Effort Estimation in Agile Software Development: A Systematic Literature Review

Muhammad Usman[1], Emilia Mendes[1], Francila Weidt[2], Ricardo Britto[1]
[1]Department of Software Engineering, Blekinge Institute of Technology, Karlskrona, Sweden
[2]Federal University of Juiz de Fora, Brazil
{[1]muhammad.usman, [1]emilia.mendes, [1]ricardo.britto}@bth.se, fran.weidt@gmail.com

ABSTRACT

Context: Ever since the emergence of agile methodologies in 2001, many software companies have shifted to Agile Software Development (ASD), and since then many studies have been conducted to investigate effort estimation within such context; however to date there is no single study that presents a detailed overview of the state of the art in effort estimation for ASD. **Objectives:** The aim of this study is to provide a detailed overview of the state of the art in the area of effort estimation in ASD. **Method:** To report the state of the art, we conducted a systematic literature review in accordance with the guidelines proposed in the evidence-based software engineering literature. **Results:** A total of 25 primary studies were selected; the main findings are: i) Subjective estimation techniques (e.g. expert judgment, planning poker, use case points estimation method) are the most frequently applied in an agile context; ii) Use case points and story points are the most frequently used size metrics respectively; iii) MMRE (Mean Magnitude of Relative Error) and MRE (Magnitude of Relative Error) are the most frequently used accuracy metrics; iv) team skills, prior experience and task size are cited as the three important cost drivers for effort estimation in ASD; and v) Extreme Programming (XP) and SCRUM are the only two agile methods that are identified in the primary studies. **Conclusion:** Subjective estimation techniques, e.g. expert judgment-based techniques, planning poker or the use case points method, are the one used the most in agile effort estimation studies. As for the size metrics, the ones that were used the most in the primary studies were story points and use case points. Several research gaps were identified, relating to the agile methods, size metrics and cost drivers, thus suggesting numerous possible avenues for future work.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – *Time Estimation, Cost Estimation, Software Process Models*

General Terms

Measurement, Management

Keywords

Effort Estimation, Agile Software Development, Systematic Literature Review

1. INTRODUCTION

Effort estimation is an integral part of software project management. There is a plethora of research in effort estimation in the form of models, techniques, methods, and tools. Different reviews of such literature are detailed in [1-7]. Effort estimation, like other software engineering activities, is performed under the umbrella of a development process. We have different types of development processes, e.g. traditional, agile etc., each having a different perspective on planning and estimation. During the past 10 years or so, there has been lot of interest and acceptability of agile processes, leading to the formal use of the term Agile Software Development (ASD). Under an ASD process, software is developed incrementally in small iterations and customer feedback serves as an important input for subsequent iterations. This also implies that estimations and plans need to be done progressively. This is achieved in ASD by planning iteratively at three levels i.e. release planning, iteration planning and the current day planning [8]. Given the high value of iterations in ASD, Planning and estimation are performed differently from traditional software development [8]. Some of the common techniques for estimation in ASD are Expert Opinion, Analogy and Disaggregation [8]. Planning Poker, introduced by Grenning in [9], is another technique often used for estimation in agile; it combines the elements of expert opinion, analogy and disaggregation [8]. Effort estimation in ASD is an active research area and in the subsequent paragraphs we briefly discuss some selected studies on effort estimation in ASD.

Planning poker has been used in a few agile effort estimation studies. Haugen [10] investigated its usefulness to estimate effort for an XP team during release planning. Results indicated planning poker produced more accurate estimates than unstructured group estimates when the team had related experience with similar projects. Two other studies [11, 12], conducted in a Scrum environment, compared group consensus estimates calculated by planning poker and statistical combination of individual expert estimates. Results showed that group consensus estimates were less optimistic and more accurate than statistical combination of individual estimates. Results of another study [13] also conducted in a Scrum environment do not fully support the results reported in [11, 12]. However, it was observed in [13] that optimism bias and estimation accuracy can be improved when planning poker is used by experienced practitioners.

Besides planning poker, other estimation techniques have also been investigated for ASD. A detailed incremental prediction model is proposed in [14] for effort prediction in agile and iterative development environments. This model was compared with a traditional global prediction model in an Extreme Programming (XP) environment, and results showed that the new incremental model outperformed the traditional estimation model in the early phases of development. Parvez [15] modified the use case point estimation method by inserting a new layer consisting of efficiency and risk factors, in order to estimate testing effort in of an agile project. The modified method was applied on four real projects to estimate the testing effort and presented more accurate estimates than the original use case method. A Constructive Agile Estimation Algorithm (CAEA) is proposed in [16] to incorporate factors believed vital to accurately investigate the cost, size and duration of a project.

The abovementioned studies are a sample of the range of studies that have been published on effort estimation in ASD. However, despite the current number of studies in such context, their evidence has not yet been aggregated and synthesized in a single place. Estimation reviews, such as [1-7] did not review the literature from an ASD perspective. In addition, although there are a number of reviews that provide details on agile studies [17-21], these reviews do not aggregate the evidence in the literature from an effort estimation perspective. To the best of our knowledge there is no systematic review, which reviews the literature from both perspectives together, i.e. effort estimation and agile software development. Therefore the goal and main contribution of this paper is to report the state of the art in the field of effort estimation in ASD by means of a Systematic Literature Review (SLR). We followed the guidelines proposed by Kitchenham and Charters [22]. This paper details our SLR and also points out the gaps and future directions for research in effort estimation for ASD. Note that given that we already knew about several studies in effort estimation in ASD and also that the evidence we wanted to gather from each study was quite detailed, we chose to carry out a SLR rather than a mapping study.

The remainder of the paper is organized as follows: Section 2 describes the steps performed in this SLR; the SLR results are presented in section 3 and discussed in section 4. Finally, conclusions and directions for the future research are given in section 5.

2. SYSTEMATIC LITERATURE REVIEW

A systematic literature review is conducted methodically by following a set of guidelines to collect and analyze all available evidence about a specific question in an unbiased and repeatable manner [22]. We started the process by defining the protocol that was designed and pilot tested by the first author (student) and reviewed by the second author (supervisor). We now describe steps performed during this SLR.

