

# Methods for Estimating Agile Software Projects: Systematic Literature Review

Edna Dias Canedo      Dandara Pereira Aranha      Maxwell de Oliveira Cardoso  
Ruyther Parente da Costa      Leticia Lopes Leite

University of Brasília (UnB), P.O. Box 4466 – Brasília-DF, CEP 70910-900, Brazil  
ednacanedo, lleite@unb.br, dandaraaranha, maxwell.oliveira2@gmail.com, ruyther@me.com

1

## Abstract

*In recent years, agile methods of software development have gained a lot of attention in the field of software engineering. Several estimation techniques have been proposed by several authors and developers in recent years. This paper performs a Systematic Literature Review aiming to identify the most used metrics e/or methods in the development of agile software and the most used size metrics regarding effort estimates, deadlines and costs in a planning of agile software project. The results suggest that Planning Poker is the most popular technique for agile teams in the planning phase, Story Point and Point of Function are the most used metrics in agile projects for estimating size, time, effort, productivity and cost.*

Keywords: Agile Software Development; Software Estimates; Software Metrics; Story Point; Planning Poker.

## 1. Introduction

The concept of agile development was proposed in 2001 by the Agile Team, and then several software development teams and companies recognized and accepted the concept, so its use gradually increased in software projects [1].

The methods for agile software development are a set of practices that have been created by experienced professionals. These methods can be seen as a reaction to the traditional method of development, which emphasizes a rationalized, engineering-based approach stating that problems are fully specifiable, and that, optimal and predictable solutions exist for any kind of problem [2].

Software metrics are often used to understand, control and improve what is done and how it is done in a software development process. Some of the motivations for using metrics are: project planning and estimation, project management and follow-up, quality understanding and business objectives, communication, processes, and improved tools for software development [3]. Thus, the measurement is applied in the process of software development or attributes of a product with the objective of improving it continuously. This technique used throughout the software development

project assists in estimating, quality control, productivity assessment and project control [4].

When estimating effort, duration, and cost of software development projects, software size is a prerequisite. The functional size of the software to be delivered is a solid basis for estimating a software development project, and one of the most common ways of obtaining the functional size of software is through Function Point Analysis (FPA).

Combining the size of a system, in function points for example, with other metrics, allows the accomplishment of estimates for the development project and definition of a plan of actions focused on meeting the goals [5]. The success of agile software development and estimates still continue to challenge current projects and organizations. Despite the importance of metrics and estimates for software development projects, research related to the theme in the context of agile projects still remains scarce, making estimates and planning inefficient and/or imprecise [6].

The main contribution of this paper is to identify which methods and metrics are considered more adequate for estimating agile software development.

## 2. Planning and Estimating Software Projects

The basis for software metrics was established in the 1970s. The first article on the subject was published in 1968 [7]. From these works, more work and interesting results emerged.

Measurements were mainly created to ensure that indicators could be obtained for optimization of production costs, since in the 1990s billions of dollars were spent on software that did not meet the needs of companies at the time [4].

When calculating metrics, you can refine one of the most important tasks of Project Management, which is planning. According to [4], software measurement makes possible a better understanding of the software engineering process and the product (which it produces) to managers and professionals. Using direct and indirect measures, productivity and quality metrics can be defined. It is also possible to identify the estimated effort, cost and time for a project project.

Estimation is one of the main activities of software planning. They provide data that allows you to predict the time required and the costs of the project. It is not possible to prepare a schedule and budget without the use of estimates.

<sup>1</sup>DOI reference number: 10.18293/SEKE2018-031

Estimates are performed based on metrics. With the application of estimates it is possible to collect metrics that allow to predict the amount of people needed, the time required and the costs for the development of the project. "Thus, it becomes important to invest in the implementation of an estimation process" [8]. Size, duration, productivity, and effort metrics are the most commonly used [9].

## 2.1 Planning and Estimating Agile Software Projects

Agile methodologies propose a large set of techniques for estimating and planning projects, especially in terms of non-algorithm-based models [10], [11]. Most agile estimation techniques focus on the use of User Stories (US). User Stories were first introduced by eXtreme Programming (XP) [12] and then popularized by [10].

