An Effort Estimation Taxonomy for Agile Software Development

Muhammad Usman*, Jürgen Börstler† and Kai Petersen‡

*Department of Software Engineering
Blekinge Institute of Technology, Karlskrona, SE 371 79, Sweden*

**muhammad.usman@bth.se*

†jurgen.borstler@bth.se

‡kai.petersen@bth.se

Received 7 December 2015

Revised 13 August 2016

Accepted 9 March 2017

In Agile Software Development (ASD) effort estimation plays an important role during release and iteration planning. The state of the art and practice on effort estimation in ASD have been recently identified. However, this knowledge has not yet been organized. The aim of this study is twofold: (1) To organize the knowledge on effort estimation in ASD and (2) to use this organized knowledge to support practice and the future research on effort estimation in ASD. We applied a taxonomy design method to organize the identified knowledge as a taxonomy of effort estimation in ASD. The proposed taxonomy offers a faceted classification scheme to characterize estimation activities of agile projects. Our agile estimation taxonomy consists of four dimensions: estimation context, estimation technique, effort predictors and effort estimate. Each dimension in turn has several facets. We applied the taxonomy to characterize estimation activities of 10 agile projects identified from the literature to assess whether all important estimation-related aspects are reported. The results showed that studies do not report complete information related to estimation. The taxonomy was also used to characterize the estimation activities of four agile teams from three different software companies. The practitioners involved in the investigation found the taxonomy useful in characterizing and documenting the estimation sessions.

Keywords: Effort estimation; agile software development; taxonomy.

1. Introduction

Software is developed incrementally and iteratively in Agile Software Development (ASD). Like other activities in ASD, planning and estimation are also performed

*Corresponding author.

This is an Open Access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution 4.0 (CC-BY) License. Further distribution of this work is permitted, provided the original work is properly cited.

iteratively at various stages such as during release or sprint planning. Estimation in ASD has been investigated in a number of studies. We have recently aggregated these works in a Systematic Literature Review (SLR) [1]. In order to get a more broader understanding on how effort estimation is practiced by agile teams, we decided to carry out a follow-up study to elicit the state of the practice on effort estimation in ASD. The SLR included only peer-reviewed empirical works as primary studies, and did not include practitioners' opinion expressed in white papers, blogs, forums, etc. The follow-up study [2], on the other hand, directly elicited opinion of agile practitioners on effort estimation in ASD to complement the findings of the SLR. These studies identified and aggregated knowledge on effort estimation in ASD from the literature (SLR) and the industry (survey). The knowledge includes aspects such as techniques used to estimate effort in ASD, employed size measures and other cost drivers and the context in which estimation is performed in ASD.

We believe that this body of knowledge on effort estimation in ASD needs to be organized to facilitate both future research and practice in this area. Taxonomies have been used to organize the body of knowledge in software engineering (SE) [3] and in other disciplines [4] as well, and have played an important role in maturing these fields. The concept of taxonomy was originally used by the Swedish scientist Carl Linnaeus in the eighteenth century wherein he used hierarchical structure to classify the nature (animals, plants and minerals) based on common physical characteristics [5]. Taxonomy mainly is a classification mechanism, and is defined as follows in two well-known dictionaries:

- The *Cambridge dictionary*^a defines taxonomy as “a system for naming and organizing things, especially plants and animals, into groups that share similar qualities.”
- The *Oxford dictionaries*^b define taxonomy as “The classification of something, especially organisms” or “A scheme of classification.”

Classifications offer both theoretical and practical advantages. On one side they serve as a tool to explore the relationships between entities and identify gaps in the existing knowledge, while on the other side they can support the selection of appropriate entities in a real scenario [6]. In this work we propose a taxonomy to classify and characterize the effort estimation activities of software projects conducted in an ASD context. Few classifications of effort estimation techniques have been proposed in the literature. They are briefly described in Sec. 2. These classifications are about estimation techniques only. Our taxonomy, on the other hand, covers the whole estimation activity that also includes, besides techniques, effort

^a www.dictionary.cambridge.org.

^b www.oxforddictionaries.com.

predictors, estimation context and effort estimates. Following are the contributions of the proposed taxonomy:

- It organizes the body of knowledge on effort estimation in ASD.
- It provides a mechanism to classify and characterize the effort estimation in ASD. The taxonomy dimensions can be used by researchers to assess and improve the reporting of research on effort estimation in ASD.
- It supports effort estimation practice in ASD. The taxonomy dimensions can be used to document important estimation-related information and thereby helping in externalizing the otherwise tacit knowledge of estimation sessions. The taxonomy can also help the agile practitioners during estimation sessions by reminding them of the important aspects that should be considered when effort is being estimated.

The taxonomy was designed by following a taxonomy design method, which is briefly described in Sec. 3. The taxonomy was used to characterize 10 estimation cases reported in the literature. The usefulness of the taxonomy was also demonstrated by classifying and characterizing the estimation activities of five agile teams.

The rest of the paper is organized as follows: Section 2 describes the related work. Section 3 details the research questions (RQs) and methods used to design and evaluate our agile estimation taxonomy. Section 4 presents the results. Section 5 compares the proposed taxonomy with existing work and also discusses the validity threats. Finally, Sec. 6 concludes the paper.

2. Related Work

In this section we describe the background works that lead to the design of the proposed taxonomy, and existing classifications of effort estimation techniques.

2.1. State of the art and practice on effort estimation in ASD

We conducted an SLR [1] with the aim to identify the state of the art on effort estimation in ASD. The SLR was performed by following Kitchenham and Charters guidelines [7]. The SLR reported the results based on 20 papers covering 25 studies on agile effort estimation. The details relating to the planning, execution and analysis of SLR results are described in [1]. To complement the findings of SLR, we carried out a follow-up study [2] to identify state of the practice on effort estimation in ASD. As the data and results of these two empirical studies form the basis for the agile estimation taxonomy presented in Sec. 4, we briefly present and compare the results of these two empirical studies in this section covering the context, estimation techniques and effort predictors used to estimate effort in ASD.

2.1.1. Estimation context

Both empirical studies identified various contextual factors around which agile teams estimate effort. In this sub-subsection, we present the findings of the two studies on three contextual factors: planning level, development activities and agile methods.

The results of the SLR and survey show that estimation is mainly performed during release and iteration planning levels in ASD. However, many primary studies (48%) in our SLR did not explicitly specify the planning level at which estimation was performed. The survey results showed that some agile teams (16.67%) estimate effort during daily meetings as well. A handful of the agile practitioners in the survey (13.33%) stated that their teams estimate effort at project bidding level as well. The results are summarized in Table 1.

The SLR and the survey also investigated the scope of the effort estimates with respect to the development activities, i.e. which development activities are accounted for in the effort estimates. Implementation and the testing were found to be the main estimated activities in both studies. In 36% of the primary studies of SLR, all development activities were accounted for in effort estimates. About 24% of the primary studies did not report this information. The results are listed in Table 2.

Scrum and eXtreme Programming (XP) were identified as the leading agile methods in the SLR studies and the survey responses. It was interesting to note that the primary studies of SLR used only individual agile method (Scrum or XP). However, a large number of the survey respondents cited using a combination of the methods. Kanban, for example, is used mostly in combination with Scrum. These results are summarized in Table 3.

Table 1. Planning levels at which agile teams estimate.

SLR	Survey
No. of studies (%)	No. of responses (%)
Iteration (It): 6 (24%)	Iteration (It): 49 (81.67%)
Release (R): 5 (20%)	Release (R): 25 (41.67%)
Both It and R: 2 (8%)	Daily: 10 (16.67%)
Not described: 12 (48%)	Project bidding: 8 (13.33%)
	Other: 5 (8.33%)

Table 2. Development activities estimated.

SLR	Survey
No. of studies (%)	No. of responses (%)
All: 9 (36%)	All: 8 (13.3%)
Testing (T): 4 (16%)	All except maintenance: 11 (18.3%)
Implementation (I): 3 (12%)	All except analysis: 5 (8.3%)
Both I and T: 3 (12%)	Both I and T: 17 (28.3%)
Not described: 6 (24%)	Implementation only: 7 (11.7%)
	Other combination: 12 (20%)

Table 3. Agile methods practiced.