2.1 Research Questions

Following research questions were framed to guide the SLR:

- **RQ1**: What techniques have been used for effort or size estimation in agile software development?
  - RQ1a: What metrics have been used to measure estimation accuracy of these techniques?
  - RQ1b: What accuracy level has been achieved by these techniques?
- **RQ2**: What effort predictors (size metrics, cost drivers) have been used in studies on effort estimation for agile software development?
- **RQ3**: What are the characteristics of the dataset/knowledge used in studies on size or effort estimation for agile software development?
- **RQ4**: Which agile methods have been investigated in studies on size or effort estimation?
  - RQ4a: Which development activities (e.g. coding, design) have been investigated?
  - RQ4b: Which planning levels (release, iteration, current day) have been investigated?

Based on the research questions, PICOC [22] is described below:

Population (P): Software projects and applications developed using any of the agile software development methods
Intervention (I): Estimation methods/techniques/models/size metrics/cost-drivers.
Comparison (C): No comparison intervention
Outcomes (O): Accuracy of estimation techniques for ASD.
Context(C): ASD.

2.2 Search Strategy

A search strategy starts with the identification of major key terms from PICOC and their alternatives and synonyms. Theses terms are used to form a query string that is used to derive the rest of the search process.

2.2.1 Query String

The formation of a query string is an iterative process. Initially we followed the SLR guidelines [22] to create an initial string using Boolean OR/AND operators (OR is used to incorporate all synonyms and alternate spellings of each term and then these terms are ANDed to form one string). We piloted the initial string on search engines such as Scopus, IEEE explore, and Science Direct to check whether the string retrieved relevant primary studies and the studies we already knew about. Keywords from known primary studies and newly fetched ones were also included if not already part of the string. We also studied the titles, abstracts and author keywords from some already known primary studies to identify search terms. Table 1 lists the keywords used by some primary studies on effort estimation in ASD.

Table 1: Keywords from known primary studies

#	Keywords	References
1	Estimat*(Estimation, Estimating, Estimate)	[10, 11, 13, 14, 25-39]
2	Predict* (Prediction, Predicting)	[14, 40-42]
3	Forecasting	[13]
4	Effort	[13, 14, 25, 27, 30, 31, 35, 37, 38, 40-42]
5	Cost	[25, 33, 35-37]
6	Size, Sizing	[26, 31, 32, 35, 43, 44]
7	Velocity	[26, 28, 33, 40]
8	User Story	[10, 13, 29, 31, 33, 39, 42, 44]
9	Metrics	[28, 34, 35, 41, 43, 45]
10	Measur* (measure, measurement, measuring)	[11, 30, 32, 34, 35, 41, 43, 45]
11	Agile Software Development	[13, 27, 31, 33, 34, 38, 41, 45]
12	Agile Methods	[26, 32, 42]
13	Agile Processes	[43]

14	Agile Development	[33, 36, 39, 43]
15	Agile Estimation, Agile Estimating	[26, 27, 30, 44]
16	Extreme Programming	[10, 14, 35, 37, 40]
17	Scrum	[13, 25, 28, 30, 36]
18	Agile Project Management	[29, 31]
19	Agile Projects	[29, 32]
20	Agile Teams	[34]
21	Agile Methodology	[35, 39]
22	Agile Planning	[39]

The term "agile software development" has a large number of synonyms and alternate terms that are used in literature; some are listed in table 1 above. Given that as we added more studies to our set of known primary studies more alternate terms for agile software development were found, we decided to simply use one word (i.e. "agile") to cater for all of its possible alternate terms and ANDed it with the word "software" to filter out completely irrelevant studies from other domains. Dybå and Dingsoyr in their SLR [17] on agile software development have also used a similar approach for the term "agile". Another SLR on usability in agile software development by Silva et.al [21] also uses the term "agile" in the search string instead of trying to add all of its alternatives. In addition, the set of known primary studies was also used as a quasi-gold standard as suggested in [46] to assess the accuracy of the search string. The final search string we used is presented below. Note that this string had to be customized accordingly for each of the databases we used.

(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)*

2.2.2 Primary and Secondary Search Strategies

As part of the primary search strategy, search strings were applied on different databases in the first week of December 2013 to fetch the primary studies since 2001 (we chose 2001 as the starting date because this is when the Agile Manifesto[1] was published). Databases, search results, before and after removing duplicates, are listed in table 2. A master library was formed in Endnote[2] wherein results from all databases were merged and duplicates were removed. In the end, after removal of duplicates, we ended up with 443 candidate primary studies.

Table 2: Search Results

Database	Before removing Duplicates	After removing duplicates
Scopus	302	177
IEEE Explore	181	154
EI Compendex	76	69
Web of Science	134	23
INSPEC	48	9
Science Direct	7	2
ACM DL	27	9
Springer Link	12	0
Total	**787**	**443**

The databases we used cover all major Software Engineering conferences and journals thus providing a comprehensive coverage of this SLR's topic. We also ensured that these databases would index all the venues where known primary studies in our topic had been published. Other SLRs, such as [6, 19, 47], have also used these databases for searching relevant primary studies.

The secondary search was performed in the next phase by going through references of all the primary studies retrieved in the first phase.

2.3 Study Selection Criteria

Inclusion and exclusion criteria were defined in the light of the SLR objectives and research questions.

2.3.1 Inclusion Criteria

It was decided that studies that:

a. Report effort or size estimation related (technique or model or metric or measurement or predictors) AND
b. Are based on any of the agile software development methods AND
c. Described in English AND
d. Are reported in peer reviewed workshop or conference or journal.
e. Are evidence-based (empirical studies)

Will be included as primary studies.

2.3.2 Study Exclusion Criteria

It was decided that studies that:

a. Are not about effort or size estimation OR
b. That are not conducted using any of the agile software development methods OR
c. Are not described in English OR
d. Are not published in a peer reviewed conference or journal or workshop OR
e. Are not empirically-based
f. Deal with software maintenance phase only
g. Deal with performance measurement only i.e. velocity measurement

Will be excluded.

2.4 Study Selection Process

The study selection process was performed in two phases, as follows:

a. Title and Abstract level screening: In the first phase the inclusion/exclusion criteria was applied to the title and abstracts of all candidate primary studies identified by applying search strategy. Three researchers (the first three authors) performed this step independently, on all 443-search results, to deal with the problem of researcher bias. To arrive at a consensus, two Skype meetings were arranged between these three researchers that resulted in 30 papers passing the inclusion criteria out of 443 search results, with another 13

[1] http://agilemanifesto.org/

[2] http://endnote.com/

marked as doubtful. It was decided that the first author would go through the introduction, background and conclusion sections of these 13 doubtful papers to resolve the doubts. After consultation between the first two authors, two out of these 13 doubtful papers eventually passed the inclusion criteria increasing the number of studies to 32 that would be screened in the next phase.

b. Full text level screening: In the second phase, the inclusion & exclusion criteria were applied on full text of 32 papers that passed phase 1. The first author performed this task for all 32 papers; the other authors performed the task on 10% (3 papers by each one) of the total number of papers. At the end results were compared and consensus was reached for all common papers via Skype and face-to-face meetings. Whenever we did not have access to the primary study due to database access restrictions, we emailed that paper's authors. However, two papers could not be obtained by any means.

