76	S27, S48
	PRED(x)	78, 80.63, 100	S7, S33
	R2	0.78	S38
	% of accuracy	50.8, 61.3	S24
	Mean(BRE)	0.32, 0.7	S24
Regression	MMRE	0.04, 0.06, 0.1, 0.22, 0.23	S8, S30, S33
	PRED(x)	66.67, 78, 100	S8, S30, S33
	R2	0.97, 0.98	S8, S62
	MAE	9.20	S19
	Mean(BRE)	0.02	S19
Algorithmic Methods	MMRE	0.00	S63
	PRED(x)	93.51	S63
	Mean(BRE)	0.03	S63
	Median(BRE)	0.02	S63

authors concluded that their tool helps reduce the inaccurate estimation often attained by people who may be unfamiliar or inexperienced with the project, but reported MMRE values greater than 0.84.

- In S48, the authors concluded that SiFP and IFPUG Function Points had low predictive power and did not help to improve the accuracy of export-based estimations of Scrum Planning Poker. They reported

TABLE 8. Accuracy summary statistics.

Estimation method	Accuracy metric	Number of papers	Mean	Median	Range
Expert-based:					
Planning Poker	MMRE	8	0.41	0.39	0.07, 1.07
	PRED(x)	4	62.80	73.00	33.00, 83.00
Expert Judgment	MMRE	3	0.22	0.22	0.12, 0.33
Data-based:					
Machine Learning	MMRE	7	0.58	0.43	0.06, 2.04
	PRED(x)	4	73.39	80.95	38.10, 100.00
Neural Network	MMRE	9	0.23	0.12	0.03, 1.58
	PRED(x)	5	88.83	94.87	57.14, 100.00
Functional Size Measurement	MMRE	5	0.41	0.28	0.06, 1.20
	PRED(x)	2	86.21	80.63	78.00, 100.00

MMRE values higher than 0.66 in the case of SiFP.

- S42 obtained an MMRE value of 1.58 for a Probabilistic Neural Network, while in the case of the other three Neural Networks (General Regression, GMDH Polynomial and Cascade Correlation) MMRE was always lower than 0.36.

Some interesting conclusions can be attained from those papers that compared different estimation methods. Gandomani *et al.* [S1] showed that both Wideband Delphi and Planning Poker helped companies estimate the cost of the projects more accurately than experts. They also attained less accuracy in S59 when the team obtained an average size of the User Stories when compared to coming to a consensus about the size, i.e., Planning Poker. With regard to their expert judgment proposal supported by checklists, Usman *et al.* [S32] showed that checklist estimates are more accurate and have considerably less underestimation bias. Pozenel and Hovelja [S3] showed that Team Estimation Game provides more accurate story estimates than Planning Poker and improves estimation accuracy from sprint to sprint. Unfortunately, they were not able to confirm that this estimation technique is less time consuming than Planning Poker. Lenarduzzi *et al.* [S48] showed that expert-based effort estimation was much better than estimation predicted by means of FSM. They confirmed that SiFP and IFPUG Function Points did not help to improve estimation accuracy in Scrum. Furthermore, Moharreri *et al.* [S31] demonstrated that J48 Decision Tree (with or without Planning Poker estimates) and the Logistic Model Tree with Planning Poker estimates, all performed better than manual Planning Poker. Furthermore, Commeyne *et al.* [S38] argued that the COSMIC measurement method provides objective evidence of better estimates although Planning Poker and Story Points are widely recognized and used in the Agile community. Ungan *et al.* [S30] observed that Story Points based estimation models performed slightly better than COSMIC based models in common analysis methods. However, significantly better results were attained with multiple regression and ANN, although only COSMIC FP was eligible to be used in such methods. In this respect, Salmanoglu *et al.* [S33] confirmed that regression models that use COSMIC FP as an independent variable provide successful estimates. Finally, Raslan and Darwish [S18] were of the opinion that the utilization of Story Points with COCOMO II factors may provide realistic effort in the constructive iteration phase. Moreover, the use of Fuzzy Logic in the proposed model increases the accuracy.

In order to be able to attain to some insights into that stated above, Table 8 shows the relevant information from Table 7. On the one hand, it includes only the accuracy values for the estimations methods most frequently studied according to this SLR, i.e., Planning Poker, Expert Judgment, Machine Learning, Neural Network and Functional Size Measurement. On the other hand, as the papers use different accuracy metrics, which makes it difficult to compare between the estimation models, only MMRE and PRED(x) have been considered. This table shows the mean, median and range of the accuracy values reported in the papers. Taking into account the mean of the PRED(x) values reported in the papers, data-based estimation methods provide better values than Planning Poker. Futhermore, Neural Networks outperform both Machine Learning and FSM techniques. This is confirmed by the mean of the MMRE values. However, there are not a sufficiently high number to be able to draw further conclusions.

With regard to the accuracy values, the authors of the Original Study found only one paper that reported acceptable MRE values for the UCP Method and Expert Judgment, and no other method achieved accuracy levels of at least 0.25. Dantas *et al.* [24] showed only the works with the best levels of accuracy by estimation method. They reported the best results for Planning Poker and Expert Judgment, in addition to those for what they denominated as intelligent techniques: Machine Learning, Bayesian Networks and Optimization Algorithms. The papers that used intelligent techniques presented better results. A clear gap in ASD cited by Usman *et al.* [21] and confirmed in [24] is the lack of studies measuring prediction accuracy and presenting acceptable accuracy.

Although Usman *et al.* [21] concluded that the lack of studies measuring prediction accuracy and presenting acceptable accuracy represents a clear gap in this field, in our study we have identified improvements in this respect. That being said, many papers continue to report inadequate accuracy results.

On the one hand, 12 out of 49 papers (24.49%) reported aspects concerning the validation of the models, even if little information could be retrieved from them. In S7, five finished projects were reserved out of the 25 projects available and the results were validated by using multiple regression models (Constrained Minimum Sum of Squared Error (CMSE), Constrained Minimum Sum of Absolute Error (CMAE) and Constrained Minimum Sum of Relative Error (CMRE)). k-fold cross-validation was used in most of the papers. Since 10-fold cross-validation is commonly used, six papers were identified here (S16 used four agile projects based on Scrum, S24 used six projects with 93 use cases, S25 used a database of 160 tasks, S41 used 21 projects [50], S43 used a total of nine projects, and S72 used eight open source projects, one from JIRA). However, S66 used 5-fold cross-validation, in which five projects were taken for testing out of a data set of 21 projects [50]. In addition to 10-fold cross-validation, S41 also used leave-one-out validation, and S42 that used 21 projects [50] but the leave-one-out validation was applied to a testing dataset of six projects. S2 and S67 also applied leave-one-out validation to the same dataset [50]. Finally, the predictive models in S19 were trained using the blocked cross-validation technique.

Furthermore, six papers out of 49, i.e., 12.24%, reported not only the statistical significance of the different accuracy values when comparing between models, but also the practical significance. To be precise, S4 and S19 used the Vargha and Delaney effect size, S3, S53 and S68 used Cliff's Delta, while S41 used Cohen's d and Glass's Delta. The effect size was reported to be small in S53, S68 and S41. In S3, the effect size measure was of medium size for only the third sprint. In the two papers that used the Vargha and Delaney statistic, the authors concluded that the proposed models outperformed the baselines with effect sizes different from 0.5. However, they did not report any interpretation for the magnitude of the effect sizes, by suggesting whether it was a small, a medium or large effect.

2) RQ2: WHAT EFFORT PREDICTORS HAVE BEEN USED FOR EFFORT ESTIMATION IN ASD?

This research question is related to the effort predictors used for effort estimation in ASD. The main effort predictors are size metrics and cost factors. These effort predictors identified in the 73 primary studies are shown as follows.

a: SIZE METRICS

Table 9 shows that the metrics most frequently used to estimate the size of the software are Story Points. 61.64% of the papers reported having used Story Points, in some cases in combination with other metrics such as COSMIC [S30], [S33], [S38], [S69], LOC [S18], [S26], [S33], FPA [S20], NESMA-FPA [S11], Quick FP [S71] and UCP [S71].

TABLE 9. Size metrics used.

Size metric	Papers	#	%
Story Points	S2, S3, S4, S5, S9, S11, S12, S13, S15, S16, S17, S18, S19, S20, S22, S23, S26, S28, S29, S30, S31, S33, S35, S36, S37, S38, S40, S41, S42, S43, S45, S46, S47, S50, S53, S57, S59, S61, S63, S66, S67, S68, S69, S71, S72	45	61.64%
Function Points:			
COSMIC	S24, S30, S33, S38, S49, S54, S56, S58, S62, S69	10	13.7%
FPA	S20, S73	2	2.74%
SiFP	S27, S48	2	2.74%
NESMA	S11	1	1.37%
IFPUG	S48	1	1.37%
Quick FP	S71	1	1.37%
Other:			
LOC	S6, S14, S18, S21, S26, S33, S52	7	9.59%
UCP	S6, S51, S70, S71	4	5.48%
XU	S7	1	1.37%
Not reported	S1, S8, S10, S25, S32, S34, S39, S44, S55, S60, S64, S65	12	16.44%

Function points occupy the second place, with a total of 16 papers, since S48 uses both SiFP and IFPUG Function Points. The main functional size measurement standards, such as COSMIC, IFPUG-FPA and NESMA-FPA are included here. S20 and S73 use Function Point Analysis (FPA) for size estimation, although the authors do not specify the use of any of the standard metrics. The following metrics are also reported:

- SiFP, Simplified Function Points [S27], [S48], as a simplified version of IFPUG-FPA [51], and
- Quick Function Points [S71], also based on IFPUG-FPA, are used to measure the software size during the early stages of the development process [52].

S7 uses eXtreme software size Units (XU) to determine the size of software projects developed using Xtreme Programming practices. The metric has been defined as a modified version of COSMIC Full Function Points adapted to this type of projects. The XU value of a user history depends mostly on the number of fields in the user interface and the logical tables created for this history, along with the number of variables to be handled.

These results are compared to those obtained in the Original Study in Figure 5, in which we decided to present COSMIC separately from the other functional size measures in order to enable a proper comparison. On the one hand, the most significant contrast concerns the number of papers that use Story Points (45 papers representing 61.64% of the total) with respect to the smaller number of papers in the Original Study (six papers representing 24%). On the other,

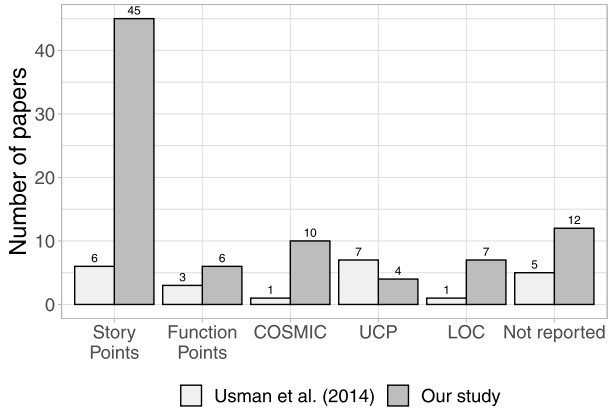

FIGURE 5. Size metrics: Comparison of our results with those of Usman *et al.* (2014).

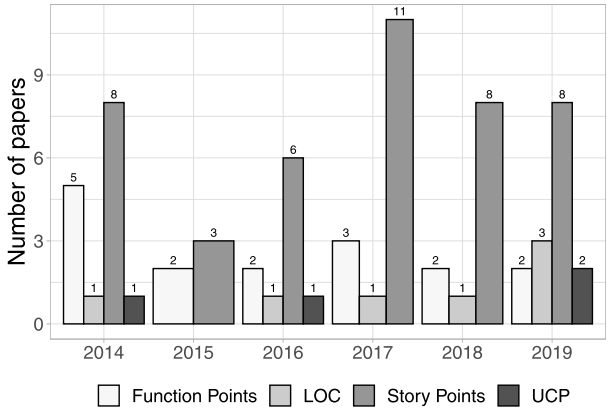

FIGURE 6. Size metrics used in the papers per year of publication.

only four papers now report the use of Use Case Points (UCP). UCP, which appeared among the main size metrics in the Original Study (seven papers), now appears among the least used metrics, since few studies rely on use cases as a means to specify functional requirements. Indeed, Dantas *et al.* [24] found no records of the use of UCP. There is a clear relationship between the main estimation method (Planning Poker), the most frequently used size metric (Story Points) and the way in which development requirements are specified in ASD (user story). This result is also consistent with that of Dantas *et al.* [24], since 70.8% of the papers used Story Points. Finally, 16.44% of the total number of papers did not report the size metric used to measure the size of the software, and this percentage was slightly higher (20%) in the Original Study.