SLR No. of studies (%)	Survey No. of responses (%)
Scrum: 8 (32%)	Scrum: 25 (41.67%)
XP: 7 (28%)	Scrum, Kanban and XP: 9 (15%)
Not described: 10 (40%)	Scrum and Kanban: 7 (11.67%)
	Scrum and XP: 5 (8.3%)
	XP: 5 (8.3%)
	Other: 9 (15%)

Some other contextual factors were also elicited in the SLR and the survey. These include, besides other factors, the team size, the project/product domain and the team setting. In most cases the team size was found to be less than 10. Agile teams in 80% of the SLR’s primary studies were co-located, while remaining 20% of the SLR studies included geographically distributed teams. In the case of survey, all respondents stated that they are currently part of co-located teams.

2.1.2. Estimation technique

The results of SLR and the survey showed that the estimation techniques (see Table 4) that rely on the subjective decision of the practitioners are used to estimate effort in ASD context. Most frequently used techniques are expert judgment, planning poker and use case points (UCP) estimation method. These findings are in line with the agile philosophy wherein people are valued more than processes, and the heavyweight models and processes are not preferred. Effort estimation in ASD is mostly a group activity, i.e. the effort is estimated by the team.

The primary studies in SLR used various measures such as Mean Magnitude of Relative Error (MMRE) to calculate the accuracy of the estimates. However according to the survey results, these measures are not very common in the industry. For most practitioners a simple comparison of the estimated and the actual efforts is sufficient. The techniques achieved varying degrees of accuracy in the primary studies of SLR, while the survey results indicated that there is a tendency to

Table 4. Estimation techniques used in ASD.

SLR No. of studies (%)	Survey No. of responses (%)
Expert judgment: 8 (32%)	Planning poker: 38 (63.33%)
Planning poker: 6 (24%)	Estimation by analogy: 28 (46.67%)
UCP method: 6 (24%)	Expert judgment: 23 (38.3%)
UCP modification: 5 (20%)	UCP method: 8 (13.33%)
Regression: 3 (12%)	Delphi: 3 (5%)
AI: 3 (12%)	Other: 9 (14.33%)
Other: 6 (24%)	

UCP = Use case points.

underestimate the effort (42% of the respondents believe that estimates are underestimated by 25% or more).

2.1.3. Effort predictors

The product size and the cost drivers constitute effort predictors. The product size is believed to be an important effort predictor. Different metrics (e.g. function points) have been used to measure product size. Our survey and the SLR results (see Table 5) showed that story points, function and use case points are the most frequently used size measures during effort estimation in ASD.

Table 5. Size measures used in ASD.

SLR No. of studies (%)	Survey No. of responses (%)
UCP: 7 (28%)	Story points: 37 (61.67%)
Story points: 6 (24%)	Function points: 17 (28.33%)
Function points: 3 (12%)	UCP: 10 (16.67%)
LOC: 1 (4%)	LOC: 7 (11.67%)
Other: 3 (12%)	Object point: 1 (1.67%)
Not described: 5 (20%)	Not used: 12 (20%)

UCP = Use case points and LOC = Lines of code.

With regard to the cost drivers, the survey and the SLR results identified that the team-related cost drivers (see Table 6) play a significant role in the effort estimation in ASD. This aligns well with values of the agile development wherein the individuals and the teams are valued higher than the processes and the tools. Other cost drivers that are also considered important are nonfunctional requirements (NFRs; e.g. performance, security), project/product domain and customer communication.

Table 6. Cost drivers used in ASD.

SLR No. of studies (%)	Survey No. of responses (%)
Team's skill level: 3 (12%)	Team's skill level: 43 (71.67%)
Team's prior exp.: 2 (8%)	Team's prior exp.: 39 (65%)
Task size: 2 (8%)	Task size: 25 (41.67%)
NFRs: 2 (8%)	Project domain: 25 (41.67%)
Other: 10 (40%)	NFRs: 24 (40%)
Not investigated: 9 (36%)	Customer communication: 14 (23.3%)
	Distributed team issues: 5 (8.33%)
	Other: 10 (16.67%)

NFRs = Nonfunctional requirements.

2.2. Recent studies on effort estimation in ASD

Section 2.1 presented a summary of review on the state of the art and practice on effort estimation in ASD till 2014. In this subsection, we discuss some recently reported studies on effort estimation in ASD.

Garg and Gupta [8] proposed a new cost estimation model for agile software projects. Using project characteristics as input, the proposed model used Principal Component Analysis (PCA) to identify a reduced set of key characteristics that are relevant for the development cost. Constraint solving approach is then employed to ensure the fulfillment of the agile manifesto criteria. The model is applied on agile projects dataset collected from multiple software companies in different countries. The authors noted that the proposed model demonstrated better estimation accuracy in terms of MMRE compared to the other approaches such as planning poker.

Continuous changes in user stories (USs) in ASD lead to uncertainty, which is one of the main reasons for inaccurate time and cost estimates [9]. Popli and Chauhan [9] proposed a technique to manage this uncertainty in ASD to improve estimation. In the proposed technique the user stories are first divided into substories, and then for each of the substories three types of story points are calculated: fastest, practical and maximum. These three estimates are then averaged to compute the final estimated story points. The authors illustrated the proposed technique using a small example project. Zahraoui and Idrissi [10] proposed adjustments in story points measure to improve effort estimation in ASD. The new adjusted story points are calculated by adjusting the story points with the priority, size and complexity of the user stories being estimated.

Lenarduzzi *et al.* [11] reported a replicated case study on functional size measures and effort estimation in Scrum. The case study consisted of a small web-based application developed in ASP.net by four part-time developers using Scrum as development process. The results of the study showed that the effort estimation by developers is more accurate than estimation through functional size measures. The study concluded that the introduction of functional size measures does not help in improving accuracy of the effort estimates in Scrum.

Tanveer *et al.* [12] performed an industrial case study to understand and improve effort estimation in ASD. The results showed that the accuracy of the effort estimation is affected by factors such as developers' knowledge and experience, and the complexity of the required changes in the system. These factors should be considered by developers during estimation process. They also noted that the use of tool to support the impact analysis can also improve effort estimation. The use of tool would aid practitioners in explicitly considering important factors during effort estimation process.

Ramirez-Noriega *et al.* [13] proposed a technique for the identification and validation of the factors that teams use to decide the complexity and importance of the tasks being estimated using planning poker in Scrum. The approach then uses Bayesian Networks (BNs), as knowledge structure, to relate these factors with each other to improve the accuracy of the estimates.

2.3. Classifications of effort estimation techniques

Many different classifications schemes have been proposed for effort or cost estimation techniques during the last 30 years or so. Boehm [14] presented seven classes of

the estimation techniques. These were: algorithmic models, analogy, expert judgment, Parkinson, price to win, top-down and bottom-up. Briand and Wieczorek [15] presented a hierarchical classification of the estimation techniques. At the top are the two main categories: model-based and nonmodel-based techniques. The model-based techniques are further divided into generic and specific model-based techniques. Mendes [16], on the other hand, divided the effort estimation techniques in three broad categories: algorithmic models, expert-based and artificial intelligence techniques.

A more detailed and recent classification of effort estimation techniques is proposed by Trendowicz and Jeffery (TJ) [17]. Effort estimation techniques are divided into three top categories (data-driven, expert-based and hybrid) that are further divided into subcategories. Data-driven techniques, for instance, are divided into proprietary and nonproprietary techniques. Model-based, memory-based and composite are the subcategories of the non-proprietary techniques.

The work presented in this study is different from these classifications. We are not proposing a new classification of effort estimation techniques. Our proposed taxonomy covers the whole effort estimation activity in ASD, whereby the actual effort estimation technique is only one facet of the proposed taxonomy. The facets of the taxonomy are described in Sec. 4. Since the recent work by Trendowicz and Jeffery [17] is very comprehensive and also covers other aspects besides estimation techniques, we will come back to their work in Sec. 5.

3. Research Methodology

In this section, we describe the research questions and methods used to design and evaluate the proposed taxonomy.

3.1. Research questions

Following research questions are addressed in this study:

- RQ1: How to organize the knowledge on effort estimation in ASD?
- RQ2: How can the agile effort estimation taxonomy be used to assess and improve reporting of research on effort estimation in ASD?
- RQ3: What is the usefulness of the taxonomy for effort estimation practice in ASD?