2.5 Quality Assessment (QA)

The studies quality assessment checklist was customized based on the checklist suggestion provided in [22]. Other effort estimation SLRs [6, 7] have also customized their quality checklists based on the suggestions given in [22]. Each question in the QA checklist was answered using a three-point scale i.e. Yes (1 point), No (0 point) and Partial (0.5 point). Each study could obtain 0 to 13 points. We used the first quartile (13/4= 3.25) as the cutoff point for including a study, i.e., if a study scored less than 3.25 it would be removed from our final list of primary studies.

Table 3: Quality Assessment Checklist adopted from [7, 22]

Question	Score
1. Are the research aims clearly specified?	Y|N|P
2. Was the study designed to achieve these aims?	Y|N|P
3. Are the estimation techniques used clearly described and their selection justified?	Y|N|P
4. Are the variables considered by the study suitably measured?	Y|N|P
5. Are the data collection methods adequately described?	Y|N|P
6. Is the data collected adequately described?	Y|N|P
7. Is the purpose of the data analysis clear?	Y|N|P
8. Are statistical techniques used to analyze data adequately described and their use justified?	Y|N|P
9. Are negative results (if any) presented?	Y|N|P
10. Do the researchers discuss any problems with the validity/reliability of their results?	Y|N|P
11. Are all research questions answered adequately?	Y|N|P
12. How clear are the links between data, interpretation and conclusions?	Y|N|P
13. Are the findings based on multiple projects	Y|N|P

2.6 Data Extraction and Synthesis Strategies

A data extraction form was designed which, besides having general fields (year, conference, author etc.), contained fields corresponding to each research question (e.g. estimation technique used, size metrics, accuracy metrics, agile method etc.). Data extraction and quality assessment were both performed by the authors in accordance with the work division scheme described in section 2.4b. The extracted data were then synthesized to answer each of the research questions.

3. RESULTS

In this section we describe the results for the overall SLR process and for each of the research questions as well. Table 4 describes the number of studies passing through different stages of the SLR. Details of the excluded papers during the full text screening and the QA phases are: eight papers [29, 40, 44, 48-52] were excluded due to not passing the inclusion criteria, two due to a low quality score [25, 53] and one [54] due to reporting a study already included in another paper [14]. Breakup of the eight papers excluded on inclusion & exclusion criteria is as under:

- 2 studies were not empirically based (i.e. they met exclusion criterion e)
- 3 studies were not conducted in an agile software development context (exclusion criterion b)
- 3 studies met the last exclusion criterion i.e. velocity measurement.

Table 4: No. Of Papers in Study Selection and QA

Database	Search
a. Search results	443
b. After titles and abstracts screening	32
c. Inaccessible papers	2
d. Excluded on inclusion exclusion criteria	8
e. Duplicate study	1
f. Excluded on low quality score	2
g. Final Papers (b-c-d-e-f)	19

As part of the secondary search, references of 19 primary studies identified in the primary search process were screened to identify any additional relevant papers. Six papers passed this initial screening in the secondary search, out of which four were excluded on inclusion & exclusion criteria applied on the full text, one for having a low quality score, leaving only one additional paper, which increased the total number of primary studies to 20. Some of the papers reported more than one study, as follows: one paper reported four studies; three papers reported two studies (one study is common in two of these three papers and is counted only once). Therefore, although we had 20 papers, they accounted for 25 primary studies. We present this SLR's results arranged by the 25 primary studies reported in 20 primary papers.

3.1 RQ1: Estimation Techniques

Four studies, out of the 25 primary studies, investigated techniques for estimating size only (S1, S7, S11, S17); three did not describe the estimation technique used (S5, S14) while the remaining 19 studies used effort estimation techniques. Table 5 lists the different estimation techniques (both size and effort) along with their corresponding frequencies (no of studies using the technique) denoted as F in last column. Size estimation techniques are grouped with effort estimation techniques because within an agile context an effort estimate is often derived from a size estimate and velocity calculations [8]. Expert judgment, planning poker and use case points method (both original and modifications) are estimation techniques frequently used in an ASD context. All of these frequently used effort estimation techniques require a subjective assessment by the experts, which suggests that, in a similar way as currently occurs within non-agile software development contexts, within an agile context the estimation techniques that take into account experts' subjective assessment are well received. Note that table 5 also shows that other types of effort estimation techniques have also been investigated e.g. regression, neural networks. Column F does not add up to 25 as many studies have used more than one technique.

Table 5: Estimation Techniques Investigated

Technique	Study ID	F
Expert Judgment	S2a, S2b, S4a, S4b, S4c, S4d, S6, S20	8
Planning Poker	S1, S6, S8, S16a, S16b, S17	6
Use Case Points (UCP) Method	S4a, S4b, S4c, S4d, S10, S12	6
UCP Method Modification	4a, 4b, 4c, 4d, 11	5
Linear Regression, Robust Regression, Neural Nets (RBF)	S2a, S2b, S3	3
Neural Nets (SVM)	S2a, S2b	2
Constructive Agile Estimation Algorithm	S13, S18	2
Other	Statistical combination of individual estimates (S8), COSMIC FP (S7), Kalman Filter Algorithm (S9), Analogy (S15), Silent Grouping (S1), Simulation (S19)	6
Not Described	S5, S14	2

3.1.1 RQ1a: Accuracy Metrics

Question 1b investigates which prediction accuracy metrics were used by the primary studies. The entries shown in table 6 are based solely on the 23 primary studies (see table 5) that used at least one estimation technique. Both the Mean Magnitude of Relative Error (MMRE) and the Magnitude of Relative Error (MRE) are the most frequently used accuracy measures. The use of MMRE is in line with the findings from other effort estimation SLRs[1,7]. Two important observations were identified herein: the metric Prediction at n% (Pred(n)) was used in two studies only and secondly 26% (6 out of 23 studies included in table 6) of the studies that employed some estimation technique have not measured their prediction accuracy. Finally, some studies have used other accuracy metrics, described in the category 'other' in table 6. Two of which (S5, S8) employed the Balanced Relative Error (BRE) metric on the grounds that BRE can more evenly balance overestimation and underestimation, when compared to MRE.

