Review Article Software Effort Estimation in Agile Methodolgy

by Pantjawati Sudarmaningtyas

Submission date: 04-Jul-2020 01:30PM (UTC+0700)
Submission ID: 1353299972
File name: ICLE_SOFTWARE_EFFORT_ESTIMATION_IN_AGILE_METHODOLOGY_SUBMIT.docx (634.66K)
Word count: 7060
Character count: 43135

REVIEW ARTICLE SOFTWARE EFFORT ESTIMATION IN AGILE METHODOLOGY

REVIEW ARTICLE SOFTWARE EFFORT ESTIMATION IN AGILE METHODOLOGY

Pantjawati Sudarmaningtyas[1,2], Rozlina Binti Mohamed[2*]

[1]Department of Information System, Universitas Dinamika, 60298 Surabaya, Jawa Timur, Indonesia
[2]Faculty of Computing, Universiti Malaysia Pahang, 26300 Gambang, Kuantan, Pahang, Malaysia

pantja@dinamika.ac.id (Pantjawati Sudarmaningtyas), +62818399171
rozlina@ump.edu.my (Rozlina Binti Mohamed), +60199422871
**Corresponding Author*

List of Table / Figure:

Table 1.
Table 2.
Table 3.
Table 4.
Table 5.
Table 6.
Table 7.
Table 8.
Table 9.
Figure 1.
Figure 2.
Figure 3.
Figure 4.
Figure 5.

REVIEW ARTICLE SOFTWARE EFFORT ESTIMATION IN AGILE METHODOLOGY

ABSTRACT

Currently, the Agile software development method has been commonly used in software development projects, and the success rate higher than waterfall projects. Despite, effort estimation in Agile is still a challenge because most existing effort estimation methods developed based on the conventional software development method. The objectives of this study are to find out the software effort estimation method applied in Agile, the approach of implementing, and the attributes that affect the effort estimation. The results of the literature review show the top three software effort estimation applied in Agile, and those are machine learning (37%), Expert Judgement (26%), and Algorithmic (21%). The implementation of all machine learning methods using a hybrid approach, which are a combination of machine learning and expert judgment, or a mix of two or more machine learning. Despite, implementation of the effort estimation through a hybrid approach only used in 47% of relevant articles. Effort estimation in Agile involving 24 attributes, where Complexity, Experience, Size, and Time attributes are the most commonly used and implemented in all those methods of effort estimation.

Keywords: Agile, Software Effort Estimation

INTRODUCTION

In the software development projects are widely used Agile software development methods, especially Scrum methodology with iteration planning techniques. It is compliant with the results of a survey that showed 94% of respondents employ Agile, and 60% have more than three years of experience in Agile. Scrum is the most common agile methodologies used by respondents' organizations (58%), while the top agile techniques are iteration planning (90%) (VersionOne.com, 2017). Besides, statistically indicate that the success rates of the Agile project are two times more likely to succeed, and one-third less likely to fail than waterfall projects (Mersino, 2018).

Based on a survey found that 45% of IT projects transcend budget because it is not established according to the factual requirement (Bloch, Blumberg, & Laartz, 2012). The estimated cost of software development is essential to avoid excessive costs. In general, estimated costs are based on effort estimation (Bloch et al., 2012). Therefore, effort estimation is a crucial part of the software development project.

The objective of the estimation is to provide an approximation of the resources needed to complete a project so being able delivering outputs in the form of products or services that accordance with the specified characteristics of functional and non-functional (Institute, 2017). Estimates are usually internally generated and conducted periodically. The early estimation of effort, schedule, and cost is a repetitive work to be compromised and review between stakeholders to reach an agreement regarding the requirement of resources and time to completion of projects (Bourque & Fairley, 2014).

The measurement of the precision of a single value effort estimate is not straightforward, so the value of effort estimates better be communicated in the interval. Although not connected between the implied effort intervals and confidence level but estimator confident that there is a possibility the actual efforts it will be inside the range (Jørgensen, 2016).

The effort estimate can be interpreted from the requirement of resources such as people and tools (Bourque & Fairley, 2014). The effort is a composite of person and time, which indicates the number of thoroughly productive working hours necessary for someone to get work complete. Units of effort are typically stated in person-hours, person-days, person-months, and person-years (Trendowicz & Jeffery, 2014).

Even though Agile success rate higher than waterfall projects, but estimation effort still to be the main challenges that must be considered. This study aims to review about methods and approaches of software effort estimation to find the most appropriate effort estimation method for Agile software development.

This research is conducted to provide a basis for further development of software effort estimation method to be used in Agile software development. Towards achieving the target as mentioned above, this paper organized into five parts. Beginning with an introduction, followed by overview of effort estimation techniques, materials and methods, results and discussion, and ended by the conclusion.

OVERVIEW OF SOFTWARE EFFORT ESTIMATION METHOD

Currently, many software effort estimation methods are implemented in software development projects such as Constructive Cost Model (COCOMO), Function point Analysis, Source Line of Code (SLOC), SEER-SEM (Software Evaluation and Estimation of Resources-Software Estimation Model), Linear Model, Multiplicative Model, Putnam Model, Brake Down Estimation, Artificial Neural Network, and Fuzzy. The existing software effort estimation methods shown in Figure 1 and three researchers have classified these according to their perspective.

The first researcher (Boehm, 1984) classified software effort estimation methods from the perspective of the technique used. Based on this perspective, classification the effort estimation method divide into seven classes are Algorithmic models, Expert Judgment, Analogy, Parkinson, Price-to-Win, Top-Down, and Bottom-Up.

The second researcher (Srivastava & Wadhwa, 2013) classifying software effort estimation methods based on an algorithm that used. From that point of view obtained four categories of software effort estimation methods viz Algorithmic, Non-Algorithmic, Parametric, and Machine learning. Some popular estimation methods in Algorithmic models: Function point Analysis, Source Line of Code (SLOC), SEER-SEM (Software Evaluation and Estimation of Resources-Software Estimation Model), Linear Model, Multiplicative Model, Putnam Model, Constructive Cost Model (COCOMO). Analogy and Expert Judgment is a part of Non-algorithmic models, while Brake Down Estimation is one of the estimation methods in Parametric models. Estimation methods that include Machine learning models are Artificial Neural Network and Fuzzy.

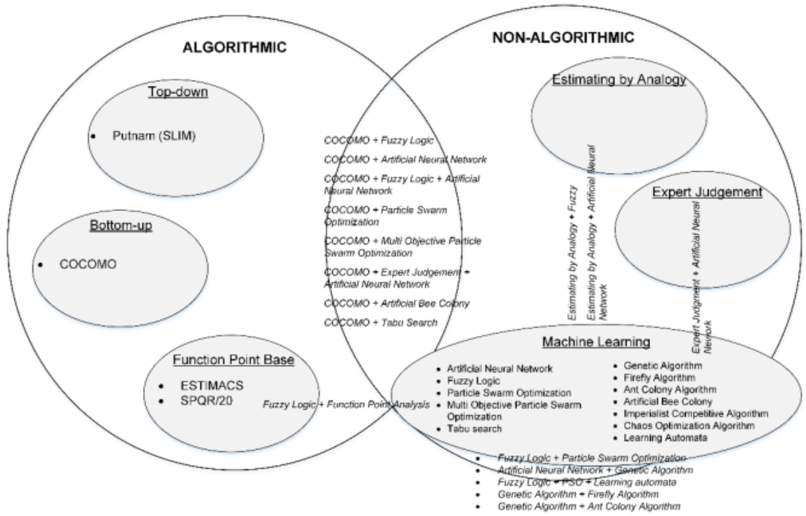

Figure 1 Existing Software Effort Estimation Method

The third researcher (Trendowicz & Jeffery, 2014) classifying software effort estimation methods by the data input type and the principle of estimation that employed. Based on that, software effort estimation methods classified into three categories: Data-driven, Expert-based, and Hybrid. Data-driven has two groups, Proprietary and Non-proprietary. Non-proprietary divided into three classes are Model-based, Memory-based, and Composite. Model-based consist of Parametric, Non-parametric, and Semi-parametric.