Figure 6 shows the relationship between the main size metrics used in the papers and the year of publication. The use of Story Points in the estimation models evaluated is higher when compared to that of the other metrics in all years of publication, with a considerable number of papers in 2014, 2016, 2017, 2018 and 2019. The Function Points metric is also present in five papers published in 2014, whereas UCP and LOC are still the least used metrics in all years except 2019. All four size metrics (SP, FP, UCP and LOC) are present in papers published in 2014, 2016 and 2019.

Another important aspect about size metrics is their relationships with the estimation methods used. Most studies that use Planning Poker or Expert Judgment for effort estimation reported having used Story Points as the size metric, apart from S1, S32, S55 and S60 which did not explicitly mention it, S56 and S62, which used COSMIC, S48, which used IFPUG Function Points and SiFP, and S21, which used LOC. All the algorithmic methods used Story Points.

b: COST FACTORS

Since a wide range of cost factors were identified, we decided to analyze the data extracted so as to answer this research question using a qualitative analysis method, namely thematic analysis [53]. The flexibility of thematic analysis allows for rich, detailed and complex descriptions of the available data, and this method, therefore, helped us identify, analyze, and report patterns within the data. We followed the five steps of the thematic analysis method [53] as detailed below:

1. Becoming familiar with the data: we examined the data items extracted, i.e., the effort predictors (cost factors) in order to form the initial ideas for analysis.
2. Generating initial codes: in the second step, we extracted the initial lists of cost factors. It should be noted that in some cases, we had to recheck the papers.
3. Searching for themes: for each data item, we attempted to combine different initial codes generated from the second step into potential themes.
4. Reviewing and refining themes: the cost factors identified from the third step were checked against each other in order to understand what themes had to be merged with others or dropped (e.g., lack of sufficient evidence).
5. Defining and naming themes: this step enabled us to define clear and concise names for cost factors, along with their related information. Table 10 shows the final results obtained for each theme.

The aforementioned thematic analysis allowed us to classify the cost factors into four categories: project, team, technical and user story factors.

Project factors include aspects such as the complexity or innovative nature of the project, quality, risk taking and the clarity of the requirements. The factors related to the project and team appear more frequently than the others mentioned in the primary studies. The most frequently used cost factor is the complexity of the project (27.4%), followed by the team's experience (23.29%). This coincides with the results of the Original Study, in which the development team's skills and experience are highlighted as playing an important role in the estimation process. Other team factors identified in our study are: familiarity within team/project, team size and velocity, communication, working hours/days and developers' availability.

TABLE 10. Cost factors used.

Cost factors	Papers	#	%
Project factors:			
Complexity	S2, S3, S5, S18, S21, S22, S25, S28, S36, S37, S39, S40, S46, S47, S51, S58, S63, S64, S70, S73	20	27.4%
Risk taking	S23, S40, S56, S58, S63	5	6.85%
Clarity of requirements	S9, S46, S54	3	4.11%
Novelty	S23, S25, S63	3	4.11%
Quality	S25, S39, S63	3	4.11%
Team factors:			
Team experience	S5, S9, S16, S18, S19, S22, S28, S39, S40, S52, S55, S59, S64, S69, S70, S72, S73	17	23.29%
Developers skills	S7, S16, S18, S25, S39, S40, S72	7	9.59%
Familiarity within team/project	S9, S40, S46, S47, S55, S70	6	8.22%
Team velocity	S12, S15, S18, S35, S67	5	6.85%
Communication	S39, S40, S46, S59, S72	5	6.85%
Team size	S9, S46, S58	3	4.11%
Working hours/days	S10, S25, S72	3	4.11%
Developers availability	S52, S72	2	2.74%
Technical factors:			
Software and development tools	S6, S46, S58	3	4.11%
Impact of existing systems	S27, S48	2	2.74%
User stories factors:			
Priority of the story	S19, S26, S31, S45, S49, S55, S63	7	9.59%
Sprint of the story	S19, S26	2	2.74%
Story development type	S27, S48	2	2.74%
Other:			
Textual information	S4, S24, S31, S43, S72	5	6.85%
Popli and Chauhan's factors	S13, S20, S46	3	4.11%
COCOMO II cost drivers	S14, S60	2	2.74%
Process maturity	S58	1	1.37%
Not reported	S1, S8, S11, S17, S29, S30, S32, S33, S34, S38, S41, S42, S44, S50, S53, S57, S61, S62, S65, S66, S68, S71	22	30.14%

Other relevant but less representative factors in the primary studies are technical. Any software or tool used in the existing systems may impact on the development effort required. Examples of technical factors are application type, database used, development platform, operation system and programming language, as occurs in traditional software development.

Finally, the factors related to the user stories comprise the fourth category, given their specific relationship with this key element in the project: priority of the story in relation to the business value, sprint in which a story is developed, and story development type. The user stories can be classified as a new feature (user stories that involve the creation of a new feature) and maintenance (bug fixing or requirement changes for an existing feature), according to the type of development.

In summary, Table 10 shows that there is no common pattern as regards the use of general and specific cost factors, as already pointed out in the Original Study. This is owing to the fact that cost factors depend mainly on the features of the projects analyzed when proposing an effort estimation model.

Only two papers explicitly mentioned the use of the COCOMO model, whereas three papers considered a specific set of cost factors referred to as people and project-related factors, which have been included in the "Others" category as *Popli's and Chauhan's factors*, owing to the names of their authors.

Furthermore, some papers have studied factors related to the textual information, such as the use of specific terms in the description of the software requirements or the frequency of terms. These factors are introduced by means of estimation based on the intensive use of data such as Machine Learning [S24], [S31], [S43], [S72] and Neural Networks [S4]. For example, S24 presents an adaptation of the HKO (Hussain, Kosseim and Ormandjieva) method [54], which is intended for the classification of informally written requirements in relation to the COSMIC functional size, by employing use cases and Natural Language Processing tools (NLP).

Despite this classification, it is possible to find intersections between the categories. For example, communication is a cost factor that depends on both the people and the environment in which the project has been developed. User stories are similarly a specific artifact of the project, particularly in agile methods, and could therefore, also be included in this category. Finally, the cost factors could not be identified

TABLE 11. Domains of the dataset used.

Domain of the dataset	Papers	#	%
Industrial	S1, S2, S4, S5, S7, S9, S11, S12, S14, S15, S16, S17, S18, S23, S26, S28, S29, S30, S32, S33, S34, S35, S36, S37, S38, S41, S42, S43, S44, S45, S47, S50, S51, S53, S54, S55, S56, S57, S58, S59, S60, S62, S64, S65, S66, S67, S69, S71, S72, S73	50	68.49%
Academic	S3, S19, S22, S25, S27, S68	6	8.22%
Industrial and Academic	S24	1	1.37%
Simulated	S21, S52, S63	3	4.11%
Not reported	S6, S8, S10, S13, S20, S31, S39, S40, S46, S48, S49, S61, S70	13	17.81%

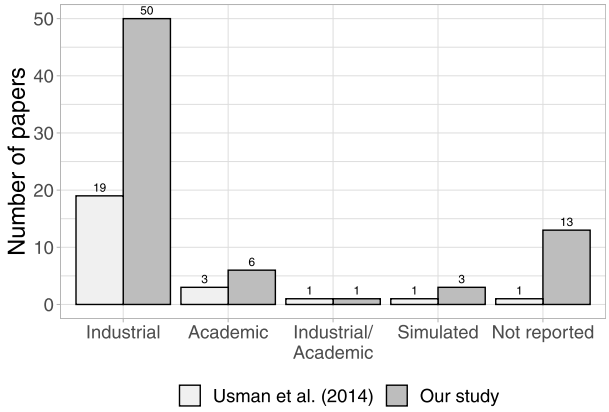

FIGURE 7. Domain of the dataset: Comparison of our results with those of Usman *et al.* (2014).

in 30.14% of the papers. This is coherent with the results of the Original Study, in which 32% of the papers did not report the cost drivers used. Despite the fact that Dantas *et al.* [24] presented a lower value of 25%, the research gap and the lack of clarity as regards cost factors still remain and justify the need for more studies.

3) RQ3: WHAT ARE THE CHARACTERISTICS OF THE DATASET USED FOR SIZE OR EFFORT ESTIMATION IN ASD?

In this research question, the characteristics of the dataset used in each paper are assessed. This aspect is relevant when considering the reliability and applicability of the models proposed in the papers. Basically, the idea is to identify whether the models have been validated with one or multiple cases, real or simulated, for academic purposes or with an industrial application, and deal with internal or cross-company data. Those aspects related principally to the domain and the type of dataset are described as follows.

a: DOMAIN OF THE DATASET

Table 11 shows that most of the papers have used data from the software development projects of specific companies. On the one hand, 50 papers have used industrial domain datasets, whereas six papers rely on data from the projects of educational institutions. The only paper to employ a mixed repository, made up of five projects implemented at the Poznan University of Technology and a project carried out by a software development company, was S24. On the other hand, only three papers explicitly indicated the use of data extracted from a simulation [S21], [S52], [S63], which correspond to estimation models based on the intensive use of data.

13 papers did not report the domain of the dataset used, although there was some sort of validation of the proposal in all of them except one [S39]. S31, for example, employed data collected through the use of iteration logs stored for each team using IBM Rational Team Concert, while the authors of S40 used hypothetical values. Some papers even worked with a considerable number of projects. This is the case of S8, S10 and S49 with 40, 10 and 15 projects, respectively.

Figure 7 shows the comparison of the results obtained herein with those of the Original Study. The main difference appears as regards those papers that did not report the domain of the dataset used - 13 papers (almost 18%) in the present study, as compared to only one (4%) in the original. Furthermore, industrial domain datasets are clearly in the first position in both studies. With regard to the domain of the dataset, in the present study the relative proportions of papers that have used academic, simulated or mixed (industrial/academic) datasets have decreased.

b: TYPE OF DATASET

Table 12 shows the type (within-company or cross-company) of datasets used in the 73 primary studies. The categories "Simulated" and "Not reported" contain the same papers as those shown in Tables 11 and 12.

46.58% of the papers present estimation models evaluated using data from the internal projects of one company (within-company). Only S18 reported the use of a sample of projects, which was extracted from the COCOMO NASA dataset. Note also that all papers that use an academic dataset include internal data. 23 studies (31.51%) indicate the use of datasets with projects from different organizations (cross-company). Nine out of these 23 papers used the same dataset of 21 software projects [50] developed by six different companies: S41, S42 and S66, which share authorship, S9 and S15, which also share authorship, and S2, S12, S35 and S67. Six papers selected projects stored in open source repositories that use the JIRA issue tracking system [S4], [S17], [S37], [S43], [S44], [S72]. Two other papers by the same authors [S14 and S60] gathered together software projects in the industry from the CSBSG[3] database. The remaining papers made use of projects from two or three companies at most,

[3]Chinese Software Benchmarking Standard Group

TABLE 12. Type of dataset used.

Type of dataset	Papers	#	%
Within company	S1, S3, S5, S7, S11, S16, S18, S19, S22, S24, S25, S26, S27, S28, S29, S30, S33, S34, S36, S38, S45, S47, S50, S53, S55, S56, S57, S59, S62, S64, S65, S68, S71, S73	34	46.58%
Cross-company	S2, S4, S9, S12, S14, S15, S17, S23, S32, S35, S37, S41, S42, S43, S44, S51, S54, S58, S60, S66, S67, S69, S72	23	31.51%
Simulated	S21, S52, S63	3	4.11%
Not reported	S6, S8, S10, S13, S20, S31, S39, S40, S46, S48, S49, S61, S70	13	17.81%

with the exception of one [S58], which reported the use of ASD projects derived from software companies all over the world.