There are several techniques for estimating agile software projects, however the ones that will be discussed here are: Planning Poker [10], and Ideal Day [13].

### 2.1.1 Planning Poker

Estimates are limited to specific numbers (each number is written in a card). Each member is holding a deck of Planning Poker cards with values like 0, 1, 2, 3, 5, 8, 13, 20, 40 and 100, which is the sequence that are more popular [14], [15]. However, it is not a standard. Each value represents the number of story points [16] which the team estimates. Story points are the most common unit of measure used for estimating the effort involved in completing a user story or resolving an issue.

Story points refer to customer requirements and describe specific functionalities. When the feature has been fully discussed, each estimator privately selects one card to represent his or her estimate. If they are all the same, the functionality receives the estimate; If not, members proposing the highest and lowest estimates explain their points of view and new rounds are played until a consensus is reached. So the team predicts the speed (the number of points they can deliver in an iteration, through historical data, a test iteration or an educated guess) and down the length of the iteration. The number of iterations is obtained by dividing the total number of points by velocity. Likewise, the project duration is calculated by multiplying the number of iterations by size.

### 2.1.2 Ideal Day

Widely used in Agile Software Projects, Ideal Day corresponds to the amount of work that a professional in the area can complete in a day of work [13], [17].

The speed is calculated from the number of hours the team spends to implement a work equivalent to an Ideal Day [18]. If the item goes through a day of work, it is suggested to decompose this item into smaller items that can be implemented in just one day. According to [18], calculate

the estimated days use the following formula:

$$DE = \frac{IED}{1 - IEDREAL}$$

At where: **DE**: represents the estimated number of days to complete the task; **IED**: represents the time needed to implement the item. This deadline is defined by the team; **IEDREAL**: represents the percentage that indicates the estimate of how much time of day the developer will be dedicated to the implementation of the item.

## 3 Systematic Literature Review

The objective of the systematic review [19] was to identify scientific works that present methodologies and metrics solutions for agile software development methodologies to identify: Common project measurement and control practices; Size metrics used; Research trends in agile development; and Open questions and research topics related to improving the estimates of agile development projects. The research was performed in four steps:

- **Step 1:** Perform automatic search and manual in order to identify a preliminary list of studies. Duplicate studies were discarded. The StArt tool was used as support for the documentation, extraction and structuring of the primary studies.
- **Step 2:** Identification of potentially relevant studies, based on title and abstract analysis, discarding studies that are clearly irrelevant to the research. If there was any doubt about a study regarding its inclusion or exclusion, the next step was to check whether the study was relevant or not.
- **Step 3:** Selected studies in previous steps were reviewed by reading the introduction, methodology section and conclusion and applying the inclusion and exclusion criteria. If reading the above items was not enough to make a firm decision, the study was read in its entirety.
- **Step 4:** thus, a list of primary studies was obtained and subsequently subjected to critical examination using the criteria established.

In order to characterize the methods for estimating agile software projects, two different questions were formulated:

- **RQ.1.** What are the metrics and methods used to make effort estimates, deadlines, and costs for agile software project planning?
- **RQ.2.** Function point metrics can be used to make estimates of effort, deadlines, and costs for agile software planning? If so, is it the most appropriate estimate?

The search string definition was based on the population, intervention, comparison and the result [20]. **Population:** The population is the agile software development. To search the population, the keyword "Agile Software Development" were used. **Intervention:** The intervention is to estimate effort, cost, time and to identify how the Function Points Analysis is performed in agile software projects. Therefore, the terms: metric, prediction, estimation and function points were considered. **Comparison:** The focus of the study was not limited to comparative studies. Therefore, the comparison was not considered in the research strategy. **Result:** The main focus is for the research for estimation metrics and methods that are used in scientific studies and/or industry, as reported by researchers. Thus, the research contained words like empirical, validation, evaluation, etc.