Table 6: Accuracy metrics used

Metrics	Study ID	F
MMRE	S2a, S2b, S6, S9, S14	5
MRE	S3, S4a, S4b, S4c, S4d	5
Pred(25)	S2a, S2b	2
Not Used	S1, S10, S13, S17, S18, S20	6
Other	S11(SD, Variance, F test), S12(comparison with actual), S16a and S16b (Box plots to compare effort estimates), S8(BRE, REbias),S5(BREbias),S6(MdMRE),S2a and S2b(MMER), S7 (Kappa index)	10

3.1.2 Accuracy Level Achieved

Question 1b looks into the accuracy levels achieved by different techniques. Due to the space limitations, table 7 only includes those techniques that have been applied in more than one study and for which the accuracy level was reported. Other than Studies 4a to 4d that report good MRE values for UCP methods (modified and original) and expert judgment, no other technique has achieved accuracy level of at least 25%[55]. Studies 4a to 4d are all reported in a single paper and focus solely on testing effort, thus clearly prompting the need for further investigation. The lack of studies measuring prediction accuracy and presenting good accuracy represents a clear gap in this field.

Table 7: Accuracy achieved by frequently used techniques

Technique	Accuracy Achieved %(Study IDs)
Planning Poker	MMRE: 48.0 (S6) Mean BRE: 42.9-110.7 (S8)
Use Case Points (UCP) Method	MRE: 10.0-21.0 (S4a to S4d)
UCP Method Modification	MRE: 2.0-11.0 (S4a to S4d)
Expert Judgment	MRE: 8.00-32.00 (S4a to S4d) MMRE: 28-38 (S2a, S2b, S6) Pred (25%): 23.0-51.0 (S2a, S2b)
Linear Regression	MMRE: 66.0-90.0 (S2a, S2b) Pred(25): 17.0-31.0 (S2a, S2b) MMER: 47.0-67.0 (S3) MRE: 11.0-57.0 (S3)
Neural Nets (RBF)	MRE: 6.00-90.00 (S3)

3.2 RQ2: Effort Predictors

Question 2 focuses on the effort predictors that have been used for effort estimation in an ASD context.

3.2.1 Size Metrics

Table 8 presents the size metrics that have been used in the primary studies focus of this SLR. Use case and story points are the most frequently used size metrics; very few studies used more traditional size metrics such as function points and lines of code, which in a way is not a surprise given our findings suggest that whenever the requirements are given in the form of stories or use case scenarios, story points and UC points are the choices commonly selected in combination with estimation techniques such as planning poker and UCP method. Finally, to our surprise, 20% of the studies (five) investigated herein did not describe any size metrics used during the study.

Table 8: Size Metrics

Metrics	Study IDs	F
Story Points	S1, S2a, S8, S13, S17, S18	6
Use Case Points	S4a, S4b, S4c, S4d, S10, S11, S12	7
Function Points	S9, S10, S17	3
Lines of Code	S3	1
Other	Number of User Stories (S19), COSMIC FP (S7), Length of User Story (S2b)	3
Not Described	S5, S6, S15, S16a, S16b	5

3.2.2 Cost Drivers

Question 2b investigates what cost drivers have been used by effort estimation studies in ASD context. Table 9 shows that different cost drivers have been used by the primary studies investigated herein, and without a common pattern. 24% of the studies (6 out of 25) used a cost driver that was not used by any other primary study. Two testing effort-related drivers are used in four studies (S4a to S4b), all reported in the same paper. Development team skills, size of the task to be estimated and teams' prior experience are reported in multiple studies. In the ASD context, where most of the knowledge is not explicit, skills and experience of the developers seems to play a crucial role in all tasks, including effort estimation. Note that 32% (9) of the studies detailed herein did not use any cost drivers for effort estimation, thus also suggesting a research gap and a possible avenue for future research.

Table 9: Cost Drivers

Cost Drivers	Study IDs	F
Dev. Team Skills/Expertise	S15, S16a, S16b	3
Task Size	S6, S11	2
Team's Prior Experience	S6, S15	2
Test Efficiency Factor, Test Risk Factor	S4a, S4b, S4c, S4c	4
Project domain, Performance requirements, Configuration requirements, Volume of data transactions, Complex processing, Operational ease for users, Multiple sites, Security	S13, S18	2
Other	S19 (12 XP Practices), S2a & S2b (Keywords in user story), S3 (Chidamber & Kemerer design metrics), S5 (Customer Communication Frequency), S11 (Use case Elements)	6
Not Investigated	S3, S7, S8, S9, S10, S12, S14, S17, S20	9

3.3 RQ3: Characteristics of the Dataset or Knowledge Used

RQ3 looks into the characteristics, i.e. domain (industrial or academic or both) and type (within-company or cross-company), of the datasets or knowledge used in primary studies. Table 10 shows that 76% (19) of the studies used an industrial dataset. This is a good sign as use of the industrial data increases the usefulness of such results for practitioners.

Table 10: Domain of the data used

Domain	Study IDs	F
Industrial	S1, S2a, S2b, S3, S4a-S4d, S5, S6, S10, S12, S13, S14, S15,S16b,S17, S18, S20	19
Academic	S8, S11, S16a	3
Other	S7 (Both), S19 (Not Described), S9 (Simulated)	3

Table 11 details the type of dataset used herein; 72% (18) used within-company data, only 8%(2) used cross-company data, and one study (S9) used log-simulated data to predict completion time of project logs. Another 16%(4) did not describe the type of dataset used. We believe these results are quite interesting as they suggest that within the scope of ASD, companies have focused on their own project data, rather than looking for data from cross-company datasets. There are contexts (e.g. Web development) where estimation models based on within-company data have clearly presented superior accuracy, when compared to models built from cross-company datasets[6]; however, the lack of primary studies herein using cross-company datasets does not necessarily mean that such datasets would not be useful within an ASD context; perhaps they have not been used because most of the publicly available datasets in software engineering contain only data on legacy systems[3]. We believe this is also a clear research gap that provides further avenues for future research in effort estimation for ASD context.