Table 1 shows a summary of existing software effort estimation methods classification based on explaining above.

Table 1 Existing Effort Estimation Methods Classification

Classification	(Boehm, 1984)	(Srivastava & Wadhwa, 2013)	(Trendowicz & Jeffery, 2014)
Algorithmic models	√	√	
Expert Judgment/Expert base	√		√
Analogy	√		
Parkinson	√		
Price-to-Win	√		
Top-Down	√		
Bottom-Up	√		
Non-algorithmic		√	
Parametric		√	√
Non-parametric			√
Semi-parametric			√
Machine Learning		√	
Data-driven			√
Hybrid			√

Based on the previous research, this study classifying software effort estimation methods based on three aspects: 1) estimation principle that employed; 2) estimation strategy that applied; 3) data that the required. In general, each of these aspects divided into two parts. Estimation principle issue consists of two, namely algorithmic and non-algorithmic. The facet of the strategy is divided into two types, Top-Down and Bottom-Up strategy. While the data aspect indicate that some methods have a high dependency on historical data, but others did not need historical data. Figure 2 shown classification that uses in this study and existing software effort estimation methods.

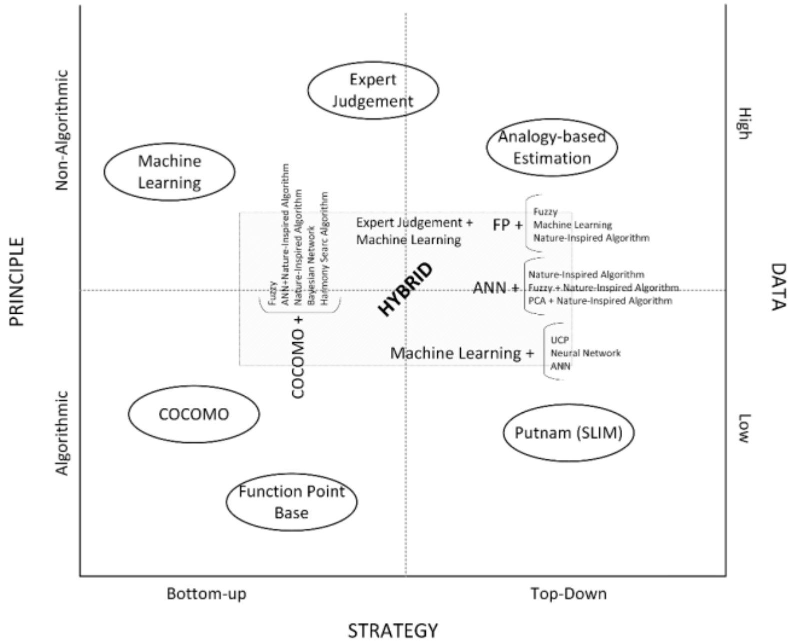

Figure 2 Software Effort Estimation Classification

In the development, the problem in estimation better resolved when using a combination of the several elements, this model called a hybrid. A hybrid model can combine components in a different aspect but also very possible to combine attributes in the aspect of itself. There have been no perfect single estimation methods, so it is more suggested to use multiple estimation methods. Confluence amongst the estimates generated by distinct methods indicates that the estimation most likely accurate. The discrepancy between the estimates indicating the possibility has been the neglect of certain factors. It is essential to find the factors that cause the difference and then reexamine again to converge result so resulted in preferable estimates (Bourque & Fairley, 2014).

MATERIALS AND METHODS

The method to conduct the study literature review is based on (Kitchenham & Charters, 2007) that comprises three main phases, namely planning, conducting, and reporting. The development of the systematic literature review protocol in this study consists of planning and conducting, as shown in Figure 3.

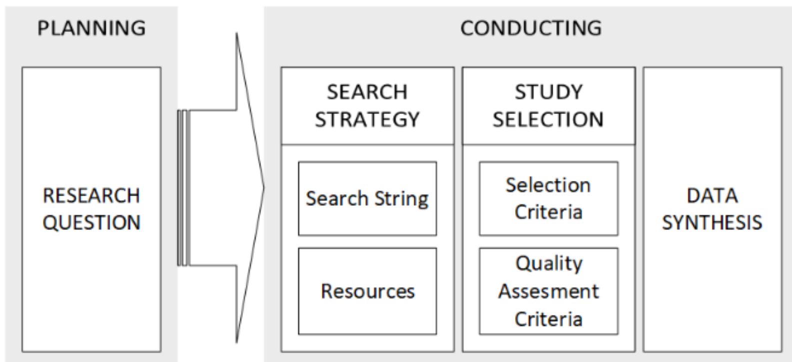

Figure 3 Systematic Literature Review Protocol

Planning.

To achieve this research aims then must be known in advance software effort estimation that is used in Agile project development. Next up is knowing how software effort estimation is implemented and what variables are involved in Agile effort estimations.

Behalf on that, at the stage planning, is formulation research questions related to Agile effort estimation as follows:

RQ#1: What kind of method used to effort estimation in Agile?

RQ#2: How it works to implements estimation effort in Agile?

RQ#3: What are attributes involves in the estimating of effort in Agile?

Conducting

Conducting phase comprises of three activities, (i) search strategy, (ii) study selection, and (iii) data synthesis. In contrast to the data synthesis that independent, the search strategy and study selection each formed of two different activities.

On search strategy, a term that would be used in search of as known as search strings is identified. This phase also determines resources, where the search will be executed. The searches process can be conducted after search strings and resources identified. Search strings used in the search process is "agile" AND ("effort" OR "cost") AND "estimation" AND ("technique" OR "model").

The search process conducted by embedding the search string that had mentioned before on seven resources as shown in Table 2.

Table 2 List of Resources

No.	Source Name	URL
1	IEEExplore	www.ieeexplore
2	ACM Digital library	https://dl.acm.org/
3	Google Scholar	scholar.google.com
4	Inspec	https://digital-library.theiet.org/
5	ScienceDirect	www.sciencedirect.com
6	SpringerLink	www.springerlink
7	World Scientific	https://www.worldscientific.com

Study selection involves two activities. The first selection activity is filtering the result of the search process based on inclusion and exclusion criteria. The second activity is to implement quality assessment criteria on the first activity result.

Inclusion criteria consist of four aspects: publication year, the language that used, the context of research that assessed from title and abstract, and valid DOI. Meanwhile, exclusion criteria determined by two aspects, namely type of work and kind of research. In this case, type of work may be in the form of a journal, conference proceeding, book, book chapter, thesis, dissertation, or course material, while the kind of research consists of study research, literature review, comparative study, or survey. The inclusion and exclusion criteria explained in Table 3.

Table 3 Inclusion and Exclusion Criteria

Inclusion Criteria	Exclusion Criteria
Publication Year: 2009 – 2019	Type of work: Book, Book Chapter, Thesis, Dissertation, course material
Language: English	Kind of Research: Literature Review, Comparative Study, Survey, Report
Title or Abstract: Contains search string	
DOI: valid DOI	

After the first activity performed, the next step is selection using quality assessment criteria. As seen in Table 4, quality assessment criteria derived from the research question. The relevant studies are fulfilled the quality assessment criteria with an accumulated score value greater than 1.

Table 4 Quality Assessment Criteria

<table>
<tr><th>ID</th><th>Question (Q)</th><th>Answer/
Score Points</th></tr>
<tr><td>RQ#1</td><td>Does the study discuss the kind of technique that used to estimate effort in Agile?</td><td>Yes | Partially | No /
1 | 0.5 | 0</td></tr>
<tr><td>RQ#2</td><td>Does the study explain the approach to implements the estimating effort methods in Agile?</td><td>Yes | Partially | No /
1 | 0.5 | 0</td></tr>
<tr><td>RQ#3</td><td>Does the study declare attributes affecting the estimating of effort in Agile?</td><td>Yes | Partially | No /
1 | 0.5 | 0</td></tr>
</table>

Data synthesis is an activity to resume the selected studies evidence to synchronized with the research question. The primary purpose of this activity is to make it clear answers to research questions revealed.

RESULTS AND DISCUSSION

Conducting Phase Result.