RQ1 is answered by organizing the knowledge on effort estimation in ASD as a taxonomy. The dimensions and facets of the taxonomy are described in Sec. 4. RQ2 is answered by applying the classification mechanism of the proposed taxonomy to characterize the effort estimation activities of the agile projects reported in literature. The characterization is detailed in Sec. 4.2. To answer RQ3, we used the taxonomy to classify and characterize the effort estimation activities of five agile teams. Further details about RQ3 are provided in Sec. 4.3.

Table 7. Taxonomy design method from [19].

Phase	Activity
Planning	Define SE knowledge area (KA)
	Describe the objectives of taxonomy
	Describe the subject matter to be classified
	Select classification structure type
	Select classification procedure type
Identification and extraction	Identify information sources
	Extract all the terms
Design and construction	Perform terminology control
	Identify and describe the taxonomy dimensions
	Identify and describe the categories of each dimension
	Identify and describe the relationships
Validation	Define guidelines for using and updating the taxonomy
	Validate the taxonomy

3.2. Taxonomy design method

In this subsection, we briefly present the method used in this study to design the agile estimation taxonomy. The method is depicted in Table 7. The method is an update of the method presented by Bayona-Oré *et al.* [18], based on our observations and experiences from a systematic mapping study [19] on taxonomies in the software engineering discipline. The revised method consists of four phases and 13 activities. Although the method is presented as sequential arrangement of phases and activities, the taxonomy designers, however, may find it necessary to iterate between phases and the activities.

3.2.1. Phase 1: Planning

Planning is the first phase wherein basic decisions about the taxonomy design are made. It includes the following activities:

Define SE KA: In this activity the SE KA or subarea is selected and described in which the taxonomy is designed. The Software Engineering Body of Knowledge (SWEBOK) version 3 [20] divides the SE discipline into 15 KAs (e.g. software requirements, software design, software construction, testing, etc.). These are further subdivided into subareas and sub-subareas.

The taxonomy proposed in this study is about effort estimation in the ASD context. Effort estimation plays an important role in managing agile projects during release and sprint planning [21]. Effort estimation falls within the scope of the “Software engineering management” KA in SWEBOK [20].

Describe objectives of the taxonomy: The objective(s) of the taxonomy should be clearly stated. The taxonomy designers should not leave it to the audience to guess the objectives of the taxonomy.

The main purpose of the taxonomy in this study is to propose a classification scheme that can be used to characterize effort estimation activity of an agile project.

A number of studies, included in SLR [1] on effort estimation in ASD, have not reported important information related to the context, techniques and predictors used during effort estimation. The results of such studies are hard to analyze and compare, and therefore not very useful for practitioners. The proposed taxonomy could be used by researchers to consistently report important aspects related to effort estimation in ASD.

Agile practitioners, on the other hand, could use the taxonomy during estimation sessions to remind themselves of important aspects that should be considered to arrive at better effort estimates. After estimation session, characterization of the estimation activities using the proposed taxonomy will also facilitate the documentation of the information related to the effort estimation sessions, which otherwise remains mostly tacit. The teams can use this documentation in future estimation sessions.

Describe the subject matter to be classified: The taxonomy designers should make an explicit effort in describing what exactly is classified in the proposed taxonomy. The KA specifies the broad topic of the taxonomy, while the subject matter defines what exactly is classified in the taxonomy. In a taxonomy of software testing techniques, for instance, software testing is the KA, while testing techniques are the subject matter.

Effort estimation activities of the projects that are managed with an agile method is the subject matter of this taxonomy. Estimation activities are characterized by a number of factors such as estimation technique(s) used, considered effort predictors, context, etc.

Select classification structure type: After these initial steps the taxonomy designers have to select an appropriate classification structure. There are four basic classification structures: hierarchy, tree, paradigm and faceted analysis [22]. These structures have their own strengths and limitations, and there are situations in which one is more suitable than the rest. The taxonomy designers should also provide a clear justification for selecting a specific classification structure.

We have selected faceted classification to structure our taxonomy. Faceted classification is suitable for the evolving areas, such as effort estimation in ASD, since it is not required to have complete knowledge of the area to design a facet-based taxonomy [22]. In faceted classification-based taxonomies, the subject matter is classified from multiple perspectives (facets). Each facet is independent and has its own attributes, making the facet-based taxonomies easily evolvable.

Select classification procedure type: The taxonomy designers also need to select and describe the type of the classification procedures that will be employed to assign samples of the subject matter to the most relevant categories. At a broader level there are two types of procedures depending upon the employed measurement system: qualitative and quantitative. The qualitative procedures are based on nominal scales wherein the relationship between the classes cannot be achieved. The researchers use the nominal scale to subjectively assign samples of the subject matter

to the relevant classes. The quantitative classification procedures, on the other hand, are based on numerical scales [4].

Each facet of our taxonomy has a set of possible values. The details are described in Sec. 4. Based on the available data, we used the qualitative procedure to select relevant facet values to characterize a specific estimation activity. In some cases it may not be possible to assign a value simply due to insufficient data.

Identify information sources: Lastly in the planning phase, the information sources to extract the relevant data are identified.

The information sources used in the design of our agile estimation taxonomy consist of: (A) The peer-reviewed empirical studies on effort estimation in ASD published between 2001 and 2013. The evidence from these studies was aggregated in the SLR summarized in Sec. 2. (B) Agile practitioners who participated in our survey study, which is summarized in Sec. 2.

3.2.2. Phase 2: Identification and extraction

In the second phase the relevant data is identified and extracted from the information sources identified in the previous phase. It consists of the following activities:

Extract all terms: The terms and concepts are extracted from the sources identified in the planning phase. Data from the literature was extracted originally as part of the SLR [1]. The data from the agile practitioners was elicited as part of our survey study [2].

Perform terminology control: Next is to remove inconsistencies in the extracted data. It is possible that different researchers have referred to the same concept with different terms or different concepts with the same terms. First level of terminology control was already performed during the data extraction and analysis phases of the two empirical studies (SLR and survey) that are providing the data for the taxonomy presented herein. We, therefore, only had to make few minor adjustments to consistently represent all terms and concepts.

3.2.3. Phase 3: Design

After the extraction phase, the next task is to identify the main dimensions and the categories therein along which the extracted data items could be organized. This phase consists of the following activities:

Identify and define taxonomy dimensions: Faceted classification-based taxonomies have multiple dimensions (or perspectives) at the top along which a subject matter is classified. These dimensions can be identified in two different ways: bottom-up and top-down. In the bottom-up approach the main dimensions and the categories emerge during the extraction and analysis process [23]. In the top-down approach, on the other hand, the taxonomy designers have *a priori* notion of some dimensions and categories [23] due to their knowledge of the selected knowledge area.

We have used a hybrid approach and identified following top-level dimensions of our taxonomy: estimation technique, effort predictors, effort estimate and estimation context. The first three are part of most estimation processes described in the literature, see for example estimation process described in [16]. The fourth dimension (estimation context) emerged during the data extraction and analysis phases to define and document the environment in which the effort estimation activity is carried out in an agile project. These dimensions are detailed in Sec. 4.

Identify and describe categories of each dimension: The next step is to identify and describe the next levels of the taxonomy, which are referred here in taxonomy design method as categories. These categories can be identified by using the same top-down, bottom-up or hybrid approaches described above.

We refer these categories as facets in our taxonomy. Most of these facets emerged when the results of the SLR [1] and survey [2] studies were analyzed. The facets related to the techniques were grouped under estimation technique dimension. Similarly cost drivers and size measures were grouped under effort predictors dimension. The data items related to the effort estimate (e.g. unit of estimate) were categorized under the effort estimate dimension. The SLR and survey studies also identified number of factors that characterize the context of the estimation activity. The identification of these factors leads to the fourth dimension of the taxonomy, i.e. estimation context. The facets are described in detail in Sec. 4.

Identify and describe relationships, if any: The relationships between dimensions and categories are identified and described clearly. Note that in some cases there is no relationship between the dimensions, i.e. this activity might be skipped.