Table 11: Type of datasets used

Type	Study IDs	F
Within Company	S2a, S2b, S3, S4a-S4d, S5, S6, S10, S11, S12, S13, S14, S15, S17, S18, S20	18
Cross Company	S1, S7	2
Not Described	S8, S16a, S16b, S19	4
Log Simulated	S9	1

3.4 RQ4: Agile Methods Used

This research question is designed to identify the specific agile methods used in effort estimation studies in an ASD context. Table 12 details the methods used in each of the studies, and corresponding frequency of method usage. Scrum and XP were the most frequently used agile methods in studies on effort estimation. However it is unfortunate to note that 40% (10) of the studies only mention that they are using an agile method without specifying the exact agile method used.

Table 12: Agile Method Used

Agile Method	Study IDs	F
XP	S2a, S2b, S3, S6, S14, S19, S20	7
Scrum	S1, S4a-S4d, S8, S12, S17	8
Not Described	S5, S7, S9, S10, S11, S13, S15, S16a, S16b, S18	10

3.4.1 RQ4a: Development Activity

Question 4a investigates the development activities to which the effort estimate applies. None of the primary studies estimated effort for analysis and design activities only; 16% of the primary studies (four studies reported in one paper) investigated testing effort estimation, 12%(3) investigated implementation effort and another 12% (3) investigated both implementation and testing effort. Close to a third of the primary studies in this SLR (9) estimated effort considering all of the development activities (analysis, design, implementation and testing activities) of an agile project.

[3] https://code.google.com/p/promisedata/

Table 13: Development Activity Investigated

Dev. Activity	Study IDs	F
Analysis	None	0
Design	None	0
Implementation (I)	S2a, S2b, S6	3
Testing (T)	S4a-S4d	4
I & T	S8, S16a, S16b	3
All	S3, S5, S9, S10, S11, S12, S13, S18, S19	9
Not Described	S1, S7, S14, S15, S17, S20	6

3.4.2 *RQ4b: Planning Level*

Planning can be performed at three different levels in an agile context i.e. release, iteration and daily planning [8]. This question looks into which planning levels were used in this SLR's primary studies. Table 14 details the results showing the studies and their count (denoted as F in last column) against each planning level. A quarter of the studies (6) dealt with iteration level planning; and release planning is investigated in 20%(5) of the studies. Both iteration and release planning levels have received almost equal coverage indicating the importance of planning at both levels; however, to our surprise, 48% (12) of the primary studies did not describe the planning level they were dealing with.

Table 14: Development Activity Investigated

Planning Level	Study IDs	F
Daily	None	
Iteration (I)	S2a, S2b, S3, S12, S14, S17	6
Release (R)	S1, S6, S8, S9, S20	5
Both I and R	S13, S18	2
Not Described	S4a-S4d, S5, S7, S10, S11, S15, S16a, S16b, S19	12

4. DISCUSSION

The results of this SLR address four research questions (see Section 2.1), which cover the following aspects:

- Techniques for effort or size estimation in an Agile Software Development (ASD) context.
- Effort predictors used in estimation studies in ASD context.
- Characteristics of the datasets used.
- Agile methods, activities and planning levels investigated.

We observed that a variety of estimation techniques have been investigated since 2001, ranging from expert judgment to neural networks. These techniques used different accuracy metrics (e.g. MMRE; Kappa index) to assess prediction accuracy, which in most cases did not turn out to meet the 25% threshold [55]. In addition, with regard to the effort predictors used in the primary studies, we see that there is little consensus on the use of cost drivers, i.e. only a few cost drivers are common across multiple studies. We believe that further investigation relating to effort predictors is warranted, as it seems that the focus of previous studies mostly on size metrics may have had an effect on the poor prediction accuracy reported. When combining the results and discussion on two questions, it can be said that much needs to be done at both technique and cost drivers level to improve effort estimation in ASD. Currently there are no guidelines for practitioners to decide as to which estimation technique or cost driver to choose in a particular agile method or context.

When it comes to the most frequently used estimation techniques, our results showed that the top four (expert judgment, planning poker, UCP method original and modified) have one common facet i.e. some form of subjective judgment by experts. Other techniques have also been investigated but not in this proportion. We can say that subjective estimation techniques are frequently used in agile projects.

With regard to prediction accuracy metrics, MMRE and MRE are the frequently used metrics; however two recent studies have recommended the use of BRE as accuracy metric on the basis that it balances over and under estimation better than MMRE. Some other alternate metrics have also been used e.g. boxplots (two studies) and MMER (two studies). Despite the existing criticism with regard to both MMRE and MRE for being inherently biased measures of accuracy (e.g. [56, 57]), they were the ones used the most within the context of this SLR. In our view this is a concern and such issues should be considered in future studies. It is surprising that 26%(6) of the studies that have applied an estimation technique did not measure that technique's prediction accuracy; in our view such studies may be of negligible use to researchers and practitioners as they failed to provide evidence on how suitable their effort estimation techniques are.

Our SLR results showed that story points and use case points were the most frequently used size metrics while traditional metrics (FP or LOC) were rarely used in agile projects. Both story points and use case points seem to synchronize well with the prevalent methods for specifying the requirements under agile methods i.e. user stories and use case scenarios. There appears to be a relationship between the way the requirements are specified (user stories, use cases), frequently used size metrics (story points, UC points) and frequently used estimation techniques (Planning poker, use case points methods). What is lacking is a consensus on the set of cost drivers that should be used in a particular agile context. In the absence of correct and appropriate cost drivers, accuracy will be compromised and we believe that this may be the reason for poor accuracy levels identified in this SLR. We believe much needs to be done in the cost drivers area.

Industrial datasets are used in most of the studies, which may be a good indication for the acceptability of the results. However, most of the data sets, i.e. 72% (18), are of within company type. This may be due the fact that cross-company datasets specifically for agile projects are not easily available. With the increasing number of companies going agile, we believe that some efforts should be made to make cross-company datasets available for ASD context in order to support and improve effort estimation.

XP and Scrum were the two most frequently used agile methods, and we have not found any estimation study that applied any other agile method other than these two. We do not know how estimation is performed in agile methods other than XP and Scrum, which points out the need to conduct estimation studies with other agile methods as well. With respect to the development activity being investigated, only implementation and testing have been investigated in isolation. About 36%(9) of the primary studies took into all activities during effort prediction, which may be due to the fact that in an ASD process, the design, implementation and testing activities are performed very closely to each other.