### 3.1 Study Selection Criteria

The following selection criteria were defined for the selection of primary studies:

1. The year of publication of the studies should be between 2007 and 2018. However, classical sources with definitions (books with classical concepts or pioneering articles) were also considered.
2. Works that propose methods or metrics to realize estimations and agile software projects planning.
3. The work has reference to software metrics and function points in agile development.

As criterion of exclusion of the studies was considered the non-fulfillment of some of the inclusion criteria, as well as:

1. Works that do not propose methods and metrics to realize agile software projects estimations.
2. Duplicated works or published as Short Paper.

### 3.2 Results

The Systematic Literature Review (SLR) is a form of secondary study that aims to identify and analyze the relevant research for a given research question [21]. As a result, a total of **291 papers** were found. After applying the inclusion criteria, **27 primary studies** were classified.

In summary, in the first stage, where the search string was inserted in the digital libraries, a total of 291 articles were obtained. In the second stage, the results were filtered by reading their title, abstract, and keywords, and a total of 189 articles were selected. In the fourth stage, another filter was applied, through the complete reading of the article, with the object of finding the answers to the research questions. After this stage, 22 articles had been chosen to answer the research questions.

In addition to the **22 studies** selected in the automatic search, **5** other studies were selected through manual search. After finishing the 3 analysis stages, **27 articles** had been selected for data extraction. Table 1 shows all the selected papers.

The results of the SLR according to each defined research question were: **RQ.1:** The most appropriate estimation model for agile projects is **Story Point** [36]. Thereby, it has been realized that **Planning Poker** is one of the most popular techniques for agile teams in planning and estimating effort before starting each iteration.

The size estimation techniques are grouped together with the effort estimation techniques, because within the context of agile development, effort estimation is often derived from Velocity estimates and calculations [36] and [35]. The Expert Judgment and Use Case Points technique are also frequently used estimation techniques in the context of agile software development. The techniques of estimating effort in the agile context and its occurrence in the selected works: Expert Opinion - 8; Estimate based on model (COCOMO, etc.) - 5; Planning Poker - 11. Use-Case Points - 3; Custom Templates - 2; Number of Lines of Code - 4; Fuzzy based Framework for Estimation - 1.

Through the number of Story Points and Velocity of the development team you can calculate the term of development of a certain functionality [36]. For example, if the total number of Story Points for the desired functionalities is 200 and the Velocity of the team is 20, then it can be concluded that the team will need 20 iterations to complete the development of the respective functionalities. However, the User Story Point (USP) is not objective and can not define a standard practice for estimating the size and complexity of the software [25].

Innovative work has been identified in the area of agile project estimates. The paper [27] for example, proposes a structure that depends on the use of fuzzy logic and aims to help in the production of accurate estimates. In the paper presented by [23] it is proposed to modify the Use-Case-Points (UCP) method to make it suitable for agile software development by naming this new version for interactive UCP version (iUCP).

The work proposed by [38] presents a proposed model of effort prediction caused by changes in software requirements. The model integrates the analysis of the impacts of the changes with the COCOMO effort estimation model to improve the precision of the effort estimates from changes in agile software development projects.

The Bayesian Network model was proposed to help agile project managers estimate project effort. It is a graphical model that describes the probabilistic relations between the related variables [47].

According to the works analyzed, Story Points is the most used size metric to carry out the estimations of agile