The objective of the taxonomy proposed in this study is to provide a classification mechanism to characterize effort estimation activity in an ASD context. Each dimension is a grouping of facets that are closely related to each other. The four dimensions together can characterize an estimation activity. However no other specific relationships between dimensions are defined, allowing for easy addition of new dimensions and the facets.

Define guidelines for using and updating the taxonomy: The aim of this activity is to describe how the taxonomy can be used. As a result of the application of the taxonomy and other future research efforts, it is highly likely that the need to update the taxonomy emerges. The taxonomy designers should also describe how the taxonomy can be updated in future.

The proposed effort estimation taxonomy can be used to assess and improve the reporting of studies on effort estimation in ASD. The taxonomy facets can be used as a checklist to ensure that all important factors about effort estimation are reported. Practitioners can document important information by using the taxonomy to characterize the specific estimation activity of an agile project. This documented information can serve as an important repository in improving future effort estimation. We applied the proposed agile taxonomy on knowledge extracted from literature and industry to demonstrate these uses. The details are provided in Secs. 4.2 and 4.3.

The dimensions and the facets of the proposed taxonomy are not exhaustive, and should be updated based on future research efforts and feedback from industry. At present they are based on knowledge that we aggregated in our SLR [1] and survey [2]. We have used faceted classification in our taxonomy. Faceted classification is a flexible structure wherein new facets can be added relatively easily. One likely extension is under the facet “effort predictor.” Currently we only included those cost drivers that are identified as important in the SLR and the survey. The future research on effort estimation in ASD may and should investigate more cost drivers, which could then be included in our taxonomy.

3.2.4. Phase 4: Validation

Validation is the last phase of this method wherein the taxonomy designers attempt to demonstrate the working or usefulness of the designed taxonomy. This phase has only one activity. A number of approaches can be used to validate a taxonomy. It includes establishing the orthogonality of taxonomy dimensions, comparing with existing taxonomies and demonstrating its usefulness by applying it on existing literature [24].

Since there is no existing taxonomy on effort estimation in ASD, benchmarking our taxonomy against existing ones is not strictly possible. However, in Sec. 5 we compare our taxonomy with a recent and extensive review of the software effort estimation area by Trendowicz and Jeffery [17]. Their work is selected for the comparison as it is not only recent and comprehensive, but it also covers aspects related to our taxonomy dimensions. Orthogonality of the taxonomy dimensions is discussed in Sec. 4.1. The four dimensions (Fig. 1), representing the top level of taxonomy, are clearly distinguishable from each other. Effort estimation is performed in a certain Context, wherein agile teams use certain estimation technique(s) and effort predictors to arrive at an effort estimate.

Two methods are employed to evaluate the usefulness of the proposed taxonomy. These methods are briefly described here.

Taxonomy evaluation — Literature: The proposed taxonomy is used to characterize 10 estimation cases reported in the literature. The studies were selected from the set of primary studies included in our SLR on effort estimation in ASD. We selected only those studies that were based on real agile projects. An extraction sheet was prepared with fields corresponding to the facets of the proposed taxonomy. The data reported in the selected studies was extracted and entered in the sheet with respect to the facets of the taxonomy. The results of the evaluation are presented in Sec. 4.2.

Taxonomy evaluation — Industry: The usefulness of the proposed taxonomy is also demonstrated by classifying and characterizing the effort estimation activities of five agile teams. We conducted interviews of four agile practitioners about effort estimation activities that their teams performed in recent sprint or release planning meeting. These four agile practitioners represent a convenience sample, i.e. they were recruited based on our personal contacts. They belong to three different companies, and

Table 8. Details of the interviewees.

#	Current location	Current role	Exp. in current company	Total exp.	No. of estimation sessions	No. of teams and products
1	Sweden	Test Lead	1.5 years	6 years	1	1
2	Pakistan	Development Manager	11 years	11 years	2	2
3	Sweden	Product Owner	7 years	11 years	2	1
4	Sweden	Developer	1 year	10 years	1	1

provided data about effort estimation activities performed in the development of five different products. Two interviewees lead the effort estimation sessions, one participate as testing lead and one as developer in their respective companies. The details of the companies and products are described in Sec. 4.3. Table 8 lists their brief profiles.

The interviews were semistructured and were conducted by the first author. The interviewees were first asked to describe when and how effort estimation is carried out in their respective teams. The interviewees were then asked to select recent instances of estimation sessions in which they participated. The taxonomy dimensions and the facets were used to ask questions and elicit the data required to characterize the specific estimation activity. The interviews lasted, on an average, for an hour. After the interviews, the data extraction sheets were sent to the respective interviewees for further feedback, validation and corrections, if required. The results of the evaluation are presented in Sec. 4.3.

Together these four practitioners from three different companies provided data about six different estimation sessions corresponding to five different products and teams.

4. Results

In this section first we present the agile estimation taxonomy to answer RQ1. Later, the applications of taxonomy in the literature and the industrial estimation cases are presented to answer RQ2 and RQ3, respectively.

4.1. RQ1: Organizing the knowledge on effort estimation in ASD

We have organized the identified knowledge on effort estimation in ASD as a taxonomy. Taxonomies are an effective tool to organize and communicate the knowledge in an area [3]. The proposed taxonomy was created by following a method that was described in Sec. 3. In this subsection, we present the outcome of that method, i.e. the agile estimation taxonomy.

At the top level the taxonomy has four dimensions, which are presented in Fig. 1. A dimension encompasses all facets that are related to each other. The dimension “Effort predictors,” for example, includes effort predictors employed during estimation. These four dimensions define the first level of taxonomy, and provide an overview of the taxonomy at a broader level.

Fig. 1. First level of the agile estimation taxonomy.

Estimation context represents grouping of those facets (or categories in more general terms) that define and characterize the context in which the estimation activity is carried out in an agile project. It includes, beside others, agile method in use, team size, project domain, etc. *Effort predictors* dimension encompasses predictors used during effort estimation. *Estimation technique* dimension includes facets related to the technique used to arrive at effort estimates. *Effort estimate* includes the facets related to effort estimates such as estimate value, estimate type, etc.

In order to completely characterize a specific effort estimation activity of an agile project, facets of all dimensions need to be described. In the following, we describe each of these four dimensions and their facets in detail.

4.1.1. *Estimation context*

Context is an inseparable part of the software practice, and it needs to be properly captured and reported in the empirical studies to communicate the applicability of the research findings. The empirical research findings are not universal due to the unique context in which the research is conducted. The empirical research needs to be properly contextualized so that the readers can determine the relevance of the results for their contexts, i.e. what is applicable for a company, and when and where to use the reported findings [25].

The context dimension has eight facets. These facets and their possible values are depicted in Fig. 2. They are described in the following.

- *Planning level*: Estimation supports planning at different levels in ASD. It includes mainly release and sprint planning, while some teams may also estimate during daily meetings [21]. Project bidding is another level wherein companies have to estimate the total development effort upfront in order to bid for the projects.
- *Main estimated activity*: This facet describes which development activities are accounted for in the effort estimates. For example, does the total effort estimate include the time spent on fixing bugs, i.e. maintenance, or it is considered separately. It is possible that some team members have different understandings towards the estimation scope due to, for example, the reason that the team is newly

Fig. 2. Estimation context dimension.

formulated, or it includes some inexperienced developers. Explicit consideration of this facet during estimation sessions will improve shared understanding amongst team members with respect to the scope of the estimation.

- *Agile method(s)*: This facet documents the agile method practiced by the agile team. Most agile teams customize and adopt agile method(s) to suit their context.
- *Product/project domain*: This facet represents the domain of product for which the development effort is being estimated. Different domains could lead to different sets of effort predictors and their impact on the total effort. For instance, different domains would have varying nonfunctional requirements. We have used the categories from the domain taxonomies reviewed by Glass and Vessey [3] as example facet values in Fig. 2.
- *Project setting*: This facet captures the setting in which the agile teams are developing the product. There are two broad settings, co-located team wherein all members work at the same location, and distributed development wherein the teams are placed at different geographical locations. The latter introduces additional challenges related to communication and coordination. In case of the distributed teams many different settings, depending on the temporal and geographical distances, are possible. Smite *et al.* [24] proposed a global software engineering (GSE) taxonomy that characterizes different settings of the global teams. We have used situations in their GSE taxonomy as example facet values in Fig. 2. The teams in distributed development could be working either in an onshore or an offshore setting.