The result of the conducting phase from seven resources gets 171 articles. The number of articles for each resource shown in Table 5. The top three articles get from Google Scholar (30.99%), SpringerLink (28.07%), and IEEExplore (22.22%).

Table 5 Result Conducting Phase

No.	Source Name	URL	Number of Articles
1	IEEExplore	www.ieeexplore	38
2	ACM Digital library	https://dl.acm.org/	15
3	Google Scholar	scholar.google.com	53
4	Inspec	https://digital-library.theiet.org/	1
5	ScienceDirect	www.sciencedirect.com	14
6	SpringerLink	www.springerlink	48
7	World Scientific	https://www.worldscientific.com	2
TOTAL			171

The inclusion criteria have implemented to get 118 articles and exclusion criteria to get 100 articles. Table 6 shows the result of the inclusion and exclusion criteria.

Table 6 Result of Inclusion and Exclusion Criteria

No.	Source Name	URL	Number of Articles	
			Inclusion Criteria	Exclusion Criteria
1	IEEExplore	www.ieeexplore	37	33
2	ACM Digital library	https://dl.acm.org/	15	15
3	Google Scholar	scholar.google.com	11	5
4	Inspec	https://digital-library.theiet.org/	1	0
5	ScienceDirect	www.sciencedirect.com	13	12
6	SpringerLink	www.springerlink	40	34
7	World Scientific	https://www.worldscientific.com	1	1
TOTAL			118	110

Table 7 shows the number of articles of each resource generated from the quality assessment criteria.

Table 7 Result of Quality Assessment Criteria

No.	Source Name	URL	Number of Articles
1	IEEExplore	www.ieeexplore	16
2	ACM Digital library	https://dl.acm.org/	6
3	Google Scholar	scholar.google.com	4
4	Inspec	https://digital-library.theiet.org/	0
5	ScienceDirect	www.sciencedirect.com	5
6	SpringerLink	www.springerlink	7
7	World Scientific	https://www.worldscientific.com	0
TOTAL			38

Data Synthesis Phase Result.

Effort Estimation Method That Implemented In Agile.

According to the relevant articles, effort estimation method that implements in Agile can classified into four (4) types, these are Expert Judgement (EJ), Algorithmic (A), Machine Learning (ML), and Statistic (St).

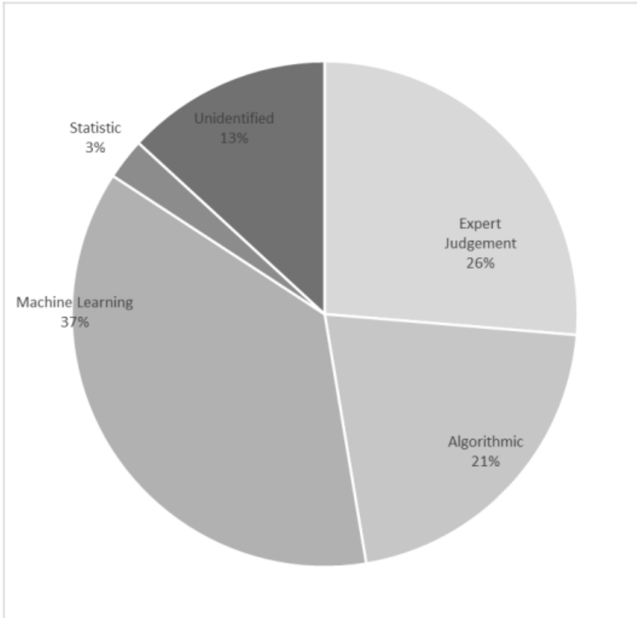

Figure 4 Distribution of Software Effort Estimation Method in Agile

Figure 4 shows the distribution of the software effort estimation method applied in Agile. From the relevant articles, most of the articles (87%) clearly stated the effort estimation method that uses, and the rest (13%) not described the effort estimation method that uses. The top three software effort estimation methods are Machine Learning (37%), Expert Judgement (26%), and Algorithmic (21%).

Expert Judgement Method.

All the EJ effort estimation method implements in Agile is based on Planning Poker. Planning Poker is a group estimation technique recommended for agile software development methods. The group discussion in this technique is assumed can provide higher accuracy of estimation and reduce excessive optimism that is characteristic of expert judgment-based methods, although little empirical evidence about it. (Mahnič & Hovelja, 2012) conduct research to fill the gaps by comparing the estimating effort based on the same user stories of the student groups and the expert groups. The comparing result indicates when the people involved in the group estimation process have the expertise that raises so that the optimism bias arising from group discussion can be diminished even disappeared.

(Lenarduzzi, Lunesu, Matta, & Taibi, 2015) state that the estimation efforts conducted by the developers have higher accuracy than that achieved thru measures the functional size. Besides, estimation accuracy in Scrum cannot improve with the help of SiFP and IFPUG Function Points because of weak predictive power. The statement was obtained after two times of replication original study through applied a proper replication to two plain Scrum development processes.

The two research results above reveal that the professional's expertise in software development companies is the primary aspect that influences the accuracy of estimation through Planning Poker. (L´opez-Mart´ınez, Ram´ırez-Noriega, Ju´arez-Ram´ırez, Licea, & Mart´ınez-Ram´ırez, 2017) strengthen the statement by conduct research that proves the existence of dissent between Scrum experts and students on the following factors: experience, time, effort, priority, and user stories value. From the five factors in the differences in opinions upon the factors of experience and time, while effort and priority have similar opinions. For the user stories value factor, many people doubted considering this factor as necessary, and some of them are not disagreeing to apply this factor in the planning phase. In line with this, the industry not even likely to consider this factor.

Although formerly research shows the existence of different opinions between Scrum experts and students (Chatzipetrou, Ouriques, & Gonzalez-Huerta, 2018) shows that the students were more engaged with the Planning Poker than estimates the user stories by applied knowledge from the course.

During the Planning Poker activity, the students more interested and involved. It indicates that the estimation with planning poker is a fun activity.

The user story is measured with relative value as known as Story points (SP) measure that commonly uses as the base of calculation in the Planning Poker. (Zahraoui & Janati Idrissi, 2015) improve the accuracy of effort and time estimates with proposing Adjusted Story Point (ASP) measure instead of SP measure to calculate the total effort of a scrum project. ASP using three adjustment factors are Priority Factor (PF), Story Size Factor (SF), and Complexity Factor (CF). Unfortunately, the use of ASP not yet apply to reel scrum projects, thereby need further research to improve the proper values of Story Point Adjustment Factors (SPF).

Although Planning Poker has a lot of benefits, the result was relying on observation experts. To overcome these issues, (López-Martínez, Juárez-Ramírez, Ramírez-Noriega, Licea, & Navarro-Almanza, 2017) proposed a new model to improved Planning Poker through estimated the complexity and importance of user stories using Bayesian Network. Even though it shows good results primarily to facilitate newbie developers in deciding when they estimate user stories, this model still needs expert knowledge to build a Bayesian Network.

(López-Martínez, Ramírez-Noriega, Juárez-Ramírez, Licea, & Jiménez, 2018) validating the model that was built on previous studies using the Bayesian network that considers user stories based on its complexity and the level of importance through collect estimation correlated with the proposed model from students and professionals. The validation result shows that the estimation from professionals has a higher degree of correlation than student's estimates. It indicates that factors in the real-world application also considered in the proposed model. Despite showing promising results, the proposed model must be tested on a whole real project. Besides, it needed a mobile application development to ease its implementation.

(Tanveer, Vollmer, & Engel, 2017) is proposing an innovative hybrid method that incorporates expert knowledge and changes impact analysis to overcome hugely impacting the process of effort estimation caused by additional functionality to the system incrementally and iteratively. That hybrid method is furnished by (Tanveer, Vollmer, & Braun, 2018) with a prototype tool built based on the framework made previously. That hybrid method can improve the effectiveness of effort estimates.

(Moharreri, Sapre, Ramanathan, & Ramnath, 2016) complementing manual PP with an automated estimation through extracted Agile story cards and their actual effort time. The Auto-Estimate developed by comparing alternate methods like Naive Bayes, Random Forest, J48, and Logistic Model Tree (LMT), where the better result gets from the combination of J48+PP, J48, and LMT+PP.