During this SLR, we found very few studies reporting all the required elements appropriately e.g. 40%(10) of the studies have not described the exact agile method used in the study, 24%(6) of them have not described the development activity that is subject to estimation, and 26%(6) of the studies have not used any accuracy metric. Quality checklists help in assessing the quality of the studies' evidence; however it proved challenging to assess the quality of all types of studies by means of a generic checklist. A potential threat to validity for this SLR is the issue of coverage i.e. Are we able to cover all primary studies? We have aimed to construct a comprehensive string, following the guidelines proposed in evidence based SE literature, which covers all relevant key terms and their synonyms. Search strings were applied to multiple databases that cover all major SE conferences and journals. Lastly we have also performed a secondary search in order to make sure that we did not miss any primary study. In regard to the quality of studies' selection and data extraction, we have also used a process in which a good percentage of studies (details in section 2.4a and 2.4b above) were selected and extracted separately by more than one researcher, and later aggregated via consensus meetings, in order to minimize any possible bias.

5. CONCLUSION

This study presents a systematic literature review on effort estimation in Agile Software Development (ASD). The primary search fetched 443 unique results, from which 32 papers were selected as potential primary papers. From these 32 papers, 19 were selected as primary papers on which the secondary search process was applied resulting in one more paper added to the list of included papers. Results of this SLR are based on 20 papers that report 25 primary studies. Data from these 25 studies were extracted and synthesized to answer each research question.

Although a variety of estimation techniques have been applied in an ASD context, ranging from expert judgment to Artificial Intelligence techniques, those used the most are the techniques based on some form of expert-based subjective assessment. These techniques are expert judgment, planning poker, and use case points method. Most of the techniques have not resulted in good prediction accuracy values. It was observed that there is little agreement on suitable cost drivers for ASD projects. Story points and use case points, which are close to prevalent requirement specifications methods (i.e. user stories, use cases), are the most frequently used size metrics. It was noted that most of the estimation studies used datasets containing data on within-company industrial projects. Finally, other than XP and Scrum, no other agile method was investigated in the estimation studies.

Practitioners would have little guidance from the current effort estimation literature in ASD wherein we have techniques with varying (and quite often low) level of accuracy (in some case no accuracy assessment reported at all) and with little consensus on appropriate cost drivers for different agile contexts. Based on the results of this SLR we recommend that there is strong need to conduct more effort estimation studies in ASD context that, besides size, also take into account other effort predictors. It is also suggested that estimation studies in ASD context should perform and report accuracy assessments in a transparent manner as per the recommendations in the software metrics and measurement literature. Finally, we also believe that further investigation into the use of cross-company datasets for agile effort estimation is needed given the possible benefits that the use of such datasets would perhaps bring to companies that employ ASD practices but that do not yet have their own datasets on software projects to use as basis for effort estimation.

6. REFERENCES

[1] Jorgensen, M. and Shepperd, M. A Systematic Review of Software Development Cost Estimation Studies. *Software Engineering, IEEE Transactions on*, 33, 1(Jan.2007), 33-53.

[2] Boehm, B., Abts, C. and Chulani, S. Software development cost estimation approaches-A survey. *Ann. Softw. Eng.*, 10, 1-4 (Nov. 2000), 177-205.

[3] Molokken, K. and Jorgensen, M. A review of software surveys on software effort estimation. *In Proceedings of 2003 International Symposium on Empirical Software Engineering(*Rome, Italy, 2003). 223-230.

[4] Jorgensen, M. A review of studies on expert estimation of software development effort. *J. Syst. Softw.*, 70, 1-2 (Feb. 2004), 37-60.

[5] Wen, J., Li, S., Lin, Z., Hu, Y. and Huang, C. Systematic literature review of machine learning based software development effort estimation models. *Inf. Softw. Technol.*, 54, 1 (Jan. 2012), 41-59.

[6] Kitchenham, B. A., Mendes, E. and Travassos, G. H. Cross versus Within-Company Cost Estimation Studies: A Systematic Review. *Software Engineering, IEEE Transactions on*, 33, 5 (May. 2007), 316-329.

[7] Azhar, D., Mendes, E. and Riddle, P. A systematic review of web resource estimation. In *proceedings of the 8th International Conference on Predictive Models in Software Engineering(Lund, Sweden,2012)*. 49-58.

[8] Cohn, M. *Agile estimating and planning*. Pearson Education, 2005.

[9] Grenning, J. Planning poker or how to avoid analysis paralysis while release planning. *Hawthorn Woods: Renaissance Software Consulting*, 3(2002).

[10] Haugen, N. C. An empirical study of using planning poker for user story estimation. *In Agile Conference* (Minneapolis, MN, 2006). 23-31.

[11] Molokken-Ostvold, K. and Haugen, N. C. Combining Estimates with Planning Poker--An Empirical Study. *In proceedings of 18th Australian Software Engineering Conference*(Melbourne, Australia, 2007). 349-358.

[12] Moløkken-Østvold, K., Haugen, N. C. and Benestad, H. C. Using planning poker for combining expert estimates in software projects. *Journal of Systems and Software*, 81, 12 (Dec. 2008), 2106-2117.

[13] Mahnič, V. and Hovelja, T. On using planning poker for estimating user stories. *Journal of Systems and Software*, 85, 9 (Sep. 2012), 2086-2095.

[14] Abrahamsson, P., Moser, R., Pedrycz, W., Sillitti, A. and Succi, G. Effort Prediction in Iterative Software Development Processes -- Incremental Versus Global Prediction Models. *In First International Symposium on Empirical Software Engineering and Measurement(* Madrid, Spain, 2007). 344-353.

[15] Parvez, A. W. M. M. Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development. In *Proceedings of International Conference on Information Technology and Electrical Engineering* (Yogyakarta, Indonesia, 2013).113-118.

[16] Bhalerao, S. and Ingle, M. Incorporating vital factors in agile estimation through algorithmic method. *International Journal of Computer Science and Applications*, 6, 1 (Jan. 2009), 85-97.

[17] Dybå, T. and Dingsøyr, T. Empirical studies of agile software development: A systematic review. *Information and Software Technology*, 50, 9–10 (Aug. 2008), 833-859.