- *Estimation entity*: Requirements are the basic input to the estimation activity. Requirements can be specified in different ways and at varying levels of granularity. In ASD requirements are mostly specified in the form of user stories. In some cases user stories are estimated, while other teams divide the stories into tasks, which are then estimated. This facet documents the entity (e.g. task, user story or use case) that is being estimated.
- *Number of entities estimated*: It records the number of user stories or tasks estimated in a specific estimation session, and thus conveys important information related to the size of the sprint or release.
- *Team size*: It documents the size of the team which is responsible for developing the estimated tasks or stories.

4.1.2. Effort predictors

Effort predictors play an important role in arriving at effort estimates. They consist of size and other cost drivers such as team capabilities, nonfunctional requirements, etc. Product size is considered as one of the main determinants of the required development effort [17]. Cost drivers also contribute to the required effort depending on the context.

The effort predictors dimension has six facets. These facets and their possible values are depicted in Fig. 3.

- *Product size*: Development effort, in general, is strongly correlated with product size. Many research efforts have been made to empirically relate effort with product size [26]. This facet documents whether the agile team uses size as a

Fig. 3. Effort predictors dimension.

predictor and which metric is used to represent the size. It is also possible that estimators considered the task or story size as one of the predictors without using any specific size metric.

- *Team's prior experience*: A development team's prior experience with similar tasks impacts the required effort. Estimated effort for sprints that involve similar and familiar tasks will be low as compared to a sprint involving unfamiliar tasks. This facet describes whether a team's prior experience was considered in arriving at the effort estimates.
- *Team's skill level*: Team's skill level also impacts the required effort. This facet documents whether skill level of the team members was considered during the effort estimation session.
- *Nonfunctional requirements*: Stringent nonfunctional requirements increase the development effort. This facet records whether or not, and which, nonfunctional requirements were considered in arriving at effort estimates. Chung *et al.* [27] provided a review of the types and classifications of the nonfunctional requirements. We have used few nonfunctional requirements, from a large list included in their work, as example facet values in Fig. 3.
- *Distributed teams' issues*: How many sites collaboratively develop software also has an impact on the development effort due to the increased complexity of the collaboration and communication. The cultural and geographical barriers between distributed development teams can increase the development effort.
- *Customer communication*: Effective communication with the customer is an important aspect of ASD, and breakdown in this communication could lead to delays and problems in resolving issues. It could further lead to effort overruns. This facet guides the estimating team to see if customer communication should be used as an effort predictor in their context.

The actual facet values for some cost drivers, such as team's skill level, in Fig. 3 are not provided. At the first level it is important to characterize whether or not certain cost drivers are considered relevant during effort estimation. It is not simple to quantitatively measure these cost drivers. A commonly used approach in most estimation methods, such as COConstructive COst MOdel (COCOMO) [28], is to rank these cost drivers on an ordinal scale such as low, medium or high. A similar approach can be used here, if specific quantitative constructs or measures are absent.

4.1.3. Estimation technique

This dimension encompasses facets related to estimation techniques that should be documented in order to characterize estimation activities of an agile team. The facets of this dimension and their corresponding example values are depicted in Fig. 4.

- *Technique*: This documents the actual estimation technique applied by the agile team. According to the survey and SLR results most frequently applied

Fig. 4. Estimation technique dimension.

techniques are planning poker, expert judgment and estimation by analogy. The agile teams can also use a combination of techniques to arrive at an effort estimate.

- *Type of technique*: There are different types of estimation techniques and models such as algorithmic, expert-based and artificial intelligence-based techniques and models. In ASD, estimation techniques that rely on subjective assessment of experts are frequently used [1]. These techniques can be used by an individual or a group of experts. This facet documents whether the effort was estimated using an individual or group-based estimation technique.

4.1.4. Effort estimate

Effort estimates are the main output of the estimation activity. The facets in this dimension characterize effort estimate. These facets and their corresponding values are depicted in Fig. 5.

- *Estimated effort*: This facet documents the main output of the estimation session or activity, i.e. the estimated effort.
- *Actual effort*: It is also important to have the actual effort, at the end of release or sprint, to enable comparison with the estimated effort.
- *Estimate type*: The estimate could be of different types, e.g. point estimate, three-point estimate or a distribution. This facet documents the type of estimate used by the agile team.

Fig. 5. Effort estimate dimension.

- *Estimate unit*: Effort can be estimated in terms of hours, weeks or months. The concept of ideal time has also been applied in ASD. This facet records the unit used to represent the effort estimates.
- *Accuracy level*: This facets documents the accuracy level achieved by the used technique. It is better to have the actual and estimated efforts reported to have a better idea about the accuracy of the estimates. However, some studies only report accuracy levels in terms of the percentages (e.g. MMRE).
- *Accuracy measure*: This facet records whether or not, and which, measure is used to assess the accuracy of the applied estimation technique. The commonly used accuracy measures are listed as example facet values in Fig. 5.

4.2. RQ2: Reporting of research on effort estimation in ASD

In order to evaluate usefulness of the proposed taxonomy in assessing the completeness of reporting, we used the taxonomy to characterize 10 effort estimation cases reported in the literature. The estimation cases were elicited from eight empirical studies on effort estimation in ASD. The characterizations are presented in Table 9. Agile estimation taxonomy is able to describe the effort estimation cases reported in literature. It is however interesting to note that some studies do not report important information, which makes it hard to compare and analyze results of these studies. We observed the following specific issues during this exercise:

- Authors make an effort to describe the context in which estimation studies are conducted. However in many cases the context is very briefly described, and is scattered in the report. We were not able to extract project domain, planning

Table 9. Taxonomy application: Literature.

Facet	[29]	[30]	[30]	[31]	[32]
Planning level	Release	Iteration	Iteration	Iteration and Release	?
Estimated activity	I	I	I	All	All
Agile method(s)	XP	XP	XP	?	?
Project domain	IS for an ISP	?	?	Web MIS	MIS
Project setting	Co-located	Co-located	Co-located	Distributed	?
Estimation unit	User story	User story	User story	User story	Use cases
No. of tasks/stories	50	1325	1325	?	12
Team size	16–20	16–20	16–20	?	3
Size	not used	Length of US	?	Story points	UC points
Team's prior experience	+	–	?	–	?
Team's skill level	+	–	?	–	?
Nonfunctional requirements	–	–	?	Performance, security	?
Distributed teams' issues	–	–	?	+	?
Customer communication	–	–	?	–	?
Technique	Planning poker	Stepwise reg.	Exp. judgment	Constructive agile estimation algorithm	UCP estimation
Type	Group	Other	Group	?	?
Estimation unit	Pair days	SPs converted to min	h	?	h
Estimate type	Point	Point	Point	Point	Point
Actual effort	?	?	?	?	?
Estimated effort	?	?	?	?	2600 h
Accuracy measure and level	MMRE (0.48), MdMMRE (0.25)	MMRE (0.90), PMRE@25% (0.17), mMER (0.67)	MMRE (0.28), PMRE@25% (0.51), mMER (0.25)	?	?

Table 9. (Continued)

Facet	[33]	[33]	[34]	[35]	[36]
Planning level	?	?	Release	Release	Iteration
Estimated activity	Testing	Testing	All	I and T	?
Agile method(s)	Scrum	Scrum	?	Scrum	XP
Project domain	Web app.	Web app.	Accounting system	Web student IS	Desktop support sys.
Project setting	Distributed	Distributed	Co-located	Co-located	Co-located
Estimation unit	Use case	Use case	User story	User story	Task
No. of tasks/stories	?	?	?	200	4000
Team size	17	17	2	40 (10 teams of size four)	15
Size measure	?	UC points	Story points	Story points	?
Team's prior experience	?	+	?	?	?
Team's skill level	?	+	?	?	?
Nonfunctional requirements	?	+	?	?	?
Distributed teams' issues	?	-	N/A	?	?
Customer communication	?	-	?	?	?
Technique	Exp. judgment	UCP method	?	Planning poker	?
Type	Single	?	?	Group	?
Estimation unit	Hours (h)	h	h	Ideal days	Ideal h
Estimate type	Point	Point	Point	Point	Point
Actual effort	1090 h	1090 h	380 h	?	?
Estimated effort	750 h	870 h	400 h	?	?
Accuracy measure and level	MRE (0.32)	MRE (0.21)	MRE (0.05)	MBRE (1.1074)	MMRE (0.19)

+ = Considered, - = Not considered and ? = Data unavailable.

level and the number of estimated stories/tasks in two, three and four cases, respectively.