Table 1: Selected Papers for Data Extraction

Number	Title	Author
1	Measuring and predicting software productivity: A systematic map and review	[22]
2	iUCP: Estimating Interactive-Software Project Size with Enhanced Use-Case Points	[23]
3	Enhancing Quality in Scrum Software Projects	[24]
4	On the Current Measurement Practices in Agile Software Development	[25]
5	Survey on agile metrics and their inter-relationship with other traditional development metrics	[26]
6	Towards a Fuzzy based Framework for Effort Estimation in Agile Software Development	[27]
7	Function Points, Use Case Points, Story Points: Observations From a Case Study.	[28]
8	Estimating, planning and managing Agile Web projects under a value-based perspective	[29]
9	Effort, duration and cost estimation in agile software development	[30]
10	Cost and effort estimation in agile software development	[31]
11	How is effort estimated in agile software development projects?	[32]
12	Identification of inaccurate effort estimates in agile software development	[33]
13	Effort estimation in Agile software development: A survey on the state of the practice	[34]
14	Effort estimation in Agile Software Development: A systematic literature review	[35]
15	Model-based dynamic cost estimation and tracking method for agile software development	[36]
16	NORPLAN: Non-functional requirements planning for agile processes	[37]
17	Predicting effort for requirement changes during software development	[38]
18	Method for personal capability assessment in agile teams using personal points	[39]
19	Understanding and improving effort estimation in agile software development	[40]
20	Agile metrics for a university software engineering course	[41]
21	Applying Software Metrics with Scrum	[42]
22	On using planning poker for estimating user stories	[15]
23	Efficiency factor and risk factor based user case point test effort estimation model	[43]
24	Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly Iterative or agile software development processes	[44]
25	Does the use of Fibonacci numbers in planning poker affect effort estimates?	[45]
26	Improving the user story Agile technique using the INVEST criteria	[46]
27	Bayesian network model for task effort estimation in agile software development	[47]

projects, however it can also be verified that Function Points are still widely used, often being combined with other metrics.

**RQ.2:** Function point metrics can be used to make estimates. A dynamic cost estimating model for agile software projects can be used, namely, it has the ability to adapt during the development process and with changes in requirements [36]. This model adopts function points such as estimation metric and a tracking algorithm, the project time estimating the size of the functionalities using function points and calculating the cost to derive the duration of the project.

To derive the project plan, three procedures are performed. 1. The project team calculates the function points of the desired functionalities; 2. The estimation model, which is composed of the remaining function points, is generated; 3. The project team develops a plan for the release, iteration, and delivery using estimated cost metrics, such as: people per month and number of lines of code.

Although some papers suggest adaptations of traditional models for estimating and measuring agile software projects, one of the weaknesses of agile communities may still be the failure to estimate and measure projects using standard metrics such as function points that are largely known and used within the industry, but which cover only the functional requirements (Jones 1998), different from Story Points, for example, that do not correspond to a software size and not even the actual effort, but to estimates (not measurement), and that cover the functional and non-functional requirements.

Some selected studies also state that Point of Function is not suitable for estimating agile projects because of their granularity and insufficient support for feedback and requirements change. Therefore, they firmly support the idea that companies that work in a traditional or agile way collect traditional measures of size (such as Function Points) for portfolio management, project management and benchmarking; and that companies working according to an agile method also do this, in addition to collecting size measures in an agile-size metric (such as Story Points) for estimation purposes.

#### 4. Conclusions and Future Work

Through the execution of this work, it was possible to perceive the relevance of size, effort, cost and time estimates in the context of agile software development, and therefore, methods and estimation metrics have been increasingly discussed in scholarly works that seek the best and most precise metrics used in a given agile context.

Through the Systematic Review of Literature (SLR) it was possible to identify the methods and the main size metrics used in estimations in the context of agile software development. Among the most used techniques are: Planning Poker, Expert Opinion and Function Point Analysis. The most used metrics for estimates are Story Points and Func-

tion Points.

The primary studies identified in the SLR showed that the methods and the metrics for estimates are mostly applied to a given context of agile development with adaptations in order to fit the project in question. Thus, it can also be concluded that the estimation metrics must always be adapted to fit the project context, since each project will have its own characteristics which influence the result of the estimates.

The case study showed that the estimates using the Function Points metric had deviation percentages of the actual values from the estimated values lower than the estimates made using the Story Points metric. This is due to the fact that the team has a lot of experience with the use of estimates with the Function Points metric, so the initial values of the productivity estimates and the hour value of 1 FP were much closer to the actual value, which consequently made the rest of the estimates more precise.

However, it is necessary to perform other experiments using real projects of different characteristics to be able to affirm with certainty that the Function Points metric is more accurate than the Story Points metric in agile development projects.