[18] Hasnain, E. An overview of published agile studies: a systematic literature review. In *Proceedings of the Proceedings of the 2010 National Software Engineering Conference* (Rawalpindi, Pakistan, 2010). 1-6.
[19] Jalali, S. and Wohlin, C. Global software engineering and agile practices: a systematic review. *Journal of Software: Evolution and Process*, 24, 6 (Oct. 2012), 643-659.
[20] Panagiotis Sfetsos and Stamelos, I. Empirical Studies on Quality in Agile Practices: A Systematic Literature Review. In *Proceedings of the 2010 Seventh International Conference on the Quality of Information and Communications Technology* (Porto, Portugal, 2010). 44-53.
[21] Silva da Silva, T., Martin, A., Maurer, F. and Silveira, M. User-Centered Design and Agile Methods: A Systematic Review. In *Proceedings of the Agile Conference (AGILE)* (Salt Lake City,UT, 2011). 77-86.
[22] Kitchenham B. and Charters S. *Guidelines for performing systematic literature reviews in software engineering (version 2.3)*. Technical report, Keele University and University of Durham, 2007.
[23] Petersen, K., Feldt, R., Mujtaba, S. and Mattsson, M. Systematic mapping studies in software engineering. In *Proceedings of 12th International Conference on Evaluation and Assessment in Software Engineering*(Bari, Italy, 2008).71-80.
[24] Kitchenham, B. A., Budgenb, D. and Pearl Brereton. Using mapping studies as the basis for further research – A participant-observer case study. *Information and Software Technology*, 53, 6 (June. 2011), 638-651.
[25] Popli, R. and Chauhan, N. A sprint-point based estimation technique in Scrum. In *Proceedings of International conference on information Systems and Computer Networks* (Mathura, India, 2013). 98-103.
[26] Bhalerao, S. and Ingle, M. Incorporating Vital Factors in Agile Estimation through Algorithmic Method. *IJCSA*, 6, 1 (2009), 85-97.
[27] Tamrakar, R. and Jorgensen, M. Does the use of Fibonacci numbers in planning poker affect effort estimates? In *Proceedings of 16th International Conference on Evaluation and Assessment in Software Engineering*(Ciudad Real, Spain, 2012). 228-232.
[28] Downey, S. and Sutherland, J. Scrum Metrics for Hyperproductive Teams: How They Fly like Fighter Aircraft. In *46th Hawaii International Conference on System Sciences(*Wailea, HI, 2013). 4870-4878.
[29] Desharnais, J. M., Buglione, L. and Kocatürk, B. Using the COSMIC method to estimate Agile User Stories. In *Proceedings of the 12th Internationala Conference on product focused software development and process improvement* (Torre Canne, Italy, 2011). 68-73.
[30] Mahnic, V. A case study on agile estimating and planning using scrum. *Electronics and Electrical Engineering*, 111, 5 (2011).123-128.
[31] Desharnais, J., Buglione, L. and Kocaturk, B. Improving Agile Software Projects Planning Using the COSMIC Method. *In workshop on Managing Client Value Creation Process in Agile Projects (Torre Cane, Italy,*2011).
[32] Trudel, S. and Buglione, L. Guideline for sizing Agile projects with COSMIC. In *Proceedings of International Workshop on Software Measurement(* Stuggart,Germany, 2010).
[33] Sungjoo, K., Okjoo, C. and Jongmoon, B. Model Based Estimation and Tracking Method for Agile Software Project. *Intl. Journal of Human Capital and Information Technology Professionals*, 3, 2 (April.2012), 1-15.
[34] Taghi Javdani, G., Hazura, Z., Abdul Azim Abd, G. and Abu Bakar Md, S. On the Current Measurement Practices in Agile Software Development. *International Journal of Computer Science Issues*, 9, 4 (2012), 127-133.
[35] Choudhari, J. and Suman, U. Phase wise effort estimation for software maintenance: An extended SMEEM Model. ACM, In *Proceedings of International Information Technology Conference(*Pune, India, 2012).397-402.
[36] Nunes, N. J. iUCP - Estimating interaction design projects with enhanced use case Points. In *Proceedings of 8th International Workshop on Task Models and Diagrams for UI Design(*Brussels, Belgium, 2010).131-145.
[37] Wang, X.-h. and Zhao, M. An estimation method of iteration periods in XP project. *Journal of Computer Applications*, 27, 5 (2007), 1248-1250.
[38] Cao, L. Estimating agile software project effort: An empirical study. In *Proceedings of 14th Americas Conference on IS*(Toronto, Canada, 2008).1907-1916.
[39] Shi, Z., Chen, L. and Chen, T.-e. Agile planning and development methods. In *Proceedings of 3rd International Conference on Computer Research and Development (*Shanghai,China, 2011).488-491.
[40] Hearty, P., Fenton, N., Marquez, D. and Neil, M. Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model. *Software Engineering, IEEE Transactions on*, 35, 1 (Jan-Feb. 2009), 124-137.
[41] Kunz, M., Dumke, R. R. and Zenker, N. Software Metrics for Agile Software Development. *In proceedings of 19th Australian Software Engineering Conference*(Perth, Australia, 2008).673-678.
[42] Abrahamsson, P., Fronza, I., Moser, R., Vlasenko, J. and Pedrycz, W. Predicting Development Effort from User Stories. *In International Symposium on Empirical Software Engineering and Measurement* (Banff, AB, 2011).400-403.
[43] Aktunc, O. Entropy Metrics for Agile Development Processes. In *Proceedings of International Symposium on Software Reliability Engineering Workshops(*Dallas, TX, 2012).7-8.
[44] Miranda, E., Bourque, P. and Abran, A. Sizing user stories using paired comparisons. *Information and Software Technology*, 51, 9 (Sep. 2009), 1327-1337.
[45] Hartmann, D. and Dymond, R. Appropriate agile measurement: using metrics and diagnostics to deliver business value. In *Proceedings of Agile Conference (*Minneapolis, 2006).126-131.
[46] Zhang, H. and Babar, M. A. On searching relevant studies in software engineering. In *Proceedings of the 14th international conference on evaluation and assessment in software engineering (EASE)(*Keele,UK,2010).1-10.
[47] Schneider, S., Torkar, R. and Gorschek, T. Solutions in global software engineering: A systematic literature review. *International Journal of Information Management*, 33, 1 (Feb. 2013), 119-132.
[48] Alshayeb, M. and Li, W. An Empirical Validation of Object-Oriented Metrics in Two Different Iterative Software Processes. *IEEE Transactions on Software Engineering*, 29, 11(Nov. 2003), 1043-1049.
[49] Sungjoo, K., Okjoo, C. and Jongmoon, B. Model-Based Dynamic Cost Estimation and Tracking Method for Agile Software Development. In *Proceedings of 9th International Conference on Computer and Information Science (*Yamagata, Japan, 2010).743-748.

