Challenges of Using Software Size in Agile Software Development: A Systematic Literature Review

Tuna Hacaloglu[1-2] and Onur Demirors[3-4]

[1] Atilim University, Ankara, 06830, Turkey
[2] Middle East Technical University, Ankara, 06800, Turkey
tuna.hacaloglu@atilim.edu.tr
[3] Izmir Institute of Technology, Izmir, 35430, Turkey
[4] University of New South Wales, Sydney, NSW 2052, Australia
onurdemirors@iyte.edu.tr

Abstract. Software size is a fundamental measure for software management. Size is used for a variety of purposes, such as benchmarking, normalization, and portfolio measurement, and it is frequently considered as the sole input of estimation. Estimations can be produced for various reasons; e.g., to predict effort, cost and duration of software development projects. There are different types of software size measures. Particularly in projects where agile methodologies are adopted, measurement becomes a significant challenge as it is perceived as a non-value-added task and records of tasks such as requirements identification are not always consistent. The difficulties of applying traditional size measurement techniques in agile contexts, however, do not diminish the need, and new methods and techniques are introduced to improve the manageability of the agile projects. In this paper, we discuss estimation and measurement approaches in relation with "software size" in agile contexts. Based on this review, we present the perceptions of software size and related challenges, such as misinterpretation of size, difficulties in implementation, and acceptability of the measurement processes. We anticipate that providing a baseline for the state of software size measures in agile contexts and presenting related challenges, particularly in terms of its acceptability by practitioners can shed light on the development of new techniques.

Keywords: Agile software development, size, measurement, estimation, function points, story points, use case points, line of code.

1 Introduction

Measurement is important in software engineering in order to accomplish three aims: understanding, controlling and improving the current situation [1]. Especially through size measurement managers are able to manage, maintain and improve projects, and make comparisons across projects [2]. In [3] the importance of software size is explained as when there is no solid baseline of size, neither the estimation and planning; nor control of a large-scale software project can be undertaken objectively. Software size is frequently represented in terms of length, functionality, and complexity attributes [1]. Function point analysis and lines of code (LOC) are the most recognized sizing methods [4].

In practice, despite being a basic concept, size is not uniquely perceived by researchers and practitioners. Size measurement is often confused with estimation. In [5], Ozkan and Demirors clearly differentiated measurement from estimation by stating that whenever the necessary details of the user requirements exist, a measurement of the functional size is possible. However, when there is a lack of details, one can only estimate the size. In other words, estimation is a substitute in situations where measurement is not possible.

In projects utilizing agile methodologies, there is uncertainty in initial requirements, and documentation of requirements is minimal and generally in the form of user stories which can be considered as “light weight requirements” [S1], p.54). For this reason, the availability of such resources and their maturity in terms of the level of details required for functional size measurement (FSM) can prevent the adoption of functional size measurement methods and complicate early estimations. Practitioners, instead, adopt expert-based estimation techniques which are criticized for their subjective nature. Moreover, different interpretations of subjective size measures can lead to a confusion of the size and effort concepts. Previously, this situation was observed by Ozkan and Demirors in [5], who reported that practitioners in the field frequently referred to functional size as an effort measure.

In the literature, there are systematic literature reviews (SLRs) exploring cost estimation using soft computing techniques [6], effort estimations [7], [8], [9] and a survey on cost/size estimation [10] in agile software development. However, none of these studies specifically focus on size measurement or estimation in agile software development and discuss how size is perceived and utilized in this process or address the related challenges.

In this study, by recognizing the importance of size in software engineering, we aimed to determine how size concept is perceived in the agile context by means of an SLR study and by focusing more specifically on the perspectives of size and related challenges. We anticipate that the findings of this paper can contribute to develop an understanding of the problems related to size measurement and estimation in the agile software development literature and can be useful while developing new measurement techniques.

The rest of the paper is structured as follows: section 2 describes the research methodology, section 3 discusses how size is interpreted in agile software development along with the challenges described in the literature, and finally section 4 provides the conclusion and recommendations for future work.

2 Research Methodology

SLRs on size estimation/measurement in the agile software development literature were conducted by utilizing the guidelines given by [11]. We focused on answering the following two research questions:

RQ1.What size measurement/estimation methods are utilized in agile software development?

RQ2.What are the challenges in agile size estimation and measurement?