#### References

- [1] J. Li, "Agile software development," 2010.
- [2] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9-10, pp. 833-859, 2008.
- [3] K. Pulford, A. Kuntzmann-Combelles, and S. Shirlaw, *A quantitative approach to software management: the AMI handbook*. Addison-Wesley Longman Publishing Co., Inc., 1995.
- [4] R. S. Pressman, *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2005.
- [5] R. Dias, "Análise por pontos de função: uma técnica para dimensionamento de sistemas de informação," *Revista Eletrônica de Sistemas de Informação ISSN 1677-3071, volume=2, number=2, year=2003*.
- [6] B. Kitchenham, "Whats up with software metrics?—a preliminary mapping study," *Journal of systems and software*, vol. 83, no. 1, pp. 37-51, 2010.
- [7] R. J. Rubey and R. D. Hartwick, "Quantitative measurement of program quality," in *Proceedings of the 1968 23rd ACM national conference*. ACM, 1968, pp. 671-677.
- [8] C. Hazan, "Análise de pontos de função: Uma aplicação nas estimativas de tamanho de projetos de software," *Engenharia de Software Magazine, Edição*, vol. 2, pp. 25-31, 2008.
- [9] P. Bourque, R. E. Fairley *et al.*, *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press, 2014.
- [10] M. Cohn, *Agile estimating and planning*. Pearson Education, 2005.
- [11] J. Highsmith, *Agile project management: creating innovative products*. Pearson Education, 2009.
- [12] K. Beck, *Extreme programming explained: embrace change*. addison-wesley professional, 2000.