[50] Hohman, M. M. Estimating in actual time [extreme programming]. In *Proceedings of Agile Conference(*Denver, Colorado,2005).132-138.

[51] Pow-Sang, J. A. and Imbert, R. Effort estimation in incremental software development projects using function points. In *Proceedings of Advanced Software Engineering and its Applications*(Jeju Island, Korea, 2012).458-465.

[52] Farahneh, H. O. and Issa, A. A. A Linear Use Case Based Software Cost Estimation Model. *World Academy of Science, Engineering and Technology*, 32(2011), 504-508.

[53] Zang, J. Agile estimation with monte carlo simulation. In *Proceedings of XP 2008(*Limerick, Ireland, 2008).172-179.

[54] Moser, R., Pedrycz, W. and Succi, G. Incremental effort prediction models in agile development using radial basis functions. In *Proceedings of* 19th Intl. Conference on Software Engineering and Knowledge Engineering,(Boston, MA, 2007).519-522.

[55] Conte, S. D., Dunsmore, H. E., & Shen, V. Y.Software engineering metrics and models. Benjamin-Cummings Publishing Co., Inc. 1986.

[56] Kitchenham, B.A.; Pickard, L.M.; MacDonell, S.G.; Shepperd, M.J.What accuracy statistics really measure [software estimation]. Software, IEE Proceedings, 148,3(June 2001),81-85.

[57] Jørgensen, Magne.A critique of how we measure and interpret the accuracy of software development effort estimation.*In First International Workshop on Software Productivity Analysis and Cost Estimation.* Information Processing Society of Japan (Nagoya. 2007),15-22.

Appendix: List of Included Primary Studies

S1:Power, Ken. "Using Silent Grouping to Size User Stories." In 12th International Conference on Agile Processes in Software Engineering and Extreme Programming, XP 2011, May 10, 2011 - May 13, 2011, 77 LNBIP, 60-72. Madrid, Spain: Springer Verlag, 2011.

S2a, S2b: Abrahamsson, P., I. Fronza, R. Moser, J. Vlasenko and W. Pedrycz. "Predicting Development Effort from User Stories." In Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on, 400-403, 2011.

S3:Abrahamsson, P., R. Moser, W. Pedrycz, A. Sillitti and G. Succi. "Effort Prediction in Iterative Software Development Processes -- Incremental Versus Global Prediction Models." In Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on, 344-353, 2007.

S4a to S4d:Parvez, Abu Wahid Md Masud. "Efficiency Factor and Risk Factor Based User Case Point Test Effort Estimation Model Compatible with Agile Software Development." In Information Technology and Electrical Engineering (ICITEE), 2013 Intl. Conference on, 113-118, 2013.

S5:Molokken-Ostvold, K. and K. M. Furulund. "The Relationship between Customer Collaboration and Software Project Overruns." In Agile Conference (AGILE), 2007, 72-83, 2007.

S6:Haugen, N. C. "An Empirical Study of Using Planning Poker for User Story Estimation." In Agile Conference, 2006, pp.23-31, 2006.

S7:Hussain, I., L. Kosseim and O. Ormandjieva. "Approximation of Cosmic Functional Size to Support Early Effort Estimation in Agile." Data and Knowledge Engineering 85, (2013): 2-14.

S8:Mahnič, V. and T. Hovelja. "On Using Planning Poker for Estimating User Stories." Journal of Systems and Software 85, no. 9 (2012): 2086-2095.

S9:Kang, S., O. Choi and J. Baik. "Model Based Estimation and Tracking Method for Agile Software Project." International Journal of Human Capital and Information Technology Professionals 3, no. 2 (2012): 1-15.

S10:Catal, C. and M. S. Aktas. "A Composite Project Effort Estimation Approach in an Enterprise Software Development Project." In 23rd International Conference on Software Engineering and Knowledge Engineering, 331-334. Miami, FL, 2011.

S11:Nunes, N., L. Constantine and R. Kazman. "Iucp: Estimating Interactive-Software Project Size with Enhanced Use-Case Points." IEEE Software 28, no. 4 (2011): 64-73.

S12:Sande, D., A. Sanchez, R. Montebelo, S. Fabbri and E. M. Hernandes. "Pw-Plan: A Strategy to Support Iteration-Based Software Planning." 1 DISI, 66-74. Funchal, 2010.

S13:Bhalerao, S. and M. Ingle. "Incorporating Vital Factors in Agile Estimation through Algorithmic Method." International Journal of Computer Science and Applications 6, no. 1 (2009): 85-97.

S14:Cao, L. "Estimating Agile Software Project Effort: An Empirical Study." In *Proceedings of 14th Americas Conference on IS*, 1907-1916. Toronto, ON, 2008.

S15:Keaveney, S. and K. Conboy. "Cost Estimation in Agile Development Projects." In 14th European Conference on Information Systems, ECIS 2006. Goteborg, 2006.

S16:Tamrakar, R. and M. Jorgensen. "Does the Use of Fibonacci Numbers in Planning Poker Affect Effort Estimates?" In 16th International Conference on Evaluation & Assessment in Software Engineering (EASE 2012), 14-15 May 2012, 228-32. Stevenage, UK: IET, 2012.

S17:Santana, C., F. Leoneo, A. Vasconcelos and C. Gusmao. "Using Function Points in Agile Projects." In Agile Processes in Software Engineering and Extreme Programming, 77, 176-191. Berlin: Springer-Verlag Berlin, 2011.

S18:Bhalerao, S. and M. Ingle. "Agile Estimation Using Caea: A Comparative Study of Agile Projects." In Proceedings of 2009 International Conference on Computer Engineering and Applications, 78-84. Liverpool: World Acad Union-World Acad Press, 2009.

S19:Kuppuswami, S., K. Vivekanandan, Prakash Ramaswamy and Paul Rodrigues. "The Effects of Individual Xp Practices on Software Development Effort." SIGSOFT Softw. Eng. Notes 28, no. 6 (2003): 6-6.

S20:Logue, Kevin, Kevin McDaid and Des Greer. "Allowing for Task Uncertainties and Dependencies in Agile Release Planning." In 4th Proceedings of the Software Measurement European Forum, 275, 2007.