Algorithmic Method.

The algorithmic effort estimation method in Agile software development consists of the COSMIC, phase-wise algorithm, Function Point (FP), and Use Case Point (UCP). This part explains the relevant articles that implement the algorithmic software effort estimation method.

COSMIC proposed by (Desharnais, Buglione, & Kocatürk, 2011) to improve the guess estimate in Agile Project Management (APM). They propose a new procedure that built by consolidating the COSMIC measurement method at the micro-level (User Stories) and the quality of the documentation for functional analysis deployment. Their study indicates that this approach can help the planner to know better why the global effort changes by the time.

The Phase-wise algorithm to computation the estimation effort is proposed by (Choudhari & Suman, 2012a) and (Choudhari & Suman, 2012b) that conducted an empirical study through a questionnaire to determine efforts of maintenance to compute phase-wise effort estimation. Their proposed provides more realistic results and worthwhile in estimating maintenance effort, especially in extreme programming based on maintenance environments. Even so, it still needs refinement on the proposed technique.

Function Point (FP) is a part of algorithmic effort estimation classification use by (Kang, Choi, & Baik, 2010) as an addition to the story point for agile projects systematic estimation. FP uses to minimize fluctuations of value estimation caused by relative values from the user story. Based on a comparison with traditional methods, the addition story point with FP showed better performance.

Excellent performance of FP applied by (Silas, Yusuf, & Bijik, 2017) to improve cost estimation in Agile software development by proposed the hybridization of Class Responsibility Collaborators models with FP. The estimation process agile software development capable of being boost through the adoption of the hybridization of Class Responsibility Collaborators models with FP.

A use case point (UCP) is an effort estimation method that included in the Algorithmic classification. (Khatri, Malhotra, & Johri, 2016) implemented UCP, as an estimation method, for Agile software development in early-stage by emphasizing on main complexity factors like technical and environmental. In this context, the uses of the UCP can estimate effort adequately and improve the accuracy based on environment and technical factors under agile development.

UCP can also collaborate with the Scrum Framework by connecting between User Stories on the Product Backlog and Use Case on UCP as done by (Yuliansyah, Qudsiah, Zahrotun, & Arfiani, 2018). In their study, UCP is required to adjust User Storie on Product Backlog through adding the transactions attribute on a user story, and thereby User Story attributes transform to Functionality.

(Popli & Chauhan, 2013) proposing algorithmic estimation method by considers various factors to be increased estimation accuracy of release date, cost, effort, and duration, especially for the project Scrum. The algorithm uses Sprint-Point Based Estimation Framework for Agile Software and involves two factors: project and people-related factors proven effective and feasible. Although these factors show strong influences on the estimation value of estimation sprints point, in the future, still can be added other factors that most affect the estimation.

Machine Learning Method.

Most of the relevant articles have the objective of improving accuracy, as a crucial issue in Agile effort estimation, by machine learning. Some of it is applying machine learning to resolve the metric size of the user story, which is commonly used as the base of effort estimation in Agile. The rest of the relevant articles, using machine learning to overcome frequent the change of the requirement, as one of the characteristics of Agile software development.

To improve the accuracy of Agile effort estimation, a combination of Neural Network proposed by (Panda, Satapathy, & Rath, 2015a), (Panda, Satapathy, & Rath, 2015b), and (Prasada Rao et al., 2018). While (Malgonde & Chari, 2018) applied predictive algorithms with ensemble-based approaches that consist of Support Vector Machine (SVM), Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN), Decision Trees (DT).

Beside Neural Network, the accuracy of Agile effort estimation also improved using metaheuristic algorithms. (Khuat & Le, 2017) devise a hybrid estimation model to enhance accuracy in Agile effort estimation by applied Particle swarm optimization (PSO) and artificial bee colony (ABC). A novel formula for agile software effort estimation based on velocity and story points is built from two metaheuristic algorithms mentioned before. However, this hybrid estimation model needs further research on its implementation.

Issue accuracy in Agile effort estimation also improved using the ontology model. (Adnan & Afzal, 2017) proposing a model to build a knowledge base by saving significant tacit knowledge during project development. Various agents of the estimation system access the existing knowledge base and autonomously perform a suitable estimate for the success of future projects.

Commonly Planning Poker (PP), one of the effort estimations in Agile, is based on a user story that measures by relative values. Some studies overcome this problem by proposing models that extracted keywords from user stories automatically.

(Abrahamsson, Fronza, Moser, Vlasenko, & Pedrycz, 2011) proposed a model Agile effort prediction model based on user stories using keyword extraction tools, while (Kowalska & Ochodek, 2014) proposed a new approach through Semantic Web technologies. On the other hand, (Choetkiertikul et al., 2016) introduced a novel combination of two robust deep learning architectures: long short-term memory and recurrent highway network to estimating story points.

(Chongpakdee & Vatanawood, 2017) applied document fingerprints, to retrieve the similar issues from the public repository of project management assets. In addition to the extraction of the keyword on a user story, (Dragicevic, Celar, & Turic, 2017) proposed to implement the Bayesian Network model.

To handling frequent requirement changes in Agile software development, two studies had been conducted to resolve this issue. (Bilgaiyan, Mishra, & Das, 2018) applied Artificial neural networks (ANNs) in Agile effort estimation. ANN-feedforward back-propagation neural network and Elman neural network are used to ease of keep track, maintain, and the estimate of the whole product. While (Soares, 2018) proposes to embed autoencoders in automatic software development effort estimation as text classification.

Even though all those studies provide many advantages, those models still need more massive datasets and other features because user stories not only written in English and sometimes influence by developers' demographics, story criticality, and other systems and framework aspects.

Improved Agile software effort estimation is conducted by proposing a model based on support vector regression (SVR) optimized by the grid search method (GS). Empirical evaluation through the leave-one-out cross-validation method against 21 historical agile software projects demonstrates that this model affords increases the performance of the SVR technique (Zakrani, Najm, & Marzak, 2018).

Statistic Method.

To make the effort estimation method from the conventional life cycle model accordance with Agile software development, (Garg & Gupta, 2015) proposed a new model by implemented Principal Component Analysis (PCA) to find the key attributes of the development cost in Agile. The study found that the proposed methodology is suitable for Agile projects in the scope of the satisfaction of Agile manifestos.

Approach For Estimating Effort In Agile.

Implementation of software effort estimation could be done through two approaches, Non-hybrid and Hybrid. The non-hybrid approach applies a single effort estimation method, whereas the hybrid approach implements a combination of several effort estimation methods. Half of the relevant articles use the non-hybrid approach, 36.84% implement the hybrid approach, while the rest did not mention the approach that was used.

The non-hybrid approach consisting of effort estimation methods such as Planning Poker, Phase Wise, COSMIC, Function Points, Use Case Point, Document Fingerprints, Bayesian Network, Text Classification, Ontology Model, and Principal Component Analysis. Hybrid approaches contain combination methods between Expert Judgement and Statistic, Expert Judgement and Machine Learning, and Machine Learning and others Machine Learning. Most of the hybrid approach is a mix of machine learning techniques. Table 8 shown the methods and approach that applied in Agile.