Figure 8, meanwhile, shows a comparison of our results with those obtained in the Original Study. The use of within-company data is dominant in both studies, although in the present study, the number of papers corresponding to cross-company datasets is more significant, representing almost 32% of the total in comparison to 8% in the Original Study. According to Dantas *et al.* [24], industry also remains the most reported domain (58.33%) and only one paper used cross-company data, while all the others used within-company data. When compared to the papers found in the Original Study, there was a considerable growth in the number of works describing validation in the academic domain (i.e., 29.17% vs. 12%). The results of the Original Study and [24], therefore, suggest that, within the scope of ASD, companies have focused on their own project data, rather than seeking data from cross-company datasets. Usman *et al.* [21] believed that some effort should be made to make cross-company datasets available for the ASD context. Finally, note that the number of papers using simulation data is insignificant (the Original Study found one, we identified three), and that this category does not appear in [24].

4) RQ4: WHAT AGILE METHODS IN ASD HAVE BEEN INVESTIGATED FOR SIZE OR EFFORT ESTIMATION?

This question relates to one of the fundamental aspects of this research since it places the object being studied (i.e., Effort Estimation) in a specific context (i.e., Agile Software Development). This means that all the effort estimation models analyzed here meet the condition of being implemented in ASD environments (see the selection process in Section IV-D). Table 13 shows the agile methods implemented in the primary studies.

Some of the best-known agile methods, i.e, Scrum, Xtreme Programming (XP) and Test Driven Development (TDD), are used in this set of studies. Some of the other agile methods implemented are:

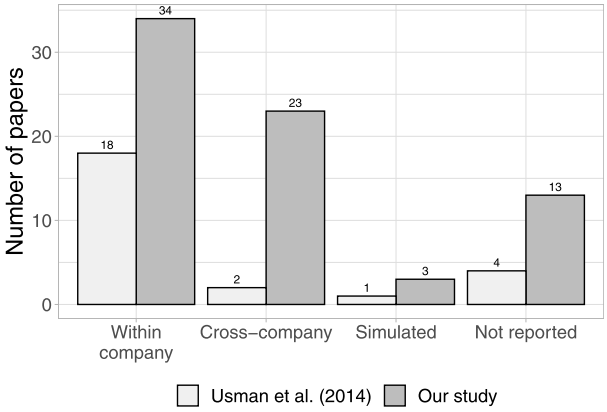

FIGURE 8. Type of dataset: Comparison of our results with those of Usman *et al.* (2014).

TABLE 13. Agile methods researched.

Agile method	Papers	#	%
Scrum	S1, S2, S3, S5, S6, S9, S11, S12, S14, S15, S16, S19, S20, S21, S23, S25, S27, S28, S29, S30, S32, S33, S34, S35, S36, S38, S40, S41, S42, S44, S45, S47, S48, S49, S50, S52, S53, S54, S55, S56, S57, S59, S60, S62, S65, S66, S67, S68, S73	49	67.12%
XP	S7, S14, S43, S56, S59, S60	6	8.22%
Agile Unified Process	S14, S60	2	2.74%
Kanban	S54, S62	2	2.74%
TDD	S54, S62	2	2.74%
Distributed Agile Software Development	S63	1	1.37%
Not reported	S4, S8, S10, S13, S17, S18, S22, S24, S26, S31, S37, S39, S46, S51, S58, S61, S64, S69, S70, S71, S72	21	28.77%

- The Agile Unified Process (AUP), a simplified revision of the Rational Unified Process (RUP), which applies agile practices such as refactoring, TDD, agile modeling, iterative and incremental development [55];
- Kanban, which is more visual and less prescriptive (when compared to other ASD methods) and is based on Toyota's "just-in-time" concept [56], [57], and
- Distributed Agile Software Development (DASD), which originates from the application of the principles, techniques and practices of Agile in distributed software development environments [58].

Scrum is found in 67.12% of the primary studies. The others use agile methods in combination with Scrum, with the exception of S7 and S43, which present software projects developed with XP only, and S63 which employs DASD only. 21 studies do not report the agile methods used, which represents almost 29% of the primary studies.

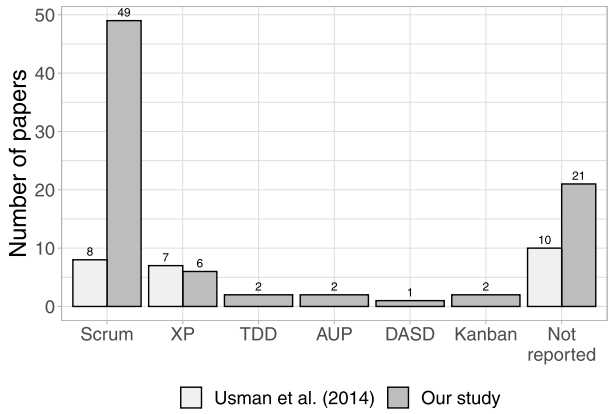

FIGURE 9. Agile methods: Comparison of our results with those of Usman *et al.* (2014).

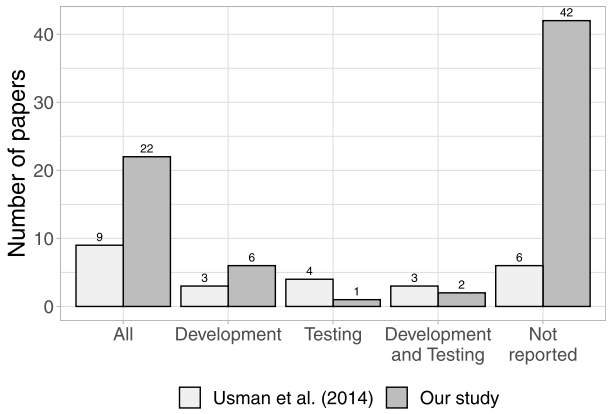

FIGURE 10. Development activities: Comparison of our results with those of Usman *et al.* (2014).

TABLE 14. Development activities considered.

Development activity	Papers	#	%
All	S1, S6, S16, S18, S19, S20, S23, S29, S30, S32, S33, S36, S40, S45, S49, S53, S54, S55, S56, S60, S65, S69	22	30.14%
Development	S14, S21, S38, S52, S64, S73	6	8.22%
Testing	S8	1	1.37%
Development and Testing	S50, S71	2	2.74%
Not reported	S2, S3, S4, S5, S7, S9, S10, S11, S12, S13, S15, S17, S22, S24, S25, S26, S27, S28, S31, S34, S35, S37, S39, S41, S42, S43, S44, S46, S47, S48, S51, S57, S58, S59, S61, S62, S63, S66, S67, S68, S70, S72	42	57.53%

Only two agile methods were identified in the Original Study: Scrum (eight papers) and XP (seven papers), as can be seen in Figure 9. The previous SLR was also unable to find examples of studies in which multiple agile methods were analyzed and compared. In the same way, Dantas *et al.* [24] covered only these two agile methods, with Scrum being the agile method most frequently used, representing 62.5% of the total of primary studies. We should also mention that a sound percentage of studies (28.77%, 40% in the Original Study, 37.5% in [24]) did not mention which agile method they used.

a: RQ4a: WHAT DEVELOPMENT ACTIVITIES HAVE BEEN INVESTIGATED?

The development activity is related to the phase of the software development process considered in the effort estimation model (analysis, design, development or test). Table 14 shows the development activities that the primary studies state are included in the estimate.

Note that most of the studies do not mention the development activity considered in the effort estimation (57.53%). Of those in which the development activity is identified, 22 papers included all the phases, six mentioned development only, one testing only, and two included a combination of estimation development and testing. In Dantas *et al.* [24], the authors only specified that just one of the papers focused exclusively on one activity in the development lifecycle (implementation). Figure 10 shows the comparison of these results with those obtained in the Original Study.

Given the dynamic characteristic of projects in ASD, it is understandable that most of the papers do not mention the development activity in the effort estimation. The short lifecycles in ASD normally include all the phases of the development process and these phases are, moreover, usually carried out in parallel and readjusted as necessary during the iteration. It is, therefore, safe to assume that most of the authors develop their estimation models from an overall perspective of the software development lifecycle. This aspect will become clearer when analyzing the planning levels that appear in the following question.

b: RQ4b: WHAT PLANNING LEVELS HAVE BEEN INVESTIGATED?

It is possible to perform planning at three different levels in an agile context, i.e., release, iteration and daily planning [59]. Effort estimation in ASD can, therefore, be performed in the following planning levels: Release, Iteration or Daily. None of the papers reviewed indicated that the effort estimation was carried out on a daily basis, but rather per iteration/sprint (45.21%) or per project/release (23.29%). It is also worth noting that effort estimation is carried out at both the iteration and the overall project planning level in seven studies, and they have, therefore, been included in the Sprint/Release category. 16 of the studies did not report the planning level. These results are shown in Table 15.

Figure 11 presents the comparison with the Original Study as regards the planning levels. As in the Original Study and in [24], software projects mostly deal with planning at the level of iterations or releases. The Original Study indicated

TABLE 15. Planning levels considered.

Planning level	Papers	#	%
Sprint	S3, S5, S10, S13, S16, S19, S20, S21, S22, S23, S24, S25, S28, S29, S31, S34, S36, S38, S40, S47, S48, S49, S50, S52, S54, S55, S57, S63, S65, S68, S69, S70, S73	33	45.21%
Release	S1, S2, S7, S9, S11, S12, S14, S15, S26, S35, S41, S42, S43, S46, S66, S67, S71	17	23.29%
Sprint/Release	S6, S18, S32, S33, S45, S53, S56	7	9.59%
Not reported	S4, S8, S17, S27, S30, S37, S39, S44, S51, S58, S59, S60, S61, S62, S64, S72	16	21.92%

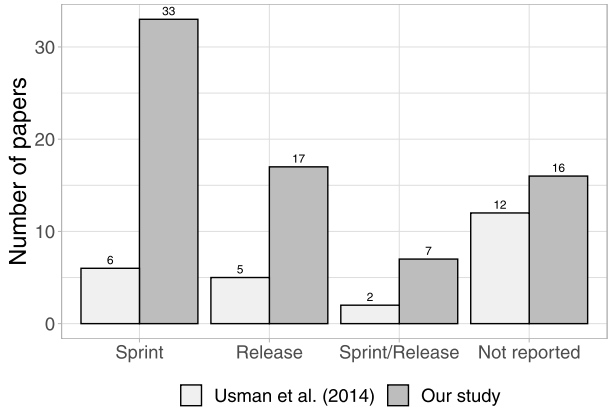

FIGURE 11. Planning levels: Comparison of our results with those of Usman *et al.* (2014).

that 48% of the cases did not mention the planning level, compared to 21.92% in the present study.

VI. DISCUSSION

A. PRINCIPAL FINDINGS

For each research question, Table 16 provides a more synthesized view of the findings obtained in the Original Study, that of Dantas *et al.* [24], and the present study. The most significant differences in our study from the Original Study have been highlighted in bold and are discussed below:

- **ASD still relies on expert opinion, but Machine Learning techniques have broken into ASD and are here to stay.**
  Although the subjective assessment made by experts continues to be the relevant information to be taken into account for the estimation of size and effort in ASD (especially Planning Poker with 24.66% of the total number of studies), there is a growing trend towards the development and use of techniques based on algorithms, systematized tools and the historical data of projects for the generation of estimates. Moreover, this prominent trend towards using techniques based on the intensive use of data (such as ML techniques with 20.55% of the total) for effort estimation in ASD is accompanied by the interest on the part of researchers to compare the accuracy and convenience of models based on data and expert opinion. To take advantage of both, combination-based effort estimation methods are promising.
- **The combination of MMRE and PRED(x) is widespread used in ASD, but other complementary accuracy metrics have come into play.**
  MMRE AND PRED(x) are the most used accuracy metrics, the latter being normally added as a complement to the former. However, additional accuracy metrics, such as R2, the percentage of accuracy, and MAE have increased their relevance.
- **Effect size is beginning to be reported even though in general the accuracy results are poorly reported in ASD, which makes the comparisons unreliable.**
  Accuracy continues to be a challenge but some improvements have been identified. On the other hand, 24.49% of the papers that reported the accuracy metric used reflected aspects concerning the validation of the models, even if little information could be retrieved from them. Moreover, 12.24% reported not only the statistical significance of the different accuracy values when comparing between models, but also the effect size. Despite all of the above, it should be noted that many studies continue to report inadequate accuracy results.
- **Story Points is the straightforward size metric for ASD, even though it is sometimes used in combination with other metrics.**
  There has been a decrease in the use of general-purpose size metrics and an increase in the use of size metrics that take the particularities of agile methods into account. Thus, few studies rely on use cases as a means to specify functional requirements (i.e., UCP), whereas Story Points has become the main size metric in ASD with a significant increase to 61.64% of the total number of studies. Story Points, Planning Poker and User Story are often considered to be closely linked. Simultaneously, there is a growing interest in the use of functional size measurement methods (COSMIC, IFPUG-FPA, NESMA-FPA, etc.) for size estimation. This effort is accompanied primarily by the willingness to bring these functional size metrics into ASD processes.
- **In ASD there is a great diversity of cost factors and a lack of standardization.**
  There is a great diversity of factors used in ASD. However, it was possible to group these factors and establish relationships among them, which indicates that they are not used in isolation but are closely linked to the characteristics of the projects. In this sense, the use of cost factors goes in line with the agile development principles, for example in the use of team and project factors in preference to the consideration of more technical aspects. Approximately 30% of the studies do not

TABLE 16. Comparison of SLRs findings.