- The reporting of the studies lack the details with respect to the effort predictors used during effort estimation. In six cases we were not even able to establish whether or not, and which, cost drivers were used. It is hard to imagine a real scenario wherein effort is estimated without considering any effort predictors. It is possible that the focus of a study is not on effort predictors. However this then should be explained and described explicitly.
- Two studies have not described the technique used to estimate effort. The remaining eight studies have described the used technique and corresponding accuracy measure used. However we were not able to determine in five cases whether the technique was used by an individual or group of experts.
- Actual and estimated efforts are not reported in six cases, which makes it hard to analyze and compare the findings.

The proposed taxonomy can be used by researchers to make sure that important information related to agile effort estimation is duly reported. This, we hope, will improve the reporting and would facilitate comparing and analyzing the results of studies on effort estimation in ASD.

4.3. RQ3: Usefulness of taxonomy for effort estimation practice in ASD

The aim of this RQ was to evaluate if the proposed taxonomy can be used by agile teams in documenting their estimation sessions for future use. We used the taxonomy to classify and characterize the estimation activity of five agile teams working in three different companies. First we provide a brief description of the companies, teams and related products. The companies are referred here as A, B and C due to privacy concerns.

4.3.1. Company A

Company A is a medium-sized software company established in the mid-90s headquartered in Finland with offices in Sweden and India as well. A develops and offers Business Support Solutions (BSS) to telecommunication service providers. It offers BSS services related to customer, product and revenue management. We collected the effort estimation-related data from the recent sprint planning meetings of two different teams at company A. We refer to these two teams as A1 and A2. A sprint normally lasts for two weeks. Both teams are located in Sweden, and are using Scrum as development process. A1 is responsible for developing Order Management System (OMS) of BSS, while A2 is developing Customer Information System (CIS).

We interviewed a developer from team A1 and a testing lead from A2, who both participated in the sprint planning meetings of their respective teams. Each

Table 10. Characterizing effort estimation activity of company A.

Dimension	Facet	OMS	CIS
Context	Planning level	Sprint	Sprint
	Main estimated activity	Implementation and Testing	Implementation and Testing
	Agile method(s)	Scrum and XP practices	Scrum
	Product domain	Telecom	Telecom
	Project setting	Co-located	Co-located
	Estimation unit	Task	Task
	Number of tasks/stories	60 tasks	36 tasks
	Team size	8	8
Effort predictors	Size	Not used	Considered without any measure
	Team's prior experience	Considered	Considered
	Team's skill level	Considered	Considered
	Nonfunctional requirements	Considered	Considered
	Distributed teams' issues	N/A	N/A
	Customer communication	Not considered	Not considered
Estimation technique	Technique name	Planning poker	Planning poker
	Type	Group estimation	Group estimation
Effort estimate	Estimation unit	h	h
	Estimate type	Point	Point
	Actual effort	750 h	325 h
	Estimated effort	580 h	240 h
	Accuracy measure used	Comparison of actual and estimate	Comparison of actual and estimate
	Accuracy level achieved	30% underestimated	35% underestimated

interview lasted for about an hour. The effort estimation data was entered in extraction sheets by the interviewer, which were later sent to the interviewees for corrections and adding the missing information, if any. Our agile estimation taxonomy was used (see Table 10) to classify the specific effort estimation activities of both teams.

4.3.2. Company B

Company B is a very large organization, headquartered in Sweden, providing services and support systems related to communication technology. B develops and provides software solutions for Operations and Business Support Solutions (OSS/BSS). We interviewed a product owner for 2 h who leads the release plan meetings that involve both design and testing leads and other team members. The teams are based in Sweden, and are following Scrum as their development process. They are responsible for developing one subsystem of a large billing system.

We elicited effort estimation data corresponding to two different releases; one is based on customer requirements, while the other is research-based internal release. We used the data elicited in the interview to characterize the effort estimation

Table 11. Characterizing effort estimation activity of company B.

Dimension	Facet	External release	Internal release
Context	Planning level	Release	Release
	Main estimated activity	Implementation and Testing	Implementation and Testing
	Agile method(s)	Scrum	Scrum
	Product domain	Telecom (billing)	Telecom (billing)
	Project setting	Co-located	Co-located
	Estimation unit	Task	Task
	Number of tasks/stories	137 tasks	227 tasks
	Team size	6–10	6–10
Effort predictors	Size	Considered without any measure	Story points
	Team's prior experience	Considered	Considered
	Team's skill level	Considered	Considered
	Nonfunctional requirements	Considered	Considered
	Distributed teams' issues	N/A	N/A
	Customer communication	Considered	Not considered
Estimation technique	Technique name	Planning poker	Planning poker
	Type	Group estimation	Group estimation
Effort estimate	Estimation unit	h	h
	Estimate type	Point	Point
	Actual effort	1685 h	4325 h
	Estimated effort	1574 h	3726 h
	Accuracy measure used	Comparison of actual and estimate	Comparison of actual and estimate
	Accuracy level achieved	6% underestimated	16% underestimated

activities performed in release planning meeting of both releases. The details are listed in Table 11.

4.3.3. Company C

Company C is a small to medium-sized Pakistan-based software company established in the late 90s. It has around 50 employees based in Islamabad, Pakistan, and develops mainly performance enhancements and database applications. C follows a customized version of Scrum in all projects. We collected effort estimation data, related to our taxonomy, from the release planning meetings of two teams working on two different products. We refer to the two teams as C1 and C2. A release normally consists of four iterations, and takes about 4–6 months to complete. An iteration is usually 4–6 weeks long.

Data was collected in a 1.5-h interview of the development manager of the company who is responsible for overseeing all development teams of the company. He has the additional responsibility of managing the systems' architecture as well. He leads both the release and sprint plan meetings of both products, which are also attended by relevant development and testing leads. The first author conducted the interview and recorded the responses in an excel file. The excel file was later sent to

Table 12. Characterizing effort estimation activity of company C.

Dimension	Facet	Team C1	Team C2
Context	Planning level	Sprint	Sprint
	Main estimated activity	Implementation and Testing	Implementation and Testing
	Agile method(s)	Customized Scrum	Customized Scrum
	Product domain	Performance enhancement app.	NoSql database app.
	Project setting	Co-located	Co-located
	Estimation unit	Task	Task
	Number of tasks/stories	40 tasks	110 tasks
	Team size	6	10
Effort predictors	Size	Considered without any measure	Considered without any measure
	Team's prior experience	Considered	Considered
	Team's skill level	Considered	Considered
	Nonfunctional requirements	Considered	Considered
	Distributed teams' issues	N/A	N/A
	Customer communication	Not considered	Not considered
Estimation technique	Technique name	Expert judgment	Expert judgment
	Type	Group estimation	Group estimation
Effort estimate	Estimation unit	h	h
	Estimate type	Point	Point
	Actual effort	1500 h	Not available yet
	Estimated effort	1450 h	4400 h
	Accuracy measure used	Comparison of actual and estimate	Comparison of actual and estimate
	Accuracy level achieved	Minor underestimation	Not available yet

the interviewee for entering missing information and validation of the recorded information.

Team C1 consists of six members, and is responsible for developing a performance enhancement application (referred as P1) for .Net and Java development environments, while team C2 consists of 10 members and is developing an NoSQL-based database product (P2). We collected the effort estimation-related data from the sprint plan meetings for the first three sprints of both products. The effort estimation activities of both teams C1 and C2 are characterized using our agile estimation taxonomy in Table 12. We only used the data of the first sprints for both products.

4.3.4. Summary

The interviewed practitioners found the taxonomy useful in characterizing their estimation activities. They noted that this offers a simple mechanism to document this information, which can be used subsequently. We asked them to suggest any changes or updates in the taxonomy when we sent the extraction sheet for validating extracted data. The following three extensions were suggested in agile estimation taxonomy.