Table 8 Types of software effort estimation method and approach

Approach	Method	Technique	Author
Non-hybrid	EJ	Planning Poker	(Chatzipetrou et al., 2018), (L´opez-Mart´ınez et al., 2017), (Lenarduzzi et al., 2015), (Mahnič & Hovelja, 2012), (Zahraoui & Janati Idrissi, 2015)
	A	Phase wise	(Choudhari & Suman, 2012a), (Choudhari & Suman, 2012b)
		COSMIC	(Desharnais et al., 2011)
		FP	(Kang et al., 2010)
		UCP	(Yuliansyah et al., 2018)
	ML	Document Fingerprints	(Chongpakdee & Vatanawood, 2017)
		Bayesian Network	(Dragicevic et al., 2017)
		Classification	(Soares, 2018)
		Ontology	(Adnan & Afzal, 2017), (Kowalska & Ochodek, 2014)
	St	PCA	(Garg & Gupta, 2015)
Hybrid	EJ, St	EJ and Impact Analysis	(Tanveer, Guzman, & Engel, 2017), (Tanveer et al., 2018)
	EJ, ML	Planning Poker and Machine learning (J48, LMT, Bayesian Network)	(Moharreri et al., 2016), (López-Martínez et al., 2017), (López-Martínez et al., 2018)
	ML & ML	ABC and PSO	(Khuat & Le, 2017)
		Adaptive Neuro-Fuzzy Modelling, Generalized Regression Neural Network and Radial Basis Function Networks (RBFNs)	(Prasada Rao et al., 2018)
		Combination of Neural Networks	(Bilgaiyan et al., 2018), (Panda et al., 2015a), (Panda et al., 2015b)
		Combination two deep learning	(Choetkiertikul et al., 2016)
		Combination of Machine Learning such as SVM, SVR, ANN, KNN, and DT.	(Abrahamsson et al., 2011), (Malgonde & Chari, 2018), (Zakrani et al., 2018)

Attributes That Affect Estimating Effort In Agile.

Volatility and change of the customer requirement in Agile software development (ASD) is a difficult and challenging task in estimation effort. Generally, effort estimation in ASD is mainly based on user story (US) that measures by story points (Zahraoui & Janati Idrissi, 2015). The US are commonly estimated using group processes that improved estimation accuracy compared to individual estimation process (Moløkken-Østvold & Jørgensen, 2004).

Most relevant articles (84.21%) explain the attributes that use on effort estimation, and 15.79% did not mention the attributes specifically. Based on the relevant articles, effort in Agile Software Development can be measured by user story, use case with sizing method story points, use case points, and function points. Nevertheless, the attribute of being used in the effort estimation can vary significantly.

Table 9 Effort Estimation Attributes

No.	Attribute	SEE Method				Frequency of use									
		EJ	A	ML	St	2010	2011	2012	2013	2014	2015	2016	2017	2018	Total
1	Application Type				■						1				1
2	Architecture				■						1				1
3	Complexity	■	■	■	■					3	1	3	5	2	14
4	Dependencies				■							1	1	1	3
5	Effort	■		■						1	2		2	2	7
6	Environmental		■											1	1
7	Experience	■		■	■							2	4	2	8
8	Function Point	■	■		■		1				2				3
9	Impact	■			■							1	1		2
10	Knowledge	■			■							1	1		2
11	Maturity				■						3				3
12	Platform				■						3				3
13	Previously Estimates	■			■							1	1		2
14	Priority	■		■									1	2	3
15	Size	■		■	■	1					1		1	5	8
16	Skill			■									1		1
17	Story Points			■						5	2		1	2	10
18	Sprint Points			■					3					1	4
19	Task	■		■	■					1				2	3
20	Time	■	■	■			1		1				3	1	6
21	Use Case Point		■									1		1	2
22	User Story	■	■	■			1	3		1	1	1	1		8
23	Velocity			■					1		2		1	1	5
24	Weight	■	■	■	■									2	3

According to Table 9, there are 24 attributes used in effort estimation. Agile software effort estimation classification that the most employ of attributes is Statistic, followed by Expert Judgment and Machine Learning. The attributes grouped by three criteria: 1) the frequency of use, 2) Implementation on Agile software effort classification, 3) the recent frequency of uses (last three years). Figure 5 showing the mapping of attributes based on those criteria.

The top five highest frequency attributes are Complexity, Story Points, Experience, Size, User Story, Effort, and Time. The implementation of attributes on the Agile software effort classification indicates that the

Complexity attribute implemented on all Software effort estimation classification. Whereas, Experience, Function Point, Size, Task, Time, User Story, Weight are attributes applied to three software effort estimation classification. In the last three years, the most attribute that uses in Agile software effort estimation is Complexity, followed by Experience, Size, Effort, and Time attributes. Attributes that fulfilment all criteria are Complexity, Experience, Size, and Time.

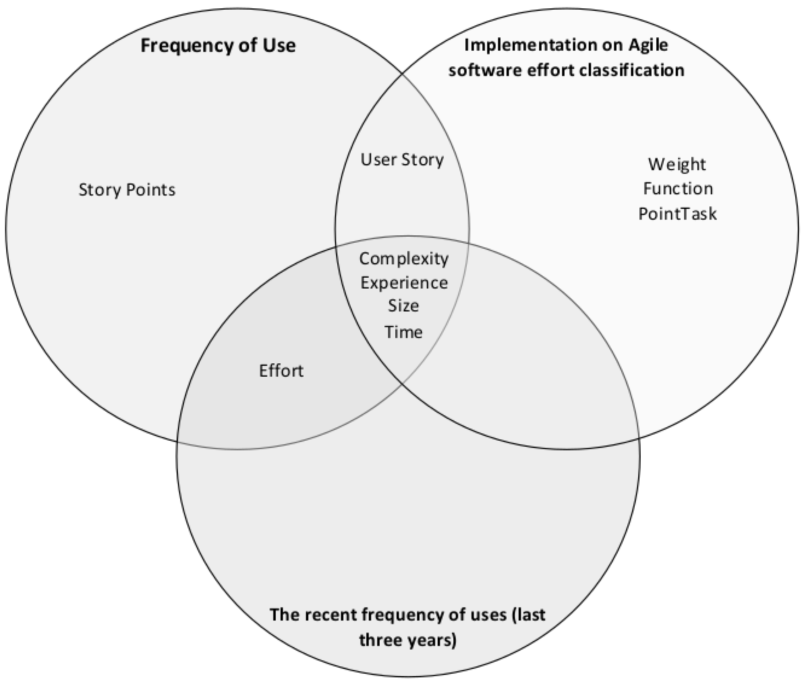

Figure 5 Mapping of Agile software effort estimation attributes

Complexity attribute interpreted in different aspects, and most study understands the complexity attribute as the complexity of the project (Popli & Chauhan, 2014b), (Garg & Gupta, 2015), (Tanveer, Guzmán, & Engel, 2016), (Tanveer, Guzman, et al., 2017), (Bilgaiyan et al., 2018). Some research uses it to represent technical complexity (Hamouda, 2014), (Khatri et al., 2016), (Yuliansyah et al., 2018). Environmental complexity is a part of the complexity attribute that applies in two research (Hamouda, 2014), (Khatri et al., 2016). The Complexity attribute also considered representing form complexity, function complexity, report Complexity, and requirements complexity (Dragicevic et al., 2017).

Many studies using story point attribute that represent itself (Popli & Chauhan, 2014a), (Hamouda, 2014), (Panda et al., 2015a), (Panda et al., 2015b), (Khuat & Le, 2017), (Zakrani et al., 2018), (Prasada Rao et al., 2018). While others apply this attribute to show baseline story-points, estimated story-points (ESP), and unadjusted value of story-points (Popli & Chauhan, 2014b).

Attribute experience in most studies used to measure the implementation experience of the developer or programmer (Tanveer et al., 2016), (Tanveer, Guzman, et al., 2017), (L´opez-Mart´ınez et al., 2017), (López-Martínez et al., 2017), (López-Martínez et al., 2018), (Malgonde & Chari, 2018). Other studies asses this attribute based on experience developers in making estimation programmer (Tanveer et al., 2016), (Tanveer, Guzman, et al., 2017).

Some studies apply size attribute that represents 1) the value of the user story (López-Martínez et al., 2017), (López-Martínez et al., 2018); 2) code metrics for each affected class (Tanveer et al., 2018); 3) or team size (Garg & Gupta, 2015). But other studies did not detail the size attribute (Kang et al., 2010), (Bilgaiyan et al., 2018), (Malgonde & Chari, 2018), (Tanveer et al., 2018).

Effort estimation in agile software development based mainly on the user story attribute (Choudhari & Suman, 2012a), (Choudhari & Suman, 2012b), (Mahnič & Hovelja, 2012), (Popli & Chauhan, 2014a), (Zahraoui & Janati Idrissi, 2015), (Chongpakdee & Vatanawood, 2017), (Choetkiertikul et al., 2016). In (Abrahamsson et al., 2011), the user story elaborated detailed into keywords, length, and priority.