- [13] F. Alves and M. Fonseca, "Ideal day e priorizao: Mtodos geis no planejamento," 2008.
- [14] N. C. Haugen, "An empirical study of using planning poker for user story estimation," in *Agile Conference, 2006*. IEEE, 2006, pp. 9–pp.
- [15] V. Mahnič and T. Hovelja, "On using planning poker for estimating user stories," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2086–2095, 2012.
- [16] M. Choetkiertikul, H. K. Dam, T. Tran, T. T. M. Pham, A. Ghose, and T. Menzies, "A deep learning model for estimating story points," *IEEE Transactions on Software Engineering*, 2018.
- [17] M. Usman, "Improving expert estimation of software development effort in agile contexts," Ph.D. dissertation, Blekinge Tekniska Högskola, 2018.
- [18] J. C. C. Martins, "Tcnicas para gerenciamento de projetos de software," 2001.
- [19] B. Kitchenham, "Procedures for performing systematic reviews," *Keele, UK, Keele University*, vol. 33, no. 2004, pp. 1–26, 2004.
- [20] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *ech. rep. EBSE 2007-001*, 2007.
- [21] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.
- [22] K. Petersen, "Measuring and predicting software productivity: A systematic map and review," *Information and Software Technology*, vol. 53, no. 4, pp. 317–343, 2011.
- [23] N. Nunes, L. Constantine, and R. Kazman, "Iucp: Estimating interactive-software project size with enhanced use-case points," *IEEE software*, vol. 28, no. 4, pp. 64–73, 2011.
- [24] A. R. Mukker, A. K. Mishra, and L. Singh, "Enhancing quality in scrum software projects," *International Journal of Science and Research (IJSR)*, vol. 3, no. 4, pp. 682–688, 2014.
- [25] T. Javdani, H. Zulzalil, A. A. A. Ghani, A. B. M. Sultan, and R. M. Parizi, "On the current measurement practices in agile software development," *arXiv preprint arXiv:1301.5964*, 2013.
- [26] S. Misra and M. Omorodion, "Survey on agile metrics and their inter-relationship with other traditional development metrics," *ACM SIGSOFT Software Engineering Notes*, vol. 36, no. 6, pp. 1–3, 2011.
- [27] A. T. Raslan, N. R. Darwish, and H. A. Hefny, "Towards a fuzzy based framework for effort estimation in agile software development," *International Journal of Computer Science and Information Security*, vol. 13, no. 1, p. 37, 2015.
- [28] J. Schofield, A. Armentrout, and R. Trujillo, "Function points, use case points, story points: Observations from a case study," *CrossTalk*, vol. 26, no. 3, pp. 23–27, 2013.
- [29] C. Torrecilla-Salinas, J. Sedeño, M. Escalona, and M. Mejías, "Estimating, planning and managing agile web development projects under a value-based perspective," *Information and Software Technology*, vol. 61, pp. 124–144, 2015.
- [30] M. Owais and R. Ramakishore, "Effort, duration and cost estimation in agile software development," in *Contemporary Computing (IC3), 2016 Ninth International Conference on*. IEEE, 2016, pp. 1–5.
- [31] R. Popli and N. Chauhan, "Cost and effort estimation in agile software development," in *Optimization, Reliability, and Information Technology (ICROIT)*. IEEE, 2014, pp. 57–61.
- [32] T. Schweighofer, A. Kline, L. Pavlic, and M. Hericko, "How is effort estimated in agile software development projects?" in *SQAMIA*, 2016, pp. 73–80.
- [33] F. Raith, I. Richter, R. Lindermeier, and G. Klinker, "Identification of inaccurate effort estimates in agile software development," in *Software Engineering Conference (APSEC), 2013 20th Asia-Pacific*, vol. 2. IEEE, 2013, pp. 67–72.
- [34] M. Usman, E. Mendes, and J. Börstler, "Effort estimation in agile software development: a survey on the state of the practice," in *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*. ACM, 2015, p. 12.
- [35] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in agile software development: a systematic literature review," in *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*. ACM, 2014, pp. 82–91.
- [36] S. Kang, O. Choi, and J. Baik, "Model-based dynamic cost estimation and tracking method for agile software development," in *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on*. IEEE, 2010, pp. 743–748.
- [37] W. M. Farid and F. J. Mitropoulos, "Norplan: Non-functional requirements planning for agile processes," in *Southeastcon, 2013 Proceedings of IEEE*. IEEE, 2013, pp. 1–8.
- [38] S. Basri, N. Kama, F. Haneem, and S. A. Ismail, "Predicting effort for requirement changes during software development," in *Proceedings of the Seventh Symposium on Information and Communication Technology*. ACM, 2016, pp. 380–387.
- [39] S. Čelar, M. Turić, and L. Vicković, "Method for personal capability assessment in agile teams using personal points," in *Telecommunications Forum Telfor (TELFOR)*. IEEE, 2014.
- [40] B. Tanveer, L. Guzmán, and U. M. Engel, "Understanding and improving effort estimation in agile software development: an industrial case study," in *Software and System Processes (ICSSP), 2016 IEEE/ACM International Conference on*. IEEE, 2016, pp. 41–50.
- [41] C. Matthies, T. Kowark, M. Uflacker, and H. Plattner, "Agile metrics for a university software engineering course," in *Frontiers in Education Conference (FIE), 2016 IEEE*. IEEE, 2016, pp. 1–5.
- [42] M. L. Gamba and A. C. G. Barbosa, "Engenharia de software-aplicação de métricas de software com scrum," *Anais SULCOMP*, vol. 5, 2010.
- [43] A. W. M. M. Parvez, "Efficiency factor and risk factor based user case point test effort estimation model compatible with agile software development," in *Information Technology and Electrical Engineering (ICITEE), 2013 International Conference on*. IEEE, 2013.
- [44] H. M. Olague, L. H. Eitzkorn, S. Gholston, and S. Quattlebaum, "Empirical validation of three software metrics suites to predict fault-proneness of object-oriented classes developed using highly iterative or agile software development processes," *IEEE Transactions on Software Engineering*, vol. 33, no. 6, pp. 402–419, 2007.
- [45] R. Tamrakar and M. Jørgensen, "Does the use of fibonacci numbers in planning poker affect effort estimates?" 2012.
- [46] L. Buglione and A. Abran, "Improving the user story agile technique using the invest criteria," in *Software Measurement and the 2013 Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2013 Joint Conference of the 23rd International Workshop on*. IEEE, 2013, pp. 49–53.
- [47] S. Dragicevic, S. Celar, and M. Turic, "Bayesian network model for task effort estimation in agile software development," *Journal of Systems and Software*, vol. 127, pp. 109–119, 2017.