	Category	Usman *et al.* (2014)	Our study	Dantas *et al.* (2018)
Effort/size estimation	Estimation methods	PPoker 24% ML 0% NN 8% ExpJudg 32% COSMIC 4% Regression 12% Algo 12% UCP 28% N.A. 8%	PPoker 24.66% **ML 20.55%** NN 17.81% **ExpJudg 10.96%** COSMIC 9,59% Regression 8.22% Algo 6.85% UCP 4.11% N.A. 4.11%	PPoker 16.67% ML 20.83% PPoker&ML 12.5% ExpJudg 16.67% BN 4.17% PPoker&BN 8.33% OptAlgo 4.17% N.A. 16.67%
	Accuracy metrics	MRE 44% PRED(x) 8% BRE 8% MER 4% N.A. 24%	MRE 47.95% **PRED(x) 26.03%** BRE 8.22% MER 6.85% N.A. 32.88% **Others (R2, % of accuracy, MAE)**	MRE 62.5% PRED(x) 29.17% BRE 8.33% MER 0% N.A. 29.17%
	Level of accuracy	- UCP and ExpJudg: acceptable MRE values (one study) - Other methods: not at least 25%	PPoker: MMRE = 0.41 ML: MMRE = 0.58 NN: MMRE = 0.23 FSM: MMRE = 0.41 ExpJudg: MMRE = 0.22	- Best results reported - PPoker and ExpJudg: unacceptable values - ML, BN, OptAlgo: better accuracy results
Effort predictors	Size metrics	SP 24% FP 12% COSMIC 4% LOC 4% UCP 28% N.A. 20%	**SP 61.64%** FP 8.22% COSMIC 13.7% LOC 9.59% **UCP 5.48%** N.A. 16.44%	SP 70.83% FP 12.5% COSMIC 0% LOC 0% UCP 0% N.A. 16.67%
	Cost factors	18 factors EXP 8% Skills 12% N.A. 32%	22 factors: **Team(T),Project(P),TECH,US** (T) EXP 23.29% (T) Skills 9.59% (P) CXTY 27.4% (P) REQ 4.11% N.A. 30.14%	10 factors: Team(T), Project(P) (T) EXP 29.17% (T) Skills 20.83% (P) CXTY 29.17% (P) REQ 37.5% N.A. 25%
Characteristics of the dataset	Domain	Ind 76% Acad 12% Ind&Acad 4% Sim 4% N.A. 4%	Ind 68.49% Acad 8.22% Ind&Acad 1.37% Sim 4.11% N.A. 17.81%	Ind 58.33% Acad 29.17% Ind&Acad 4.17% Sim 0% N.A. 8.33%
	Type	With 72% Cross 8% Sim 4% N.A. 16%	With 46.58% **Cross 31.51%** Sim 4.11% N.A. 17.81%	With 87.5% Cross 4.17% Sim 0% N.A. 8.33%
Agile methods	Methods	Scrum 32% XP 28% Scrum&XP 0% N.A. 40%	**Scrum 61.64%** XP 2.74% Scrum&XP 5.48% N.A. 28.77% **Others (TDD, AUP, Kanban, DASD)**	Scrum 45.83% XP 0% Scrum&XP 16.67% N.A. 37.5%
	Development activities	All 36% Dev 12%, Test 16% Dev&Test 12% N.A. 24%	All 30.14% Dev 8.22%, Test 1.37% Dev&Test 2.74% **N.A. 57.53%**	All ? Dev 4.17%, Test ? Dev&Test 0% N.A. ?
	Planning levels	Sprint 24% Release 20% Sprint&Release 8% N.A. 48%	**Sprint 45.21%** Release 23.29% Sprint&Release 9.59% N.A. 21.92%	Sprint ? Release 8.33% Sprint&Release 0% N.A. ?

indicate the cost factors considered. All in all, the lack of standardization as regards cost factors is still present and justify the need for more studies to fill this research gap.

- **Most of the datasets used for effort or size estimation in ASD come from the industry.**
  With regard to the characteristics of the datasets used for size or effort estimation in ASD, most of the datasets are from the industrial domain (68.49%) and are internal to a specific company (46.58%).
- **Cross-company datasets are trendy or are in a prominent position in effort estimation in ASD projects.**
  A significant increase in the use of cross-company data has been observed, which is consistent with the expansion of agile methods for global software development that has taken place in the last decade. In this respect, datasets such as the ISBSG[4] (International Software Benchmarking Standards Group) repository which, in its latest version of May 2017, incorporates the features of agile methods, provide useful information with which to conduct parametric estimations. This may have positive implications for the applicability of the models presented and the generalization of the results obtained.
- **Despite the great variety of agile methodologies, Scrum stands out.**
  Six agile methods have been identified in the present study (Scrum, XP, TDD, Agile Unified Process, Kanban and Distributed Agile Software Development). It is evident that the development of effort estimation models has been extended to cover the main agile development processes, with Scrum being the agile method most frequently used in the primary studies (representing 67.12%) of the total. This is also consistent with the fact that Scrum is the most frequently used agile method in the present-day development industry.
- **The development activity involved is usually not made explicit in ASD.**
  30.14% of the studies address effort estimation from a general perspective, that is, including all the project development activities. However, most of the studies do not make explicit the development activity involved (57.53%).
- **Estimation is mostly performed at sprint planning level in ASD.**
  Effort estimation in ASD is generally performed at the iteration/sprint (45.21%) or per project/release (23.29%) planning level, and under no circumstances takes place on a daily basis. All of this is consistent with the iterative and incremental approach of agile methods, in which development activities do not appear as blocks separated by strictly sequential phases, but rather a greater value is placed on the interaction between the different activities and their possible overlap.

[4]www.isbsg.org

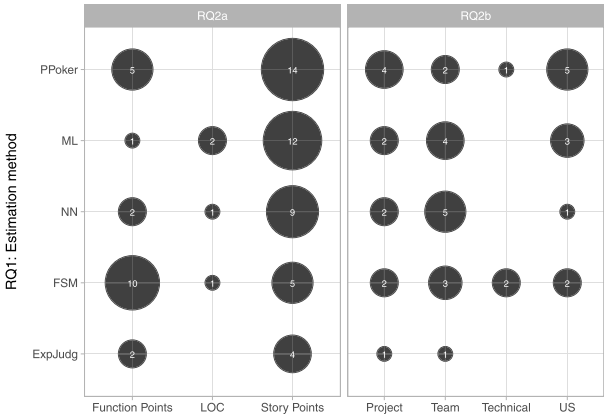

FIGURE 12. Bubble plot: RQ1 vs. RQ2.

We further discuss the results of this study by analyzing RQ2, RQ3 and RQ4 in relation to the most frequently used effort estimation methods identified in the primary studies, i.e., Planning Poker, Expert Judgment, Machine Learning, Neural Network and Functional Size Measurement. All of these estimation methods are shown in Table 8, which presents a summary of the statistics concerning the accuracy values. At this point, please recall that most estimation methods are applied to Scrum and that the other agile methods are mentioned in only a minority of studies.

First, Figure 12 shows the results in a bubble chart that combines data concerning effort predictors (RQ2) with the aforementioned estimation methods. Upon analyzing the size metrics (RQ2a), Story Points appears to be the most widely used metric, with a presence in almost 65% of the papers that use any of these estimation methods, and especially in Planning Poker and Machine Learning. However, Function Points are mostly used in papers in which Functional Size Measurement methods are proposed (62.5%). When analyzing the cost drivers (RQ2b), team factors are those most frequently used ones by these estimation methods and are used to the same extent by all of them. Note that the papers mostly include team factors in their Neural Network models, and project and User Story factors when employing Planning Poker estimation. Furthermore, technical factors are not used at all in those primary studies that report Machine Learning, Neural Network and Expert Judgment models.

Second, Figure 13 depicts the relationships between estimation methods and the characteristics of the datasets used (RQ3). Since the use of academic databases is negligible, with industrial databases prevailing in most of the primary studies (68.49% vs. 8.22%), this chart cannot provide much information in this respect. However, with regard to the domain of the datasets (RQ3a), it should be noted that FSM is the estimation method that proportionally uses more academic datasets, while papers based on Expert Judgment methods do not use academic datasets at all. When considering the type of dataset (RQ3b), Figure 13 suggests

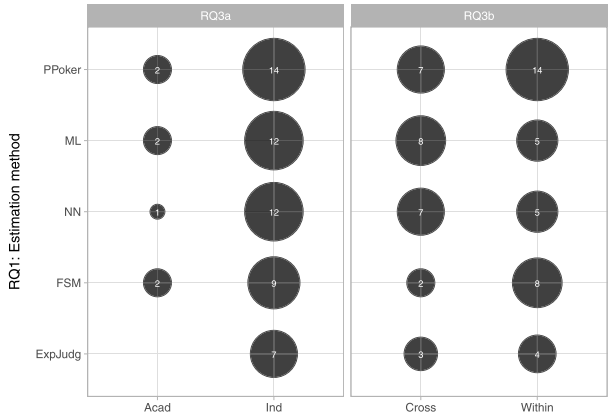

FIGURE 13. Bubble plot: RQ1 vs. RQ3.

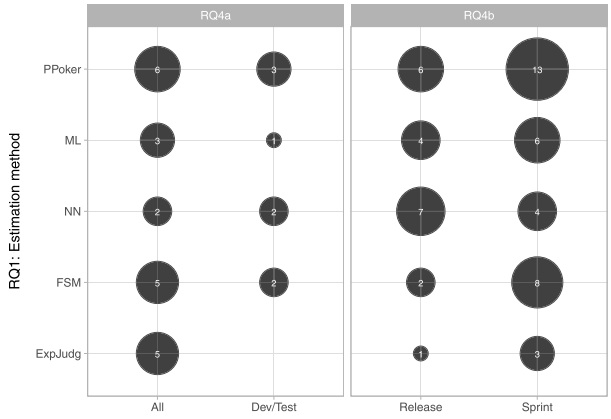

FIGURE 14. Bubble plot: RQ1 vs. RQ4.

that even if within-company data was used in most of the estimation models in absolute terms, this is not the case for Machine Learning and Neural Network methods. These models tend to rely on data from several companies in contrast to expert-based methods, such as Planning Poker and Expert Judgment, which mainly make use of within-dataset data (66.7% and 57.1% respectively). Finnally, it is worth noting that FSM methods mostly make use of within-dataset data by as much as 80% versus cross-company datasets.

Third, Figure 14 shows the relationships between estimation methods and the development activities considered in the models (RQ4a), in addition to the different levels at which the planning can be performed in an agile context (RQ4b). On the one hand, since most of the primary studies reported the values of the estimates without indicating the activity phases involved, the distribution among the different estimation methods is not very conclusive. However, note that the inclusion of all the phases while reporting the estimates prevails for all the estimation methods, with the exception of Neural Network, in which the ratio is the same (50%). The Expert Judgment methods particularly provide all their estimates including all the phases of the development process. On the other hand, with regard to the planning level at which the estimation is undertaken, this plot shows that, only in the case of Neural Network methods, the estimation at the release level is preferred to the estimation at the sprint level.