Attribute effort can be assumed as effort per person (Popli & Chauhan, 2014a) or actual effort (Panda et al., 2015a), (Panda et al., 2015b). However, The attribute effort also represents itself without supplement (L´opez-Mart´ınez et al., 2017), (López-Martínez et al., 2017), (López-Martínez et al., 2018), (Malgonde & Chari, 2018).

Commonly, the attribute of time refers to the time it takes to complete a project (Popli & Chauhan, 2013), (L´opez-Mart´ınez et al., 2017), (López-Martínez et al., 2017), (López-Martínez et al., 2018). In some studies, the attribute of time explicitly explained as Person-hours (Desharnais et al., 2011) or Working Hours (Dragicevic et al., 2017).

CONCLUSION

Machine learning is the most effort estimation method uses in Agile software development, followed by Expert judgment and Algorithmic. Even though many used, machine learning has the limitation in implementation because of needs very large dataset that source from expert's knowledge.

The implementation approach of the estimation effort in Agile software development does not differ between the non-hybrid and hybrid. The non-hybrid approach applied on 50.00% of relevant articles and the hybrid approach performed on 36.84% articles.

Twenty-four attributes are involved in Agile effort estimation. The attributes with the top five highest frequency of use are Complexity, Story Points, Experience, Size, User Story, Effort, and Time. The attribute that implemented on all classification of the Agile effort estimation is Complexity. Attributes with the highest frequency used in the last three years are Complexity, Experience, Size, Effort, and Time attributes. Attributes that fulfilment all criteria are Complexity, Experience, Size, and Time.

REFERENCES

Abrahamsson, P., Fronza, I., Moser, R., Vlasenko, J., & Pedrycz, W. (2011). Predicting Development Effort from User Stories. *2011 International Symposium on Empirical Software Engineering and Measurement*, 400–403. https://doi.org/10.1109/ESEM.2011.58

Adnan, M., & Afzal, M. (2017). Ontology Based Multiagent Effort Estimation System for Scrum Agile Method. *IEEE Access*, *5*, 25993–26005. https://doi.org/10.1109/ACCESS.2017.2771257

Bilgaiyan, S., Mishra, S., & Das, M. (2018). Effort estimation in agile software development using experimental validation of neural network models. *International Journal of Information Technology*, 1–5. https://doi.org/10.1007/s41870-018-0131-2

Bloch, M., Blumberg, S., & Laartz, J. (2012). Delivering Large Scale IT.pdf. Retrieved October 23, 2017, from https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/delivering-large-scale-it-projects-on-time-on-budget-and-on-value

Boehm, B. W. (1984). Software Engineering Economics. *IEEE Transactions on Software Engineering*, *SE-10*(1), 4–21. https://doi.org/10.1109/TSE.1984.5010193

Bourque, P., & Fairley, R. E. (2014). Guide to the Software Engineering - Body of Knowledge. In *IEEE Computer Society*. https://doi.org/10.1234/12345678

Chatzipetrou, P., Ouriques, R., & Gonzalez-Huerta, J. (2018). Approaching the Relative Estimation Concept with Planning Poker. *Proceedings of the 7th Computer Science Education Research Conference on ZZZ - CSERC '18*, 21–25. https://doi.org/10.1145/3289406.3289409

Choetkiertikul, M., Dam, H. K., Tran, T., Pham, T. T. M., Ghose, A., & Menzies, T. (2016). A deep learning model for estimating story points. *ArXiv Cs.SE*, *9*, 00489. https://doi.org/10.1109/TSE.2018.2792473

Chongpakdee, P., & Vatanawood, W. (2017). Estimating user story points using document fingerprints. *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 149–152. https://doi.org/10.1109/ICSESS.2017.8342885

Choudhari, J., & Suman, U. (2012a). Phase wise effort estimation for software maintenance. *Proceedings of the CUBE International Information Technology Conference on - CUBE '12*, 397. https://doi.org/10.1145/2381716.2381790

Choudhari, J., & Suman, U. (2012b). Story Points Based Effort Estimation Model for Software Maintenance. *Procedia Technology*, *4*, 761–765. https://doi.org/https://doi.org/10.1016/j.protcy.2012.05.124

Desharnais, J.-M., Buglione, L., & Kocatürk, B. (2011). Using the COSMIC Method to Estimate Agile User Stories. *Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement*, 68–73. https://doi.org/10.1145/2181101.2181117

Dragicevic, S., Celar, S., & Turic, M. (2017). Bayesian network model for task effort estimation in agile software development. *Journal of Systems and Software*, *127*, 109–119. https://doi.org/10.1016/j.jss.2017.01.027

Garg, S., & Gupta, D. (2015). PCA based cost estimation model for agile software development projects. *2015 International Conference on Industrial Engineering and Operations Management (IEOM)*, 1–7. https://doi.org/10.1109/IEOM.2015.7228109

Hamouda, A. E. D. (2014). Using Agile Story Points as an Estimation Technique in CMMI Organizations. *2014 Agile Conference*, 16–23. https://doi.org/10.1109/AGILE.2014.11

Institute, P. M. (2017). A Guide to the Project Management Body of Knowledge (PMBOK® Guide)-Sixth Edition. In *Project Management Institute, Inc.* (Sixth Edit). Pennsylvania: Project Management Institute, Inc.

Jørgensen, M. (2016). The Use of Precision of Software Development Effort Estimates to Communicate Uncertainty. *8th International Conference on Software Quality Days (SWQD)*, 156–168. https://doi.org/10.1007/978-3-319-27033-3_11

Kang, S., Choi, O., & Baik, J. (2010). Model-Based Dynamic Cost Estimation and Tracking Method for Agile Software Development. *2010 IEEE/ACIS 9th International Conference on Computer and Information Science*, 743–748. https://doi.org/10.1109/ICIS.2010.126

Khatri, S. K., Malhotra, S., & Johri, P. (2016). Use case point estimation technique in software development. In P. Shukla, B and Khatri, SK and Kapur (Ed.), *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)* (pp. 123–128). https://doi.org/10.1109/ICRITO.2016.7784938

Khuat, T. T., & Le, M. H. (2017). A Novel Hybrid ABC-PSO Algorithm for Effort Estimation of Software Projects Using Agile Methodologies. *Journal of Intelligent Systems*, *27*(3), 489–506. https://doi.org/10.1515/jisys-2016-0294

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3.

Engineering, *45*(4ve), 1051. https://doi.org/10.1145/1134285.1134500

Kowalska, J., & Ochodek, M. (2014). Supporting Analogy-based Effort Estimation with the Use of Ontologies. *E-Informatica Software Engineering Journal*, *8*(1), 53–64. https://doi.org/10.5277/e-Inf140104

L´opez-Mart´ınez, J., Ram´ırez-Noriega, A., Ju´arez-Ram´ırez, R., Licea, G., & Mart´ınez-Ram´ırez, Y. (2017). Analysis of Planning Poker Factors between University and Enterprise. *2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 54–60. https://doi.org/10.1109/CONISOFT.2017.00014

Lenarduzzi, V., Lunesu, I., Matta, M., & Taibi, D. (2015). Functional Size Measures and Effort Estimation in Agile Development: A Replicated Study. In *International Conference on Agile Software Development* (pp. 105–116). https://doi.org/10.1007/978-3-319-18612-2_9

López-Martínez, J., Juárez-Ramírez, R., Ramírez-Noriega, A., Licea, G., & Navarro-Almanza, R. (2017). Estimating User Stories' Complexity and Importance in Scrum with Bayesian Networks. In *World Conference on Information Systems and Technologies* (pp. 205–214). https://doi.org/10.1007/978-3-319-56535-4_21

López-Martínez, J., Ramírez-Noriega, A., Juárez-Ramírez, R., Licea, G., & Jiménez, S. (2018). User stories complexity estimation using Bayesian networks for inexperienced developers. *Cluster Computing*, *21*(1), 715–728. https://doi.org/10.1007/s10586-017-0996-z