Fig. 6. Extending the context dimension.

- When the estimation is done as part of release planning, the taxonomy should also document the number and duration of the iterations/sprints in a release.
- While capturing the context, the taxonomy should also characterize the development mode, i.e. whether the agile team is engaged in a product- or project-based development.
- It is not enough to capture the project/product domain alone in one factor. The taxonomy should have separate facets for industry and application domains to fully characterize product domain. Few examples of industry are transportation, telecommunication, insurance, education, etc.; while examples of application domain are information system, command and control, avionics, data processing, etc.

These suggested extensions are added in the context dimension of the agile estimation taxonomy. Figure 6 presents the extensions as new facets in the context dimension. It is relatively easy to add or remove a facet in any dimension. This is one of the benefits of using the faceted classification structure in a taxonomy.

We have used the categories of software types proposed by Forward and Lethbridge [37] as example facet values for the application-type facet in Fig. 6. These are high-level categories. More specific types of the software applications are also listed in their taxonomy.

5. Discussion and Validity Threats

The agile estimation taxonomy is based on the knowledge identified in our SLR [1] and survey [2] studies. Companies and researchers can introduce new dimensions and

facets into the taxonomy. Since we used a faceted classification structure in designing this taxonomy, it is flexible enough to accommodate changes in the future.

In this section, we compare our taxonomy with an existing work and also describe limitations of our work.

5.1. Comparing agile estimation taxonomy with existing work

Trendowicz and Jeffery [17] provide a comprehensive review on software effort estimation covering aspects related to effort estimation foundations, techniques, drivers, context and guidelines to select an appropriate technique in a specific context. We compare dimensions and facets of our taxonomy with the relevant aspects of the work by Trendowicz and Jeffery^c [17]. It is important to note the following points before moving further.

- Trendowicz and Jeffery do not present their work as a taxonomy or classification. However, the book covers many important aspects related to software effort estimation, and therefore provides an opportunity for comparing our taxonomy.
- The TJ estimation reference is about software effort estimation in general, is not meant specifically for ASD.
- The evidence identified and analyzed in our SLR [1] served as one of the inputs to the agile estimation taxonomy presented in this paper. The SLR was published in 2014, and included only peer-reviewed works published before December 2013. The TJ estimation reference was published as a book in 2014. Our taxonomy and the TJ estimation reference can therefore be considered as independent works.

5.1.1. Effort estimation context

The facets corresponding to the estimation context dimension of our agile estimation taxonomy are compared with the contextual factors in the TJ estimation reference in Table 13. There are six contextual facets that are not covered in the TJ estimation reference: planning level, main estimated activity, project/product domain, type and number of estimation entities, development mode and iteration/sprint details. The TJ estimation reference is general and therefore some factors that are more specific for ASD are not covered. Planning levels (i.e. release planning, iteration planning), estimation entities such as user stories and tasks and concept of sprint/iterations are more specific to the ASD context.

There are two contextual factors listed in the TJ estimation reference that are not part of our taxonomy: the programming language and the development type (new development or enhancement). These could be added as new facets in the estimation context dimension in our taxonomy as both provide additional information that might be helpful in characterizing the estimation context.

^cWe refer to their work as “TJ estimation reference” in this comparison.

Table 13. Comparison of facets in the estimation context dimension.

Agile estimation taxonomy	TJ estimation reference
Planning level	Not covered
Main estimated activities	Not covered
Agile method(s)	Covered as development life cycle in contextual factors list
Project/product domain	Not covered
Project setting	Partly covered in team structure as an effect of scale factor
No. and type of estimation entity	Not covered
Team size	Covered as an effect of scale factor
Application type	Covered as application domain in contextual factors list
Development mode	Not covered
Iteration duration and count	Not covered

5.1.2. Effort estimation techniques

The TJ estimation reference includes a very detailed classification of the effort estimation techniques. The classification is presented as a hierarchy which includes specific techniques as leaf nodes and their types at higher levels of the hierarchy. The facets of the estimation technique dimension are compared with this work in Table 14. All facets of the estimation technique dimension of our taxonomy are also covered in the TJ estimation reference.

5.1.3. Effort predictors

All facets of the effort predictors dimension of our agile estimation taxonomy are also present in the TJ estimation reference (see Table 15).

There are many effort predictors, described in the TJ estimation reference as common effort drivers, that are not part of our taxonomy: software, database and architecture complexity, requirements novelty, requirements stability, project

Table 14. Comparison of facets in the estimation technique dimension.

Agile estimation taxonomy	TJ estimation reference
Estimation techniques	Covered in the classification of effort estimation techniques
Type	Type is also covered in the classification of techniques
Accuracy measures	Covered as accuracy metric as part of handling uncertainty

Table 15. Comparison of facets in the effort predictors dimension.

Agile estimation taxonomy	TJ estimation reference
Size	Software size covered as scale factor
Team's prior experience with similar tasks	Covered as application familiarity in effort drivers list
Team's skill level	Covered as team capabilities in effort drivers list
Nonfunctional requirements	Covered as required software quality in effort drivers list
No. of development sites	Covered as project distribution in effort drivers list
Customer communication	Covered as customer involvement in effort drivers list

Table 16. Comparison of facets in the effort estimate dimension.

Agile estimation taxonomy	TJ estimation reference
Estimated effort	Described in the principles of effort estimation
Actual effort	Described in the principles of effort estimation
Estimate type	Partly described as part of handling uncertainty
Estimate unit	Not covered
Accuracy level	N/A

constraints, schedule pressure, tool usage, CASE tools and testing tools. These effort predictors are currently not part of our taxonomy as we could not find any evidence wherein these have been investigated as drivers of project effort in ASD. There is a need to investigate whether these generic effort drivers are relevant in an ASD context.

5.1.4. *Effort estimate*

Apart from estimation unit and accuracy level, other facets of this dimension of our taxonomy are described in the TJ estimation reference (see Table 16). These are, however, described in different places in the book. Some studies only report the accuracy level achieved in terms of percentages (e.g. MMRE: 30%), and do not specify the actual and estimated efforts. As for estimation unit, in ASD some teams use the concept of ideal time as the unit of effort instead of the standard person hours/month unit [21].

5.1.5. *Summary*

The main findings of this comparison are described in the following:

- There are aspects that are specific to ASD, such as some facets in the context dimension. These are not part of TJ estimation reference, which is about effort estimation in general. Agile practitioners should therefore consult estimation works, such as our taxonomy, that are specific to ASD besides considering general works on effort estimation.
- There are many aspects, such as some cost drivers, included in the TJ estimation reference that are not covered by our agile estimation taxonomy. There is a need to investigate the relevance and importance of these effort drivers in an ASD context in the future research efforts.

5.2. *Validity threats*

We evaluated the proposed taxonomy by using it to characterize estimation cases from the literature and the industry. We also compared our taxonomy with a reference work by Trendowicz and Jeffery [17]. However, this study may still have some limitations. We discuss the validity threats using the categorization proposed by

Petersen and Gencel [38] corresponding to two main phases of research: data collection and analysis.

- Descriptive validity is concerned with issues that may arise due to poor gathering and recording of data, which would result in the inaccurate description of the “truth”. The data identified and aggregated in the SLR [1] and the survey [2] was used as main input to the taxonomy designed in this study. In the SLR the studies were identified and analyzed following Kitchenham and Charters guidelines [7]. The survey instrument (questionnaire) was designed and reviewed iteratively to ensure readability and correctness. The survey was based on a sample of 60 agile practitioners. The validity threats and the corresponding mitigating actions for the SLR and the survey are described in [1, 2] respectively.
- Interpretive validity is concerned with ensuring that reasonable conclusions are drawn based on the collected data, and the issues such as researchers’ bias do not lead to incorrect conclusions. We applied a taxonomy design method to organize the knowledge on effort estimation in ASD as taxonomy in a systematic manner. The taxonomy was evaluated by using it to characterize estimation cases from the literature and the industry. The taxonomy was also compared with a recent detailed review on software effort estimation [17].
- The taxonomy is applied to characterize 10 agile effort estimation cases from the literature and six estimation sessions of five different agile teams. The results are generalizable to only those projects and teams that have similar contexts.