B. STUDY LIMITATIONS

The main limitation of this work concerns the possible biases introduced in the selection process. We attempted to avoid this bias by defining the search strategy for primary studies in accordance with Kitchenham's guidelines for performing SLRs in Software Engineering [40]. However, it is still not possible to ensure that all relevant studies have been included owing to the wide variety of documents. Moreover, three out of eight digital libraries used in the Original Study could not be accessed, which may have in some way reduced the number of candidate primary studies. ScienceDirect was also discarded, since this digital library did not allow us to ensure a similar search to that carried out in the other digital libraries, although we were able to verify that it did not contribute to the primary studies of the period until 2017 in any respect. The selection based on ACM Digital Library, IEEE Xplore, SCOPUS and WOS is, nevertheless, considered adequate, since these digital libraries are widely recognized by the scientific community and provide a sound and sufficient set of results. Despite the fact that our primary studies were retrieved from trustworthy sources, we did not consider this to be sufficient to guarantee the quality of the primary studies, and consequently performed a quality assessment, as the result of which eight studies were discarded owing to their low quality. The selection process was, overall, carried out in pairs in order to minimize potential biases, and any conflicts were resolved by a third party. If necessary, a debate then took place in order to reach a consensus, and all the reasons for the inclusion and exclusion of the studies were recorded at each stage.

Another threat to validity originates from missing data, which affects the reliability of the research. Indeed, we found few studies reporting all the data that we wished to extract. For example, 57.53% of the studies did not mention the development activity that was subject to estimation. In fact, this was the research question with the highest level of missing data. Less than 35% of the data was missing for the remaining research questions, although this figure was, in many cases, over 25%.

Finally, the integration of the results of the primary studies was not conducted via a meta-analysis to synthesize quantitative research questions. The underlying problem here was the difficulty of combining and synthesizing results from a diverse set of primary studies. However, a more in-depth analysis at different levels allowed us to obtain valuable insights that were used to identify trends in ASD effort estimation and potential cross-relationships between variables of interest. Furthermore, in order to ensure the comparability of our results with those obtained in the Original Study, we relied strongly on the protocol employed in that study, although some adaptations were necessary.

C. IMPLICATIONS FOR RESEARCH AND PRACTICE

This study indicates that researchers' interest in this topic has increased significantly in recent years. Our results also showed that software effort estimation studies concerning ASD are progressively focusing on data-based methods. Machine Learning is taking precedence, although Expert Judgment is still preferred by the industry owing to its intuitiveness [18]. The developers basically look to past projects or iterations, and draw on their own experiences to produce estimates for the stories [S58]. In the state of the practice as regards effort estimation in ASD, however, the dominant trend appears to be effort underestimation [60]. Expert Judgment is prone to bias and is subject to human error [S1], [S24]. Common psychological effects in human estimations, e.g., these experts' character traits or personal interests, may lead to biases impacting on the accuracy of their estimates [S17].

As a result, our study suggests that the overall understanding of how effort estimation in ASD is performed requires further exploration. In particular, we identified several implications for researchers.

It is first necessary to optimize the estimation techniques currently employed in agile software development projects. There is also a need to further analyze the effectiveness of individual soft computing techniques such as Neural Networks, Support Vector Machine, Fuzzy Logic and Bayesian Networks, in addition to the combined use of multiple techniques. Indeed, we found as many as nine primary studies that employed combination-based effort estimation methods [S6], [S14], [S55], [S56], [S57], [S60], [S61], [S70], [S71]. We can conclude that a combination of data-based methods (using project historical data and context-specific methods) with expert-based methods may be promising as regards improving accuracy levels.

Our study also showed that there has been a decrease in the use of general-purpose size metrics (e.g., UCP) and an increase in the use of size metrics that take the particularities of agile methods into account. In this respect, Story Points was the metric most frequently used to determine the size of the software. This result coincides with that of Usman *et al.* [60] who also found that Story Points is the size metric most frequently used by agile teams. However, some authors suggest that Story Points should be estimated collectively in an attempt to reach a team consensus so as to alleviate the chances of over-optimism and reduce the issues of anchoring and strong personalities [S1], [S36], [S63]. More research efforts are, therefore, required to provide evidence of the effectiveness of Story Points for effort estimation by taking into account the team characteristics, such as their experience and knowledge/expertise of agile methods.

The results of our study also highlight the relevance of team factors in ASD effort estimation, such as the developers' availability and their familiarity with the project [S19], [S40], [S47]. Other human factors such as motivation [S70] and leadership [S40] have also been identified, along with the ability to form proper working teams with balanced skills. However, there is still a need to standardize the cost factors used for effort estimation in agile software development. In this respect, insufficient training and the absence of a standard way in which teams can estimate their work are the most relevant causes of failed agile implementations [S1].

As a result, from a research perspective, this study was a valuable means to analyze the state of the art on effort estimation in agile software development. Moreover, the review protocol described in this article provides a step-by-step approach that can easily be applied by other researchers in order to update other SLRs.

Our findings also have practical implications. Project managers may wish to employ the results attained in this study to obtain an updated overview of how effort estimation is performed in ASD. Indeed, Usman *et al.* [61] organized the knowledge identified as a taxonomy of effort estimation in ASD. The proposed taxonomy provided a classification scheme with which to characterize the estimation activities of agile projects in four dimensions: estimation context, estimation technique, effort predictors and effort estimate. Project managers must be aware of the specific effort estimation techniques and cost factors which may, in turn, affect the effectiveness of their developments.

VII. CONCLUSION AND FUTURE WORK

We have presented an SLR with the objective of updating the study concerning effort estimation in ASD carried out in 2014 by Usman *et al.* [21]. The publication period employed to identify the primary studies was from December 2013 to January 2020. The present study has followed the original review protocol, but with some minor changes. The current update focuses on 73 papers that provided new evidence with which to characterize estimation activities in agile projects. These studies have been analyzed with the purpose of understanding their findings in an attempt to shed light on various issues, and particularly on the set of issues that have arisen during the period selected. With regard to the integration of the findings, we considered that comparing the results of both subsets of primary studies was more interesting than merging them to present an overall result regarding all the papers on this subject published from 2001 to the present.

Our results show that effort estimation methods have been used in six different agile methods (i.e., Scrum, XP, TDD, Agile Unified Process, Kanban and Distributed Agile Software Development). Planning Poker has become the estimation method that is currently most widely used in the primary studies, and it is very closely related to the most frequently used size metric (story points) and the way in which the software requirements are specified in these agile methods (user story). Despite the fact that expert-based estimation methods continue to play an important role, there is a prominent trend towards studying techniques based on the intensive use of data. In this respect, most of the datasets are industrial domain datasets and are internal to a specific company.

In spite of the vast number of approaches, the accuracy of software effort estimation models for agile development is still inconsistent. The results of our study highlighted that accuracy remains a challenge in most of the papers analyzed. However, we have identified significant improvements in this respect. First, an increasing number of papers report adequate ranges of accuracy values. Second, of the papers that report the accuracy metric used, an increasing percentage also reflect aspects concerning the validation of the models. Third, those papers that report the statistical significance of the different accuracy values have also begun to report the effect size.

Project managers must be aware of the specific effort estimation techniques and cost factors, that may, in turn, affect the effectiveness of their projects. They may, therefore, wish to utilize the results of this study to obtain an updated overview of how effort estimation is performed in agile development projects. Our results also suggested that a combination of data-based and expert opinion methods may be promising.

Finally, we believe that performing SLR updates with adequate integration or comparison with the findings of the original study is necessary, and more researchers should be engaged in this kind of studies in order to keep the evidence regarding a particular topic up-to-date. When these updates are performed by different researchers to the original authors, which is the case of this study, this may be especially useful as regards detecting possible bias in the original study and/or confirming or refining their results.

In future work, we plan to investigate the influence of cost factors on the accuracy of the estimation models used in ASD. We particularly wish to assess to what extent the use of specific cost factors may imply more accurate estimates in particular contexts. We also plan to perform incremental updates in order to follow the evolution of effort estimation methods and cost factors in ASD. Finally, the overall understanding of how effort estimation in ASD can be improved may lead to the proposal of an estimation model based on the historical data obtained from the ISBSG repository. This would require a detailed analysis of the relevant variables and their relationships. Since ISBSG is a cross-company repository, with data on software projects developed in different countries over the world, the possibilities of generalization and applicability of the results are promising.

APPENDIX A
PRIMARY STUDIES

This section provides the primary studies resulting from the selection process, sorted alphabetically by title:

[S1] T. J. Gandomani, K. T. Wei, and A. K. Binhamid, "A Case Study Research on Software Cost Estimation Using Experts' Estimates, Wideband Delphi, and Planning Poker Technique," International Journal of Software Engineering and its Applications, vol. 8, no. 11, pp. 173–182, 2014.

[S2] A. Kaushik, D. K. Tayal, and K. Yadav, "A Comparative Analysis on Effort Estimation for Agile and Non-agile Software Projects Using DBN-ALO," Arabian Journal for Science and Engineering, pp. 1–14, 2019, doi: 10.1007/s13369-019-04250-6.

[S3] M. Pozenel and T. Hovelja, "A Comparison of the Planning Poker and Team Estimation Game: A Case Study in Software Development Capstone Project Course," International Journal of Engineering Education, vol. 35, no. 1(A), pp. 195–208, 2019.

[S4] M. Choetkiertikul, H. K. Dam, T. Tran, T. Pham, A. Ghose, and T. Menzies, "A Deep Learning Model for Estimating Story Points," IEEE Transactions on Software Engineering, vol. 45, no. 7, pp. 637–656, Jul. 2019.

[S5] J. M. Alostad, L. R. A. Abdullah, and L. S. Aali, "A Fuzzy based Model for Effort Estimation in Scrum Projects," International Journal of Advanced Computer Science and Applications, vol. 8, no. 9, pp. 270–277, Sep. 2017.

[S6] N. A. Bhaskaran and V. Jayaraj, "A Hybrid Effort Estimation Technique for Agile Software Development (HEETAD)," IJEAT, vol. 9, no. 1, pp. 1078–1087, Oct. 2019, doi: 10.35940/ijeat.A9480.109119.

[S7] E. Karunakaran and N. Sreenath, "A Method to Effort Estimation for XP Projects in Clients Perspective," International Journal of Applied Engineering Research, vol. 10, no. 7, pp. 18529–18550, 2015.

[S8] N. V. Prykhodko and S. B. Prykhodko, "A Multiple Non-Linear Regression Model to Estimate the Agile Testing Efforts for Small Web Projects," Radio Electronics, Computer Science, Control, no. 2, pp. 158–166, 2019, doi: 10.15588/1607-3274-2019-2-17.

[S9] T. T. Khuat and M. H. Le, "A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies," Journal of Intelligent Systems, vol. 27, no. 3, pp. 489–506, 2018, doi: 10.1515/jisys-2016-0294.

[S10] I. Manga and N. V. Blamah, "A Particle Swarm Optimization-based Framework for Agile Software Effort Estimation," The International Journal of Engineering and Science, vol. 3, no. 6, pp. 30–36, 2014.

[S11] H. Huijgens and R. van Solingen, "A Replicated Study on Correlating Agile Team Velocity Measured in Function and Story Points," in 5th International Workshop on Emerging Trends in Software Metrics (WETSoM), Hyderabad, India, 2014, pp. 30–36, doi: 10.1145/2593868.2593874.

[S12] Ch. P. Rao, P. S. Kumar, S. R. Sree, and J. Devi, "An Agile Effort Estimation Based on Story Points Using Machine Learning Techniques," in Second International Conference on Computational Intelligence and Informatics, 2018, pp. 209–219, doi: 10.1007/978-981-10-8228-3_20.

[S13] R. Popli and N. Chauhan, "An Agile Software Estimation Technique based on Regression Testing Efforts," in 13th Annual International Software Testing Conference in India, Bangalore, India, 2013, pp. 04–05.

[S14] S. Basri, N. Kama, H. M. Sarkan, S. Adli, and F. Haneem, "An Algorithmic-based Change Effort Estimation Model for Software Development," in 23rd Asia-Pacific

Software Engineering Conference (APSEC), Dec. 2016, pp. 177–184, doi: 10.1109/APSEC.2016.034.