Mahnič, V., & Hovelja, T. (2012). On using planning poker for estimating user stories. *Journal of Systems and Software*, *85*(9), 2086–2095. https://doi.org/10.1016/j.jss.2012.04.005

Malgonde, O., & Chari, K. (2018). An ensemble-based model for predicting agile software development effort. *Empirical Software Engineering*, 1–

39. https://doi.org/10.1007/s10664-018-9647-0

Mersino, A. (2018). Agile Project Success Rates are 2X Higher than Traditional Projects (2019). Retrieved January 27, 2020, from https://vitalitychicago.com/blog/agile-projects-are-more-successful-traditional-projects/

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

Moløkken-Østvold, K., & Jørgensen, M. (2004). Group Processes in Software Effort Estimation. *Empirical Software Engineering*, *9*(4), 315–334. https://doi.org/10.1023/B:EMSE.0000039882.39206.5a

Panda, A., Satapathy, S. M., & Rath, S. K. (2015a). Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points. *Procedia Computer Science*, *57*, 772–781. https://doi.org/https://doi.org/10.1016/j.procs.2015.07.474

Panda, A., Satapathy, S. M., & Rath, S. K. (2015b). Neural Network Models for Agile Software Effort Estimation based on Story Points. *Proceedings of the International Conference on Advances in Computing, Control and Networking*, *57*(57), 26–30. https://doi.org/10.15224/978-1-63248-038-5-06

Popli, R., & Chauhan, N. (2013). A sprint-point based estimation technique in Scrum. In M. Mishra (Ed.), *2013 International Conference on Information Systems and Computer Networks* (pp. 98–103). https://doi.org/10.1109/ICISCON.2013.6524182

Popli, R., & Chauhan, N. (2014a). Cost and effort estimation in agile software development. *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, 57–61. https://doi.org/10.1109/ICROIT.2014.6798284

Popli, R., & Chauhan, N. (2014b). Estimation in agile environment using resistance factors. *2014 International Conference on Information Systems and Computer Networks (ISCON)*, 60–65. https://doi.org/10.1109/ICISCON.2014.6965219

Prasada Rao, C., Siva Kumar, P., Rama Sree, S., Devi, J., Rao, C. P., Kumar, P. S., … Devi, J. (2018). An Agile Effort Estimation Based on Story Points Using Machine Learning Techniques. In *Proceedings of the Second International Conference on Computational Intelligence and Informatics* (pp. 209–219). https://doi.org/10.1007/978-981-10-8228-3_20

Silas, F. A., Yusuf, M., & Bijik, A. H. (2017). Hybridization of Class Responsibility Collaborators Model (HCRCM) with Function Point to enhance Project Estimation Cost in Agile Software Development. *Circulation in Computer Science*, *2*(6), 20–24. https://doi.org/10.22632/ccs-2017-252-32

Soares, R. G. F. (2018). Effort Estimation via Text Classification And Autoencoders. *2018 International Joint Conference on Neural Networks (IJCNN)*, 01–08. https://doi.org/10.1109/IJCNN.2018.8489030

Srivastava, B., & Wadhwa, M. (2013). Relative Analysis of Software Cost and Effort Estimation Techniques. *International Journal of Computer Science and Engineering (IJCSE)*, *2*(3), 53–68. Retrieved from http://s3.amazonaws.com/academia.edu.documents/31617658/8._Comp_Sci_-_IJCSE_-Relative_analysis_-Bhawana_Srivastava.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y5

3UL3A&Expires=1489052817&Signature=AZWW5X4YNYTFQ8C4g5nw0lK8PLk%3D&response-content-disposition=inline%3B f

Tanveer, B., Guzman, L., & Engel, U. M. (2017). Effort estimation in agile software development: Case study and improvement framework. *JOURNAL OF SOFTWARE-EVOLUTION AND PROCESS*, *29*(11). https://doi.org/10.1002/smr.1862

Tanveer, B., Guzmán, L., & Engel, U. M. (2016). Understanding and improving effort estimation in Agile software development. *Proceedings of the International Workshop on Software and Systems Process - ICSSP '16*, 41–50. https://doi.org/10.1145/2904354.2904373

Tanveer, B., Vollmer, A. M., & Braun, S. (2018). A hybrid methodology for effort estimation in Agile development: an industrial evaluation. *Proceedings of the 2018 International Conference on Software and System Process*, 21–30. https://doi.org/10.1145/3202710.3203152

Tanveer, B., Vollmer, A. M., & Engel, U. M. (2017). Utilizing Change Impact Analysis for Effort Estimation in Agile Development. *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 430–434. https://doi.org/10.1109/SEAA.2017.64

Trendowicz, A., & Jeffery, R. (2014). *Software Project Effort Estimation: Foundation and Best Practice Guidelines for Success*. https://doi.org/10.1007/978-3-319-03629-8

Yuliansyah, H., Qudsiah, S. N., Zahrotun, L., & Arfiani, I. (2018). Implementation of use case point as software effort estimation in Scrum Framework. *IOP Conference Series: Materials Science and Engineering*, *403*(1), 012085. https://doi.org/10.1088/1757-899X/403/1/012085

Zahraoui, H., & Janati Idrissi, M. A. (2015). Adjusting story points calculation in scrum effort & time estimation. *2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA)*, 1–8. https://doi.org/10.1109/SITA.2015.7358400

Zakrani, A., Najm, A., & Marzak, A. (2018). Support Vector Regression Based on Grid-Search Method for Agile Software Effort Prediction. *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, 1–6. https://doi.org/10.1109/CIST.2018.8596370

Review Article Software Effort Estimation in Agile Methodolgy

ORIGINALITY REPORT

18%	7%	15%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	link.springer.com Internet Source	1%
2	Mirla R. R. Braga, Carla I. M. Bezerra, José Maria S. Monteiro, Rossana M. C. Andrade. "A pattern language for agile software estimation", Proceedings of the 9th Latin-American Conference on Pattern Languages of Programming - SugarLoafPLoP '12, 2012 Publication	1%
3	scitepress.org Internet Source	1%
4	Hind Zahraoui, Mohammed Abdou Janati Idrissi. "Adjusting story points calculation in scrum effort & time estimation", 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), 2015 Publication	1%
5	ceur-ws.org Internet Source	<1%

6	www.jssm.org Internet Source	<1%
7	Sarah Cosentino, Salvatore Sessa, Atsuo Takanishi. "Quantitative Laughter Detection, Measurement, and Classification—A Critical Survey", IEEE Reviews in Biomedical Engineering, 2016 Publication	<1%
8	Submitted to Cranfield University Student Paper	<1%
9	Panut Chongpakdee, Wiwat Vatanawood. "Estimating user story points using document fingerprints", 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2017 Publication	<1%
10	Raufu Oluwatoyin Raheem. "Workflow Development Effort Estimation as Applied to Web Human Resource Management", Global Journal of Flexible Systems Management, 2016 Publication	<1%
11	www.degruyter.com Internet Source	<1%
12	Muhammad Adnan, Muhammad Afzal. "Ontology Based Multiagent Effort Estimation	<1%

System for Scrum Agile Method", IEEE Access, 2017
Publication

13	tel.archives-ouvertes.fr Internet Source	<1 %
14	"Advanced Computing and Intelligent Engineering", Springer Science and Business Media LLC, 2020 Publication	<1 %
15	Abdelali Zakrani, Assia Najm, Abdelaziz Marzak. "Support Vector Regression Based on Grid-Search Method for Agile Software Effort Prediction", 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), 2018 Publication	<1 %
16	Kayhan Moharreri, Alhad Vinayak Sapre, Jayashree Ramanathan, Rajiv Ramnath. "Cost-Effective Supervised Learning Models for Software Effort Estimation in Agile Environments", 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), 2016 Publication	<1 %
17	R. Popli, N. Chauhan. "A sprint-point based estimation technique in Scrum", 2013 International Conference on Information Systems and Computer Networks, 2013	<1 %