The authors of this study have experience in ASD and/or effort estimation. We followed a method to design the taxonomy in a systematic way. However, we still believe that the involvement of more experts could have further improved the completeness and utility of the proposed taxonomy.

6. Conclusions and Future Work

The design of taxonomies helps in structuring and communicating existing knowledge, and in advancing the research [3]. We organized the existing body of knowledge on effort estimation in agile software development as a taxonomy. The taxonomy was developed in a systematic way by following a taxonomy design method. The usefulness of the taxonomy was demonstrated by applying it on data extracted from literature and industry. The taxonomy was applied to characterize 10 selected estimation cases reported in the literature. It was observed that some studies do not report important information related to effort estimates, effort predictors used and the context in which the estimation activity was performed. The taxonomy can be used by researchers in reporting their findings in a consistent way to allow the audience to compare, analyze and aggregate their findings. The taxonomy was also applied to characterize the estimation activities of agile teams of three companies.

The involved practitioners noted that the taxonomy provides a useful mechanism to document their estimation sessions.

The taxonomy has not been used during effort estimation sessions. We plan to develop a checklist instrument based on our agile estimation taxonomy with the aim to support agile practitioners during estimation sessions. The idea is to employ the checklist during sprint or release plan meetings while estimation is in progress. Practitioners could use the checklist to remind themselves of important factors that should be considered while estimating user stories or tasks. It will also support in documenting the knowledge used by agile practitioners to arrive at effort estimates, which otherwise would remain tacit in most cases. Our taxonomy could also be used as a tool in the development of a repository of the effort estimation knowledge, which could be used in the long term to better understand and improve the practice of effort estimation in ASD.

References

1. 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 in Software Engineering*, 2014, pp. 82–91.
2. 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. Evaluation and Assessment in Software Engineering*, 2015, pp. 12:1–12:10.
3. R. L. Glass and I. Vessey, Contemporary application-domain taxonomies, *IEEE Softw.* **12** (1995) 63–76.
4. G. R. Wheaton, Development of a taxonomy of human performance: A review of classificatory systems relating to tasks and performance, Technical report, American Institute for Research, Washington, DC, 1968.
5. Wikipedia, Carl Linnaeus (2015), https://en.wikipedia.org/wiki/Carl_Linnaeus.
6. S. Vegas, N. Juristo and V. Basili, Maturing software engineering knowledge through classifications: A case study on unit testing techniques, *IEEE Trans. Softw. Eng.* **35** (2009) 551–565.
7. B. Kitchenham and S. Charters, Guidelines for performing systematic literature reviews in software engineering, EBSE Technical report, No. EBSE-2007-01, Keele University, 2007.
8. S. Garg and D. Gupta, PCA based cost estimation model for agile software development projects, in *Int. Conf. Industrial Engineering and Operations Management*, 2015, pp. 1–7.
9. R. Popli and N. Chauhan, Managing uncertainty of story-points in agile software, in *2nd Int. Conf. Computing for Sustainable Global Development*, 2015, pp. 1357–1361.
10. H. Zahraoui and M. A. J. Idrissi, Adjusting story points calculation in scrum effort time estimation, *10th Int. Conf. Intelligent Systems: Theories and Applications*, 2015, pp. 1–8.
11. V. Lenarduzzi, I. Lunesu, M. Matta and D. Taibi, Functional size measures and effort estimation in agile development: A replicated study, in *Proc. 16th Int. Conf. Agile Processes in Software Engineering and Extreme Programming: XP 2015*, C. Lassenius, T. Dingsøyr and M. Paasivaara (Springer International Publishing, 2015), pp. 105–116.
12. B. Tanveer, L. Guzmán and U. M. Engel, Understanding and improving effort estimation in agile software development: An industrial case study, in *Proc. Int. Workshop on Software and Systems Process*, 2016, pp. 41–50.

13. A. Ramirez-Noriega, R. Juarez-Ramirez, R. Navarro and J. Lopez-Martinez, Using Bayesian networks to obtain the task's parameters for schedule planning in scrum, *4th Int. Conf. Software Engineering Research and Innovation*, 2016, pp. 167–174.
14. B. W. Boehm, Software engineering economics, *IEEE Trans. Softw. Eng.* **SE-10**(1) (1984) 4–21.
15. L. C. Briand and I. Wieczorek, *Resource Estimation in Software Engineering* (John Wiley and Sons, 2002).
16. E. Mendes, *Cost Estimation Techniques for Web Projects* (IGI Global, 2007).
17. A. Trendowicz and R. Jeffery, *Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success* (Springer, 2014).
18. S. Bayona-Oré, J. A. Calvo-Manzano, G. Cuevas and T. San-Feliu, Critical success factors taxonomy for software process deployment, *Softw. Qual. Control* **22**(1) (2014) 21–48.
19. M. Usman, R. Britto, J. Börstler and E. Mendes, Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method, *Inf. Softw. Technol.* **85** (2017) 43–59.
20. P. Bourque *et al.*, *Guide to the Software Engineering Body of Knowledge, Version 3.0* (IEEE Computer Society Press, 2014).
21. M. Cohn, *Agile Estimating and Planning* (Pearson Education, 2005).
22. B. H. Kwasnik, The role of classification in knowledge representation and discovery, *Libr. Trends* **48**(1) (1999) 22–47.
23. V. Broughton, *Essential Classification* (Facet Publishing, 2004).
24. D. Šmite, C. Wohlin, Z. Galviņa and R. Prikladnicki, An empirically based terminology and taxonomy for global software engineering, *Empir. Softw. Eng.* **19**(1) (2014) 105–153.
25. T. Dybå, D. I. Sjøberg and D. S. Cruzes, What works for whom, where, when and why?: On the role of context in empirical software engineering, in *Proc. ACM-IEEE Int. Symp. Empirical Software Engineering and Measurement*, 2012, pp. 19–28.
26. O. Benediktsson and D. Dalcher, Effort estimation in incremental software development, *IEE Proc., Softw.* **150**(6) (2003) 351–357.
27. L. Chung, B. A. Nixon, E. Yu and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*, Vol. 5 (Springer Science and Business Media, 2012).
28. B. W. Boehm *et al.*, *Software Engineering Economics*, Vol. 197 (Prentice-Hall, Englewood Cliffs, 1981).
29. N. C. Haugen, An empirical study of using planning poker for user story estimation, in *Proc. AGILE Conf.*, 2006, pp. 23–34.
30. P. Abrahamsson, I. Fronza, R. Moser, J. Vlasenko and W. Pedrycz, Predicting development effort from user stories, in *Proc. Int. Symp. Empirical Software Engineering*, 2011, pp. 400–403.
31. S. Bhalerao and M. Ingle, Incorporating vital factors in agile estimation through algorithmic method, *Int. J. Comput. Sci. Appl.* **6**(1) (2009) 85–97.
32. C. Catal and M. S. Aktas, A composite project effort estimation approach in an enterprise software development project, in *Proc. 23rd Int. Conf. Software Engineering and Knowledge Engineering*, 2011, pp. 331–334.
33. A. Parvez, Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development, in *Proc. Int. Conf. Information Technology and Electrical Engineering*, 2013, pp. 113–118.
34. D. Sande, A. Sanchez, R. Montebelo, S. Fabbri and E. M. Hernandez, A strategy to support software planning based on piece of work and agile paradigm, in *Enterprise Information Systems*, Lecture Notes in Business Information Processing, Vol. 73 (Springer, 2011), pp. 104–118.

35. V. Mahnič and T. Hovelja, On using planning poker for estimating user stories, *J. Syst. Softw.* **85**(9) (2012) 2086–2095.
36. L. Cao, Estimating agile software project effort: An empirical study, in *Proc. 14th Americas Conf. Information Systems*, 2008, pp. 401:1–401:10.
37. A. Forward and T. C. Lethbridge, A taxonomy of software types to facilitate search and evidence-based software engineering, in *Proc. Conf. Center for Advanced Studies on Collaborative Research: Meeting of Minds*, 2008, pp. 14:179–14:191.
38. K. Petersen and C. Gencel, Worldviews, research methods and their relationship to validity in empirical software engineering research, in *Joint Conf. 23rd Int. Workshop on Software Measurement and 8th Int. Conf. Software Process and Product Measurement*, 2013.