[S15] T. T. Khuat and M. H. Le, "An Effort Estimation Approach for Agile Software Development using Fireworks Algorithm Optimized Neural Network," International Journal of Computer Science and Information Security, vol. 14, no. 7, pp. 122–130, Jul. 2016.

[S16] E. Dantas, A. Costa, M. Vinicius, M. Perkusich, H. Almeida, and A. Perkusich, "An Effort Estimation Support Tool for Agile Software Development: An Empirical Evaluation," in 31st International Conference on Software Engineering and Knowledge Engineering (SEKE), Jul. 2019, pp. 82–87, doi: 10.18293/SEKE2019-141.

[S17] H. H. Arifin, J. Daengdej, and N. T. Khanh, "An Empirical Study of Effort-Size and Effort-Time in Expert-Based Estimations," in 8th International Workshop on Empirical Software Engineering in Practice (IWESEP), Mar. 2017, pp. 35–40, doi: 10.1109/IWESEP.2017.21.

[S18] A. T. Raslan and N. R. Darwish, "An Enhanced Framework for Effort Estimation of Agile Projects," International Journal of Intelligent Engineering and Systems, vol. 11, no. 3, pp. 205–214, 2018, doi: 10.22266/ijies2018.0630.22.

[S19] O. Malgonde and K. Chari, "An Ensemble-based Model for Predicting Agile Software Development Effort," in 24th Workshop on Information Technologies and Systems (WITS), Auckland, New Zealand, Dec. 2014.

[S20] S. Dhir, D. Kumar, and V. b. Singh, "An Estimation Technique in Agile Archetype using Story Points and Function Point Analysis," International Journal of Process Management and Benchmarking, vol. 7, no. 4, pp. 518–539, Jan. 2017, doi: 10.1504/IJPMB.2017.086933.

[S21] B. Tanveer, A. M. Vollmer, S. Braun, and N. bin Ali, "An Evaluation of Effort Estimation Supported by Change Impact Analysis in Agile Software Development," Journal of Software: Evolution and Process, vol. 31, no. 5, pp. 1–17, 2019, doi: 10.1002/smr.2165.

[S22] S. Kumar C, A. Kumari, S. P. Ramalingam, and S. P. Ramalingam, "An Optimized Agile Estimation Plan using Harmony Search Algorithm," International Journal of Engineering and Technology, vol. 6, pp. 1994–2001, Oct. 2014.

[S23] P. Rola and D. Kuchta, "Application of Fuzzy Sets to the Expert Estimation of Scrum-Based Projects," Symmetry, vol. 11, no. 8, pp. 1–23, 2019, doi: 10.3390/sym11081032.

[S24] M. Ochodek, "Approximation of COSMIC Functional Size of Scenario-based Requirements in Agile Based on Syntactic Linguistic Features: A Replication Study," in Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM/Mensura), Oct. 2016, pp. 201–211, doi: 10.1109/IWSM-Mensura.2016.039.

[S25] S. Dragicevic, S. Celar, and M. Turic, "Bayesian Network Model for Task Effort Estimation in Agile Software Development," Journal of Systems and Software, vol. 127, pp. 109–119, May 2017, doi: 10.1016/j.jss.2017.01.027.

[S26] R. Mas'ad, R. Ñanculef, and H. Astudillo, "BlackSheep: Dynamic Effort Estimation in Agile Software Development Using Machine Learning," in 22nd Ibero-American Conference on Software Engineering (CIbSE), La Habana, Cuba, Apr. 2019, pp. 16–29.

[S27] V. Lenarduzzi and D. Taibi, "Can Functional Size Measures Improve Effort Estimation in SCRUM?," in Ninth International Conference on Software Engineering Advances (ICSEA), Nice, France, Oct. 2014.

[S28] M. Conoscenti, V. Besner, A. Vetrò, and D. M. Fernández, "Combining Data Analytics and Developers Feedback for Identifying Reasons of Inaccurate Estimations in Agile Software Development," Journal of Systems and Software, vol. 156, pp. 126–135, 2019, doi: 10.1016/j.jss.2019.06.075.

[S29] A. Vetrò, R. Dürre, M. Conoscenti, D. M. Fernández, and M. Jørgensen, "Combining Data Analytics with Team Feedback to Improve the Estimation Process in Agile Software Development," Foundations of Computing and Decision Sciences, vol. 43, no. 4, pp. 305–334, 2018, doi: 10.1515/fcds-2018-0016.

[S30] E. Ungan, N. Çizmeli, and O. Demirörs, "Comparison of Functional Size Based Estimation and Story Points, Based on Effort Estimation Effectiveness in SCRUM Projects," in 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Aug. 2014, pp. 77–80, doi: 10.1109/SEAA.2014.83.

[S31] K. Moharreri, A. V. Sapre, J. Ramanathan, and R. Ramnath, "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments," in IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), Jun. 2016, vol. 2, pp. 135–140, doi: 10.1109/COMPSAC.2016.85.

[S32] M. Usman, K. Petersen, J. Börstler, and P. S. Neto, "Developing and Using Checklists to Improve Software Effort Estimation: A Multi-Case Study," Journal of Systems and Software, vol. 146, pp. 286–309, 2018, doi: 10.1016/j.jss.2018.09.054.

[S33] M. Salmanoglu, T. Hacaloglu, and O. Demirors, "Effort Estimation for Agile Software Development: Comparative Case Studies Using COSMIC Functional Size Measurement and Story Points," in 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement (IWSM/Mensura), Gothenburg, Sweden, 2017, pp. 41–49, doi: 10.1145/3143434.3143450.

[S34] H. M. Premalatha and C. V. Srikrishna, "Effort Estimation in Agile Software Development using Evolutionary Cost-Sensitive Deep Belief Network," International Journal of Intelligent Engineering and Systems, vol. 12, no. 2, pp. 261–269, 2019, doi: 10.22266/ijies2019.0430.25.

[S35] S. Bilgaiyan, S. Mishra, and M. Das, "Effort Estimation in Agile Software Development Using Experimental Validation of Neural Network Models," Int. j. inf. tecnol., vol. 11, no. 3, pp. 569–573, Sep. 2019, doi: 10.1007/s41870-018-0131-2.

[S36] B. Tanveer, L. Guzmán, and U. M. Engel, "Effort Estimation in Agile Software Development: Case Study and Improvement Framework," Journal of Software: Evolution and Process, vol. 29, no. 11, pp. 1–14, Nov. 2017, doi: 10.1002/smr.1862.

[S37] R. G. F. Soares, "Effort Estimation via Text Classification And Autoencoders," in International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, Jul. 2018, pp. 01–08, doi: 10.1109/IJCNN.2018.8489030.

[S38] C. Commeyne, A. Abran, and R. Djouab, "Effort Estimation with Story Points and COSMIC Function Points: An Industry Case Study," Software Measurement News, vol. 21, no. 1, pp. 25–36, 2016.

[S39] L. D. Radu, "Effort Prediction in Agile Software Development with Bayesian Networks," in 14th International Conference on Software Technologies (ICSOFT), 2019, pp. 238–245, doi: 10.5220/0007842802380245.

[S40] M. Owais and R. Ramakishore, "Effort, Duration and Cost Estimation in Agile Software Development," in Ninth International Conference on Contemporary Computing (IC3), Aug. 2016, pp. 1–5, doi: 10.1109/IC3.2016.7880216.

[S41] S. M. Satapathy and S. K. Rath, "Empirical Assessment of Machine Learning Models for Agile Software Development Effort Estimation using Story Points," Innovations in Systems and Software Engineering, vol. 13, no. 2–3, pp. 191–200, Sep. 2017, doi: 10.1007/s11334-017-0288-z.

[S42] A. Panda, S. M. Satapathy, and S. K. Rath, "Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points," Procedia Computer Science, vol. 57, pp. 772–781, Jan. 2015, doi: 10.1016/j.procs.2015.07.474.

[S43] S. Porru, A. Murgia, S. Demeyer, M. Marchesi, and R. Tonelli, "Estimating Story Points from Issue Reports," in 12th International Conference on Predictive Models and Data Analytics in Software Engineering (PROMISE), Ciudad Real, Spain, 2016, pp. 2:1–2:10, doi: 10.1145/2972958.2972959.

[S44] P. Chongpakdee and W. Vatanawood, "Estimating User Story Points Using Document Fingerprints," in 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017, pp. 149–152.

[S45] C. J. Torrecilla-Salinas, J. Sedeño, M. J. Escalona, and M. Mejías, "Estimating, Planning and Managing Agile Web Development Projects under a Value-based Perspective," Information and Software Technology, vol. 61, pp. 124–144, May 2015, doi: 10.1016/j.infsof.2015.01.006.

[S46] R. Popli and N. Chauhan, "Estimation in Agile Environment using Resistance Factors," in International Conference on Information Systems and Computer Networks (ISCON), Mar. 2014, pp. 60–65, doi: 10.1109/ICISCON.2014.6965219.

[S47] S. Grapenthin, S. Poggel, M. Book, and V. Gruhn, "Facilitating Task Breakdown in Sprint Planning Meeting 2 with an Interaction Room: An Experience Report," in 40th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Aug. 2014, pp. 1–8, doi: 10.1109/SEAA.2014.71.

[S48] V. Lenarduzzi, I. Lunesu, M. Matta, and D. Taibi, "Functional Size Measures and Effort Estimation in Agile Development: A Replicated Study," in International Conference on Agile Software Development (Agile Processes in Software Engineering and Extreme Programming), May 2015, pp. 105–116, doi: 10.1007/978-3-319-18612-2_9.

[S49] A. Sellami, M. Haoues, N. Borchani, and N. Bouassida, "Guiding the Functional Change Decisions in Agile Project: An Empirical Evaluation," in International Conference on Software Technologies (ICSOFT), 2018, pp. 327–348, doi: 10.1007/978-3-030-29157-0_15.

[S50] A. R. Ahmed, M. Tayyab, S. N. Bhatti, A. J. Alzahrani, and M. I. Babar, "Impact of Story Point Estimation on Product using Metrics in Scrum Development Process," International Journal of Advanced Computer Science and Applications, vol. 8, no. 4, pp. 385–391, Apr. 2017.

[S51] P. V. de Campos Souza, A. J. Guimaraes, V. S. Araujo, T. S. Rezende, and V. J. S. Araujo, "Incremental Regularized Data Density-Based Clustering Neural Networks to Aid in the Construction of Effort Forecasting Systems in Software Development," Applied Intelligence, vol. 49, no. 9, pp. 3221–3234, 2019, doi: 10.1007/s10489-019-01449-w.

[S52] R. Štrba, J. Štolfa, S. Štolfa, and M. Košinár, "Intelligent Software Support of the SCRUM Process," in 24th International Conference on Information Modelling and Knowledge Bases (EJC), Jun. 2014, pp. 408–416, doi: 10.3233/978-1-61499-472-5-408.

[S53] N. Zabkar, T. Hovelja, J. Urevc, and V. Mahnic, "Introducing Agile Software Development: Lessons Learned from the First Scrum Project in a Slovenian Company," in International Conference on Advances in Management Engineering and Information Technology, Jan. 2015, doi: 10.2495/AMEIT140781.

[S54] T. Hacaloglu and O. Demirors, "Measureability of Functional Size in Agile Software Projects: Multiple Case Studies with COSMIC FSM," in 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), 2019, pp. 204–211, doi: 10.1109/SEAA.2019.00041.

[S55] M. Adnan and M. Afzal, "Ontology based Multiagent Effort Estimation System for Scrum Agile Method," IEEE Access, vol. 5, pp. 25993–26005, 2017, doi: 10.1109/ACCESS.2017.2771257.

[S56] M. Adnan, M. Afzal, and K. H. Asif, "Ontology-Oriented Software Effort Estimation System for E-Commerce Applications Based on Extreme Programming and Scrum Methodologies," The Computer Journal, vol. 62, no. 11, pp. 1605–1624, 2019, doi: 10.1093/comjnl/bxy141.