Publication

18	Submitted to Universiti Malaysia Pahang Student Paper	<1 %
19	"Agile Processes in Software Engineering and Extreme Programming", Springer Science and Business Media LLC, 2015 Publication	<1 %
20	Submitted to The University of the South Pacific Student Paper	<1 %
21	bmcpulmmed.biomedcentral.com Internet Source	<1 %
22	ethesis.nitrkl.ac.in Internet Source	<1 %
23	www.diva-portal.org Internet Source	<1 %
24	Alhejab Alhazmi, Shihong Huang. "A Decision Support System for Sprint Planning in Scrum Practice", SoutheastCon 2018, 2018 Publication	<1 %
25	Submitted to The University of Manchester Student Paper	<1 %
26	arxiv.org Internet Source	<1 %

27	A Seetharaman, M Senthilvelmurugan, T Subramanian. "Budgeting and accounting of software cost: Part 1", Journal of Digital Asset Management, 2005 Publication	<1 %
28	Submitted to CSU, Fullerton Student Paper	<1 %
29	H Yuliansyah, S N Qudsiah, L Zahrotun, I Arfiani. "Implementation of use case point as software effort estimation in Scrum Framework", IOP Conference Series: Materials Science and Engineering, 2018 Publication	<1 %
30	Janeth Lopez-Martinez, Alan Ramirez-Noriega, Reyes Juarez-Ramirez, Guillermo Licea, Yobani Martinez-Ramirez. "Analysis of Planning Poker Factors between University and Enterprise", 2017 5th International Conference in Software Engineering Research and Innovation (CONISOFT), 2017 Publication	<1 %
31	Jitender Choudhari, Ugrasen Suman. "Phase wise effort estimation for software maintenance", Proceedings of the CUBE International Information Technology Conference on - CUBE '12, 2012 Publication	<1 %

32	Submitted to Turun yliopisto Student Paper	<1 %
33	pdfs.semanticscholar.org Internet Source	<1 %
34	Submitted to University of College Cork Student Paper	<1 %
35	www.mdpi.com Internet Source	<1 %
36	Binish Tanveer, Anna Maria Vollmer, Stefan Braun. "A hybrid methodology for effort estimation in Agile development", Proceedings of the 2018 International Conference on Software and System Process - ICSSP '18, 2018 Publication	<1 %
37	Submitted to Stefan cel Mare University of Suceava Student Paper	<1 %
38	jultika.oulu.fi Internet Source	<1 %
39	"Software Engineering", Springer Science and Business Media LLC, 2019 Publication	<1 %
40	Submitted to Campbellsville University Student Paper	<1 %

41	Diego A. Kietzmann, Maria Paula Iglesia Llanos, Melisa Kohan Martínez. "Astronomical Calibration of the Tithonian – Berriasian in the Neuquén Basin, Argentina: A Contribution From the Southern Hemisphere to the Geologic Time Scale", Elsevier BV, 2018 Publication	<1 %
42	Juana López-Martínez, Edgar Arnoldo Arzola-Sotelo, Manuel Otilio Nevárez-Martínez, F. Javier Álvarez-Tello et al. "Modeling growth on the cannonball jellyfish Stomolophus meleagris based on a multi-model inference approach", Hydrobiologia, 2020 Publication	<1 %
43	Submitted to Universiti Putra Malaysia Student Paper	<1 %
44	eprints.staffs.ac.uk Internet Source	<1 %
45	journals.sagepub.com Internet Source	<1 %
46	www.thinkmind.org Internet Source	<1 %
47	"Innovative Entrepreneurship in Action", Springer Science and Business Media LLC, 2020 Publication	<1 %

48	Azzeh, M.. "Analogy-based software effort estimation using Fuzzy numbers", The Journal of Systems & Software, 201102 Publication	<1 %
49	Submitted to City University Student Paper	<1 %
50	Hussain Azath, Marimuthu Mohanapriya, Somasundaram Rajalakshmi. "Software Effort Estimation Using Modified Fuzzy C Means Clustering and Hybrid ABC-MCS Optimization in Neural Network", Journal of Intelligent Systems, 2018 Publication	<1 %
51	Iman Attarzadeh, Amin Mehranzadeh, Ali Barati. "Proposing an Enhanced Artificial Neural Network Prediction Model to Improve the Accuracy in Software Effort Estimation", 2012 Fourth International Conference on Computational Intelligence, Communication Systems and Networks, 2012 Publication	<1 %
52	Submitted to Jacksonville State University Student Paper	<1 %
53	Jingzhou Li, Guenther Ruhe, Ahmed Al-Emran, Michael M. Richter. "A flexible method for software effort estimation by analogy", Empirical Software Engineering, 2006 Publication	<1 %

54	Submitted to Masinde Muliro University of Science and Technology Student Paper	<1%
55	Melvina Autar Ramessur, Soulakshmee Devi Nagowah. "Chapter 43 Factors Affecting Sprint Effort Estimation", Springer Science and Business Media LLC, 2020 Publication	<1%
56	Muhammad Faisal Abrar, M. Sohail, Sikandar Ali, Muhammad Faran Majeed, Ibrar Ali Shah, Nasir Rashid, Naimat Ullah. "De - motivators for the adoption of agile methodologies for large - scale software development teams: An SLR from management perspective", Journal of Software: Evolution and Process, 2020 Publication	<1%
57	Submitted to Napier University Student Paper	<1%
58	Panagiota Chatzipetrou, Raquel Ouriques, Javier Gonzalez-Huerta. "Approaching the Relative Estimation Concept with Planning Poker", Proceedings of the 7th Computer Science Education Research Conference on ZZZ - CSERC '18, 2018 Publication	<1%
59	Qijun Hu, Zihe Shou, Leping He, Qijie Cai, Mengfei Qu, Yuhao Zhang. "Three-	<1%

dimensional characterization method of pile–rock interface roughness based on fractal geometry", Arabian Journal of Geosciences, 2019
Publication

60	Srdjana Dragicevic, Stipe Celar, Mili Turic. "Bayesian network model for task effort estimation in agile software development", Journal of Systems and Software, 2017 Publication	<1 %
61	Submitted to University of Auckland Student Paper	<1 %
62	Submitted to University of Huddersfield Student Paper	<1 %
63	Submitted to University of Warwick Student Paper	<1 %
64	etd.lib.metu.edu.tr Internet Source	<1 %
65	umpir.ump.edu.my Internet Source	<1 %
66	www.slideshare.net Internet Source	<1 %
67	www.theseus.fi Internet Source	<1 %
68	Janeth López-Martínez, Reyes Juárez-Ramírez, Alan Ramírez-Noriega, Guillermo Licea, Raúl	<1 %

Navarro-Almanza. "Chapter 21 Estimating User Stories' Complexity and Importance in Scrum with Bayesian Networks", Springer Science and Business Media LLC, 2017

Publication

69 N.C. Haugen. "An Empirical Study of Using Planning Poker for User Story Estimation", AGILE 2006 (AGILE'06), 2006 <1 %

Publication

70 Saurabh Bilgaiyan, Samaresh Mishra, Madhabananda Das. "Effort estimation in agile software development using experimental validation of neural network models", International Journal of Information Technology, 2018 <1 %

Publication

71 Submitted to University of Southampton <1 %

Student Paper

72 Submitted to Victoria University <1 %

Student Paper

73 "A Hybrid Effort Estimation Technique for Agile Software Development (HEETAD)", International Journal of Engineering and Advanced Technology, 2019 <1 %

Publication

74 "Computational Intelligence in Pattern Recognition", Springer Science and Business <1 %

Media LLC, 2020
Publication

75 "Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology – ISAT 2016 – Part I", Springer Science and Business Media LLC, 2017 <1%
Publication

76 "Recent Advances in Information Systems and Technologies", Springer Science and Business Media LLC, 2017 <1%
Publication

77 Alan Ramirez-Noriega, Reyes Juarez-Ramirez, Raul Navarro, Janeth Lopez-Martinez. "Using Bayesian Networks to Obtain the Task's Parameters for Schedule Planning in Scrum", 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT), 2016 <1%
Publication

78 Submitted to Higher Education Commission Pakistan <1%
Student Paper

79 Mohd. Owais, R. Ramakishore. "Effort, duration and cost estimation in agile software development", 2016 Ninth International <1%

Conference on Contemporary Computing (IC3), 2016

Publication