[S57] D. Taibi, V. Lenarduzzi, P. Diebold, and I. Lunesu, "Operationalizing the Experience Factory for Effort Estimation in Agile Processes," in 21st International Conference on Evaluation and Assessment in Software Engineering (EASE), Karlskrona, Sweden, 2017, pp. 31–40, doi: 10.1145/3084226.3084240.

TABLE 17. Search string used in each digital library.

Digital library	Search string
ACM Digital Library	[[[Publication Title: agile] OR [Publication Title: "extreme programming"] OR [Publication Title: scrum] OR [Publication Title: "feature driven development"] OR [Publication Title: "dynamic systems development method"] OR [Publication Title: "crystal software development"] OR [Publication Title: "crystal methodology"] OR [Publication Title: "adaptive software development"] OR [Publication Title: "lean software development"]] AND [[Publication Title: estimat*] OR [Publication Title: predict*] OR [Publication Title: forecast*] OR [Publication Title: calculat*] OR [Publication Title: assessment] OR [Publication Title: measur*] OR [Publication Title: sizing]] AND [[Publication Title: effort] OR [Publication Title: resource] OR [Publication Title: cost] OR [Publication Title: size] OR [Publication Title: metric] OR [Publication Title: "user story"] OR [Publication Title: velocity]] AND [Publication Title: software]] OR [[[Abstract: agile] OR [Abstract: "extreme programming"] OR [Abstract: scrum] OR [Abstract: "feature driven development"] OR [Abstract: "dynamic systems development method"] OR [Abstract: "crystal software development"] OR [Abstract: "crystal methodology"] OR [Abstract: "adaptive software development"] OR [Abstract: "lean software development"]] AND [[Abstract: estimat*] OR [Abstract: predict*] OR [Abstract: forecast*] OR [Abstract: calculat*] OR [Abstract: assessment] OR [Abstract: measur*] OR [Abstract: sizing]] AND [[Abstract: effort] OR [Abstract: resource] OR [Abstract: cost] OR [Abstract: size] OR [Abstract: metric] OR [Abstract: "user story"] OR [Abstract: velocity]] AND [Abstract: software]] OR [[[Keywords: agile] OR [Keywords: "extreme programming"] OR [Keywords: scrum] OR [Keywords: "feature driven development"] OR [Keywords: "dynamic systems development method"] OR [Keywords: "crystal software development"] OR [Keywords: "crystal methodology"] OR [Keywords: "adaptive software development"] OR [Keywords: "lean software development"]] AND [[Keywords: estimat*] OR [Keywords: predict*] OR [Keywords: forecast*] OR [Keywords: calculat*] OR [Keywords: assessment] OR [Keywords: measur*] OR [Keywords: sizing]] AND [[Keywords: effort] OR [Keywords: resource] OR [Keywords: cost] OR [Keywords: size] OR [Keywords: metric] OR [Keywords: "user story"] OR [Keywords: velocity]] AND [Keywords: software]] AND [Publication Date: (12/01/2013 TO 03/31/2020)]
IEEE Xplore	((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") 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 (software)) Filters Applied: Conferences Journals; 2013 - 2020
SCOPUS	TITLE-ABS-KEY((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") 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 (software)) AND (LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017) OR LIMIT-TO (PUBYEAR, 2016) OR LIMIT-TO (PUBYEAR, 2015) OR LIMIT-TO (PUBYEAR, 2014) OR LIMIT-TO (PUBYEAR, 2013)) AND (LIMIT-TO (DOCTYPE, "cp") OR LIMIT-TO (DOCTYPE, "ar") OR LIMIT-TO (DOCTYPE, "re")) AND (LIMIT-TO (LANGUAGE, "English"))
Web of Science (WOS)	(TS=((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") 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 (software))) AND LANGUAGE: (English) Refined by: DOCUMENT TYPES: (PROCEEDINGS PAPER OR ARTICLE OR REVIEW) Timespan: 2013-2020. Indexes: SCI-EXPANDED, SSCI, A&HCI, CPCI-S, CPCI-SSH, BKCI-S, BKCI-SSH, ESCI, CCR-EXPANDED, IC.

[S58] S. Garg and D. Gupta, "PCA based Cost Estimation Model for Agile Software Development Projects," in International Conference on Industrial Engineering and Operations Management (IEOM), Dubai, United Arab Emirates, Mar. 2015, pp. 1–7, doi: 10.1109/IEOM.2015.7228109.

[S59] T. J. Gandomani, H. Faraji, and M. Radnejad, "Planning Poker in Cost Estimation in Agile Methods: Averaging Vs. Consensus," in IEEE 5th Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, Feb. 2019, pp. 66–71, doi: 10.1109/KBEI.2019.8734960.

[S60] S. Basri, N. Kama, F. Haneem, and S. Adli, "Predicting Effort for Requirement Changes During Software Development," in Seventh Symposium on Information and Communication Technology (SoICT), Ho Chi Minh City, Viet Nam, 2016, pp. 380–387, doi: 10.1145/3011077.3011096.

[S61] R. Popli, N. Chauhan, and H. Sharma, "Prioritising User Stories in Agile Environment," in International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Feb. 2014, pp. 515–519, doi: 10.1109/ICICICT.2014.6781336.

[S62] J. F. Dumas-Monette and S. Trudel, "Requirements Engineering Quality Revealed through Functional Size Measurement: An Empirical Study in an Agile Context," in Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement (IWSM/Mensura), Oct. 2014, pp. 222–232, doi: 10.1109/IWSM.Mensura.2014.43.

[S63] W. Aslam, F. Ijaz, M. I. Lali, and W. Mehmood, "Risk Aware and Quality Enriched Effort Estimation for Mobile Applications in Distributed Agile Software Development," Journal of Information Science and Engineering, vol. 33, no. 6, pp. 1481–1500, Nov. 2017, doi: 10.6688/JISE.2017.33.6.6.

[S64] M. Lusky, C. Powilat, and S. Böhm, "Software Cost Estimation for User-Centered Mobile App Development in Large Enterprises," in International Conference on Advances in Human Factors, Software, and Systems

Engineering (AHFE), 2018, pp. 51–62, doi: 10.1007/978-3-319-60011-6_6.

[S65] W. Septian and W. Gata, "Software Development Framework on Small Team using Agile Framework for Small Projects (AFSP) with Neural Network Estimation," in 11th International Conference on Information Communication Technology and System (ICTS), Oct. 2017, pp. 259–264, doi: 10.1109/ICTS.2017.8265680.

[S66] S. M. Satapathy, A. Panda, and S. K. Rath, "Story Point Approach based Agile Software Effort Estimation using Various SVR Kernel Methods," in International Conference on Software Engineering and Knowledge Engineering (SEKE), 2014, vol. 2014-January, pp. 304–307.

[S67] A. Zakrani, A. Najm, and A. Marzak, "Support Vector Regression Based on Grid-Search Method for Agile Software Effort Prediction," in IEEE 5th International Congress on Information Science and Technology (CiSt), Oct. 2018, pp. 492–497, doi: 10.1109/CIST.2018.8596370.

[S68] S. Grapenthin, M. Book, T. Richter, and V. Gruhn, "Supporting Feature Estimation with Risk and Effort Annotations," in 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), Aug. 2016, pp. 17–24, doi: 10.1109/SEAA.2016.24.

[S69] J. Angara, S. Prasad, and G. Sridevi, "Towards Benchmarking User Stories Estimation with COSMIC Function Points: A Case Example of Participant Observation," Int J Elec & Comp Eng (IJECE), vol. 8, no. 5, pp. 3076–3083, Oct. 2018, doi: 10.11591/ijece.v8i5.pp3076-3083.

[S70] S. K. Khatri, S. Malhotra, and P. Johri, "Use Case Point Estimation Technique in Software Development," in 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Sep. 2016, pp. 123–128, doi: 10.1109/ICRITO.2016.7784938.

[S71] A. E. D. Hamouda, "Using Agile Story Points as an Estimation Technique in CMMI Organizations," in Agile Conference, Jul. 2014, pp. 16–23, doi: 10.1109/AGILE.2014.11.

[S72] E. Scott and D. Pfahl, "Using Developers' Features to Estimate Story Points," in International Conference on Software and Systems Process (ICSSP), Gothenburg, Sweden, May 2018, pp. 106–110, doi: 10.1145/3202710.3203160.

[S73] E. G. Wanderley, A. Vasconcelos, and B. T. Avila, "Using Function Points in Agile Projects: A Comparative Analysis Between Existing Approaches," in Brazilian Workshop on Agile Methods (WBMA), 2017, pp. 47–59, doi: 10.1007/978-3-319-73673-0_4.

APPENDIX B
SEARCH STRING USED IN EACH DIGITAL LIBRARY

See Table 17.

REFERENCES

[1] O. Hazzan and Y. Dubinsky, "The agile manifesto," in *Agile Anywhere* (Springer Briefs in Computer Science). Cham, Switzerland: Springer, 2014, pp. 9–14.

[2] Agile Alliance, *Agile Practice Guide*, 1st ed. Newtown Square, NM, USA: Project Management Institute, 2017.

[3] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, nos. 9–10, pp. 833–859, Aug. 2008.

[4] C. Larman and V. R. Basili, "Iterative and incremental developments. A brief history," *Computer*, vol. 36, no. 6, pp. 47–56, Jun. 2003, doi: 10.1109/MC.2003.1204375.

[5] J. Erickson, K. Lyytinen, and K. Siau, "Agile modeling, agile software development, and extreme programming: The state of research," *J. Database Manage.*, vol. 16, no. 4, pp. 88–100, Oct. 2005.

[6] K. Schwaber, "SCRUM development process," in *Business Object Design and Implementation*, J. Sutherland, C. Casanave, J. Miller, P. Patel, and G. Hollowell, Eds. London, U.K.: Springer, 1997, pp. 117–134, doi: 10.1007/978-1-4471-0947-1_11.

[7] K. Schwaber and M. Beedle, *Agile Software Development With Scrum*, 1st ed. London, U.K.: Pearson, 2001.

[8] K. Beck, *Extreme Programming Explained: Embrace Change*. Boston, MA, USA: Addison-Wesley, 2000.

[9] S. R. Palmer and J. M. Felsing, *A Practical Guide to Feature-Driven Develop.*, 1st ed. London, U.K.: Pearson, 2001.

[10] M. B. Poppendieck and T. Poppendieck, *Lean Software Development: An Agile Toolkit*. Boston, MA, USA: Addison-Wesley, 2003.

[11] J. Highsmith, *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. New York, NY, USA: Dorset House, 2000.

[12] A. Cockburn, *Agile Software Development*. Boston, MA, USA: Addison-Wesley, 2002.

[13] J. Stapleton, *Dynamic Systems Development Method: The Method in Practice*. Boston, MA, USA: Addison-Wesley, 1997.

[14] A. Trendowicz and R. Jeffery, *Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success*. Berlin, Germany: Springer, 2014.

[15] M. Unterkalmsteiner, T. Gorschek, A. M. Islam, C. K. Cheng, R. B. Permadi, and R. Feldt, "Evaluation and measurement of software process improvement—A systematic literature review," *IEEE Trans. Softw. Eng.*, vol. 38, no. 2, pp. 398–424, Mar./Apr. 2012.

[16] A. R. Altaleb and A. M Gravell, "Effort estimation across mobile app platforms using agile processes: A systematic literature review," *J. Softw.*, vol. 13, no. 4, pp. 242–259, Apr. 2018, doi: 10.17706/jsw.13.4.242-259.

[17] M. Jorgensen and M. Shepperd, "A systematic review of software development cost estimation studies," *IEEE Trans. Softw. Eng.*, vol. 33, no. 1, pp. 33–53, Jan. 2007.

[18] S. P. Pillai, S. D. Madhukumar, and T. Radharamanan, "Consolidating evidence based studies in software cost/effort estimation—A tertiary study," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2017, pp. 833–838, doi: 10.1109/TENCON.2017.8227974.

[19] S. Shekhar and U. Kumar, "Review of various software cost estimation techniques," *Int. J. Comput. Appl.*, vol. 141, no. 11, pp. 0975–8887, May 2016.

[20] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," EBSE, Goyang-si, South Korea, Tech. Rep. EBSE-2007-01, Jul. 2007.

[21] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in agile software development: A systematic literature review," in *Proc. 10th Int. Conf. Predictive Models Softw. Eng. (PROMISE)*, 2014, pp. 82–91, doi: 10.1145/2639490.2639503.

[22] T. Schweighofer, A. Kline, L. Pavlic, and H. Marjan, "How is effort estimated in agile software development projects?" in *Proc. 5th Workshop Softw. Qual., Anal., Monitor., Improvement, Appl. (SQAMIA)*, Aug. 2016, pp. 73–80.

[23] S. Bilgaiyan, S. Sagnika, S. Mishra, and M. Das, "A systematic review on software cost estimation in agile software development," *J. Eng. Sci. Technol. Rev.*, vol. 10, no. 4, pp. 51–64, 2017, doi: 10.25103/jestr.104.08.

[24] E. Dantas, M. Perkusich, E. Dilorenzo, D. F. S. Santos, H. Almeida, and A. Perkusich, "Effort estimation in agile software development: An updated review," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 28, nos. 11–12, pp. 1811–1831, Nov. 2018, doi: 10.1142/S0218194018400302.

[25] S. W. Munialo and G. M. Muketha, "A review of agile software effort estimation methods," *Int. J. Comput. Appl. Technol. Res.*, vol. 5, no. 9, pp. 612–618, Sep. 2016, doi: 10.7753/IJCATR0509.1009.

[26] M. Vyas, A. Bohra, C. S. Lamba, and A. Vyas, "A review on software cost and effort estimation techniques for agile development process," *Int. J. Recent Res. Aspects*, vol. 5, no. 1, pp. 1–5, Mar. 2018.

[27] R. Hoda, N. Salleh, J. Grundy, and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study," *Inf. Softw. Technol.*, vol. 85, pp. 60–70, May 2017, doi: 10.1016/j.infsof.2017.01.007.
[28] S. D. Conte, H. E. Dunsmore, and V. Y. Shen, *Software Engineering Metrics and Models*. Redwood City, CA, USA: Benjamin-Cummings, 1986.
[29] D. Port and M. Korte, "Comparative studies of the model evaluation criterions MMRE and PRED in software cost estimation research," in *Proc. 2nd ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, Oct. 2008, pp. 51–60, doi: 10.1145/1414004.1414015.
[30] E. D. Canedo, D. P. Aranha, M. D. O. Cardoso, R. P. D. Costa, and L. L. Leite, "Methods for estimating agile software projects: A systematic review," in *Proc. 30th Int. Conf. Softw. Eng. Knowl. Eng. (SEKE)*, Jul. 2018, pp. 34–39, doi: 10.18293/SEKE2018-031.
[31] A. A. Mohammed, A. Ahmad, and M. Omar, "Improvement of agile software development size & effort estimation methods," *Int. J. Innov. Technol. Exploring Eng.*, vol. 8, no. 8, pp. 357–362, Jun. 2019.
[32] J. P. T. Higgins and S. Green, *Cochrane Handbook for Systematic Reviews of Interventions, Version 5.1.0*. London, U.K.: The Cochrane Collaboration, Mar. 2011.
[33] V. Nepomuceno and S. Soares, "On the need to update systematic literature reviews," *Inf. Softw. Technol.*, vol. 109, pp. 40–42, May 2019, doi: 10.1016/j.infsof.2019.01.005.
[34] F. Q. B. da Silva, A. L. M. Santos, S. Soares, A. C. C. França, C. V. F. Monteiro, and F. F. Maciel, "Six years of systematic literature reviews in software engineering: An updated tertiary study," *Inf. Softw. Technol.*, vol. 53, no. 9, pp. 899–913, Sep. 2011.
[35] D. Moher, A. Tsertsvadze, A. Tricco, M. Eccles, J. Grimshaw, M. Sampson, and N. Barrowman, "When and how to update systematic reviews," *Cochrane Database Systematic Rev.*, no. 1, pp. 1–22, Jan. 2008, Art. no. MR000023, doi: 10.1002/14651858.MR000023.pub3.
[36] P. G. Shekelle, *et al.*, *Identifying Signals for Updating Systematic Reviews: A Comparison of Two Methods*. Rockville, MD, USA: Agency for Healthcare Research and Quality, 2011.
[37] B. A. Kitchenham, O. P. Brereton, and D. Budgen, "Protocol for extending a tertiary study of systematic literature reviews in software engineering," EPIC, Keele, U.K., Tech. Rep. EBSE-2008-006, Jun. 2008.
[38] K. R. Felizardo, E. Mendes, M. Kalinowski, E. F. Souza, and N. L. Vijaykumar, "Using forward snowballing to update systematic reviews in software engineering," in *Proc. 10th ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, 2016, pp. 53:1–53:6, doi: 10.1145/2961111.2962630.
[39] C. Wohlin, "Second-generation systematic literature studies using snowballing," in *Proc. 20th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2016, pp. 15:1–15:6, doi: 10.1145/2915970.2916006.
[40] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-Based Software Engineering and Systematic Reviews*, 1st Boca Raton, FL, USA: CRC Press, 2015.
[41] M. Petticrew and H. Roberts, "Systematic reviews in the social sciences: A practical guide," *Counselling Psychotherapy Res.*, vol. 6, no. 4, pp. 304–305, Nov. 2006.
[42] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Eng.*, vol. 14, no. 2, p. 131, Apr. 2009, doi: 10.1007/s10664-008-9102-8.
[43] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proc. 18th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2014, pp. 38:1–38:10, doi: 10.1145/2601248.2601268.
[44] S. K. Sehra, Y. S. Brar, N. Kaur, and S. S. Sehra, "Research patterns and trends in software effort estimation," *Inf. Softw. Technol.*, vol. 91, pp. 1–21, Nov. 2017, doi: 10.1016/j.infsof.2017.06.002.
[45] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, "Systematic literature review of machine learning based software development effort estimation models," *Inf. Softw. Technol.*, vol. 54, no. 1, pp. 41–59, Jan. 2012, doi: 10.1016/j.infsof.2011.09.002.
[46] D. R. Amancio, C. H. Comin, D. Casanova, G. Travieso, O. M. Bruno, F. A. Rodrigues, and L. da Fontoura Costa, "A systematic comparison of supervised classifiers," *PLoS ONE*, vol. 9, no. 4, Apr. 2014, Art. no. e94137, doi: 10.1371/journal.pone.0094137.
[47] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th ed. San Mateo, CA, USA: Morgan Kaufmann, 2016.
[48] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Proc. Conf. Emerg. Artif. Intell. Appl. Comput. Eng. Real Word AI Syst. Appl. eHealth, HCI, Inf. Retr. Pervasive Technol.*, 2007, pp. 3–24.
[49] M. Shepperd and S. MacDonell, "Evaluating prediction systems in software project estimation," *Inf. Softw. Technol.*, vol. 54, no. 8, pp. 820–827, Aug. 2012.
[50] S. K. T. Ziauddin and S. Zia, "An effort estimation model for agile software development," *Adv. Comput. Sci. Appl.*, vol. 2, no. 1, pp. 314–324, 2012.
[51] L. Lavazza and R. Meli, "An evaluation of simple function point as a replacement of IFPUG function point," in *Proc. Joint Conf. Int. Workshop Softw. Meas. Int. Conf. Softw. Process Product Meas.*, Oct. 2014, pp. 196–206, doi: 10.1109/IWSM.Mensura.2014.28.
[52] L. Santillo, M. Conte, and R. Meli, "Early & quick function point: Sizing more with less," in *Proc. 11th IEEE Int. Softw. Metrics Symp.*, Sep. 2005, pp. 1–6, doi: 10.1109/METRICS.2005.18.
[53] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative Res. Psychol.*, vol. 3, no. 2, pp. 77–101, 2006, doi: 10.1191/1478088706qp063oa.
[54] I. Hussain, L. Kosseim, and O. Ormandjieva, "Approximation of COSMIC functional size to support early effort estimation in agile," *Data Knowl. Eng.*, vol. 85, pp. 2–14, May 2013, doi: 10.1016/j.datak.2012.06.005.
[55] S. Ambler. (2005). *The Agile Unified Process (AUP)*. [Online]. Available: http://www.ambysoft.com/unifiedprocess/agileUP.html
[56] H. Kniberg and M. Skarin, *Kanban and Scrum—Making The Most of Both*. Lexington, KN, USA: C4Media Inc., 2010.
[57] E. Brechner, *Agile Project Management With Kanban*. Redmond, WA, USA: Microsoft Press, 2015.
[58] S. V. Shrivastava and H. Date, "Distributed agile software development: A review," *J. Comput. Sci. Eng.*, vol. 1, no. 1, pp. 10–17, Jun. 2010.
[59] M. Cohn, *Agile Estimating Planning*. London, U.K.: Pearson, 2005.
[60] M. Usman, E. Mendes, and J. Börstler, "Effort estimation in agile software development: A survey on the state of the practice," in *Proc. 19th Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2015, pp. 12:1–12:10, doi: 10.1145/2745802.2745813.
[61] M. Usman, J. Börstler, and K. Petersen, "An effort estimation taxonomy for agile software development," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 27, no. 04, pp. 641–674, May 2017, doi: 10.1142/S0218194017500243.

MARTA FERNÁNDEZ-DIEGO received the European Ph.D. degree in electronics and telecommunications engineering from the Lille University of Science and Technology, France, in 2001. She was, for several years, a member of a Software Development Team for mobile phone applications at an international information technology service company. She is currently an Associate Professor with the Department of Business Organisation, Universitat Politècnica de València, Spain, where she teaches at the School of Computer Science. Her research interests include empirical software engineering, software effort estimation, and project risk management.

ERWIN R. MÉNDEZ received the B.Sc. degree (Hons.) from APEC University, Dominican Republic, in 2015, and the master's degree in project management from the Universitat Politècnica de València, Spain, in 2018. He is currently pursuing the master's degree in business analytics and big data with the IMF Business School, Madrid, Spain. He is an Information Systems Engineer with the IMF Business School. Over the past years, he has been involved in software development projects, in the academic and industrial world, working as a Business and Data Analyst for companies in a variety of fields, including health insurance, social security, environmental management, logistics, and distribution. His research interests include machine learning data mining, process automation, and agile software development.

FERNANDO GONZÁLEZ-LADRÓN-DE-GUEVARA received the Ph.D. degree in industrial engineering, in 2001. He has worked at several universities and IT companies across Europe and Latin America. He is currently an Associate Professor with the Telecommunications Engineering School, Universitat Politècnica de València (UPV). He has coauthored several articles published in well-known international journals. He regularly participates in the organising committees of several national and international conferences. His research interests include crowdsourcing, ERP systems, and software engineering. He has participated in 27 research projects and contracts with different organisations, and was responsible for seven of them.

SILVIA ABRAHÃO (Member, IEEE) received the Ph.D. degree in computer science from the Universitat Politècnica de València (UPV), Spain, in 2004. She was a Visiting Professor with the Software Engineering Institute, Carnegie Mellon University, in 2010 and 2012, the Université Catholique de Louvain, in 2007 and 2017, and Ghent University, in 2004. She is currently an Associate Professor with UPV. She has coauthored over 150 peer-reviewed publications. She leads the Spanish Network of Excellence on Software Quality and Sustainability. Her main research interests include quality assurance in model-driven engineering, the empirical assessment of software modeling approaches, the integration of usability into software development, and cloud services monitoring and adaptation. She is a member of the Editorial Board of the *Software and Systems Modeling* (SoSyM) Journal. She is an Associate Editor of the IEEE Software, where she is responsible for *Software Quality*.

EMILIO INSFRAN (Member, IEEE) received the M.Sc. degree in computer science from the University of Cantabria, Spain, in 1994, and the Ph.D. degree from the Universitat Politècnica de València (UPV), Spain, in 2003. He worked as a Visiting Researcher at the Université Catholique de Louvain, Belgium, in 2017, and the Software Engineering Institute (SEI), CMU, USA, in 2012, and also performed research stays at the University of Twente, The Netherlands, Brigham Young University, UT, USA, and the University of Porto, Porto Alegre, Brazil. He is currently an Associate Professor with the Department of Information Systems and Computation (DISC), UPV. His research interests include cloud service architectures, DevOps, model-driven development, requirements engineering, and software quality. He has had more than 140 journal articles and conference papers published and worked on a number of national and international research projects and on several technology and transfer projects with companies.