Five databases were chosen to search for the articles; Scopus[1], IEEE Xplore[2], Web of Science[3], ScienceDirect[4], and ACM digital Library[5]. For this search, the keywords were determined from the domain knowledge of the authors and also by observing the existing SLR in the literature, such as that of [8]. The resulting keyword sets were as follows:

Set 1= {agile OR XP OR "extreme programming" OR Scrum}

Set 2= {size AND {estimat* OR measur* OR metric}}

Set 3= {COSMIC OR IFPUG OR "function point" OR "functional size"}

Set 4= {"story point" OR "use case point" OR "line of code" OR "LOC" OR "object points"}

The search was performed by combining each keyword in set 1 with each keyword in sets 2, 3 and 4 using the AND clause.

After the search, a total of 2,581 articles were obtained. The number of articles retrieved from each database is shown in Table 1.

Table 1. Initial search results.

Database	Number of articles retrieved
Scopus	944
IEEE Xplore	580
ScienceDirect	434
ACM	458
Web of Science	165

After removing the duplicates from the 2,581 articles, the title and abstract of each article were read to eliminate irrelevant sources. During this process, the following inclusion and exclusion criteria were used:

Inclusion Criteria:

- Reporting the findings of a software size-related study in the agile software development context. To assess the improvement in the agile size measurement domain, we only included the papers in which software size was the primary objective or referred to improvements or comparisons by explicitly using software size to estimate effort, cost, etc.
- Written in English,
- Published in a journal or conference / workshop proceedings,
- Full-text available.

Exclusion Criteria:

- Not using software size in agile software development,

[1] https://www.scopus.com/search/form.uri?display=basic
[2] https://ieeexplore.ieee.org/Xplore/home.jsp
[3] http://www.webknowledge.com
[4] https://www.sciencedirect.com/
[5] https://dl.acm.org/

• Written in a language other than English,
• Partially available or inaccessible

A backward (looking in references) and forward (using the citations) snowballing step suggested in [12] is also followed to reach more number of relevant studies. For this aim, we adopted a similar strategy conducted in [13] such as choosing randomly five studies in the pool and applying snowballing process on these articles.

The final pool yielded 40 articles which are read to seek an answer to the research questions defined.

3 Results

3.1 Observed measures

From the analysis of the papers, it was observed that story points were the most frequently referenced measure, and COSMIC Function Points (CFP) was the second. The measures observed in relation to RQ1 and the respective studies are given in Table 2.

Table 2. Articles by size measure.

Main Size Measure Type[6]	Articles[7]
Simplified Function Points (SiFP)[25]	[S2], [S3]
NESMA Function Points [24]	[S4]
Function points [29]	[S5], [S6]
IFPUG Function Points [23]	[S7], [S3]
COSMIC Function Point (CFP) [30]	[S8], [S9], [S10], [S11], [S12], [S13], [S14], [S16], [S17], [S18], [S19]
Story Points (SP)	[S20], [S7], [S21], [S4], [S22], [S23], [S24], [S5], [S25], [S26], [S27], [S28],[S29], [S30], [S31], [S32], [S33], [S37], [S38]
Use Case Points	[S34], [S35], [S36],
Actual times	[S39]
Web Objects [28]	[S40], [S6]
User point	[S1]

In this paper, with respect to RQ2, we focus on the size measurement and estimation related challenges identified in these papers. These findings led us to explore additional articles in the literature, in addition to those retrieved from the SLR to discuss

6 These measure types are not always used as is but enhanced with other attributes; for example, [S36] added two elements, efficiency and risk factor, to the existing use case point estimation method. However, in Table 2, we only included the major measure types.

7 It is possible that the same article corresponds to distinct measures due to using more than one measure type or comparing different measures.

the challenges in a more accurate manner. This discussion is presented in the following section 3.2.

3.2 Challenges related to software size in agile projects

In this section, we present the challenges identified related to software size in agile projects, which were categorized mainly as misinterpretation of size, difficulties in implementation, and acceptability of the measurement process. Each of these challenges is individually presented in the following sections. The challenges depicted in the articles are given in Table 3.

Table 2. Challenges observed

Misinterpretation of size measure	Story point can denote size= [S5, S7, S20, S21, S22, S23, S24, S25, S26, S31, S32, S37], and complexity=[S39]. Size can also refer to number of days [S30].
Difficulties in application	Story points are relative [S4, S5, S6, S20, S28], subjective [S4, S18], given based on team members estimating them [S8, S18, S21], not transferrable outside of the team [S18], given by comparing with simplest [S5, S28] or average [S8] story, assigned as high points for non-functional requirements [S8], under/overestimation related problems [S29], problems related to requirement uncertainty [S29], Difficulty related to FSM application [S4], is for complete projects or part of projects [S3]. Estimations are not model [S18] and mathematical based [S21].
Measurement/ estimation process acceptability	Difficulty related to FSM application [S4], Problems related to adjusting requirements [S12, S15], Agile team members do not want a pressure one them [S39], Story point assignment is costly [S1], tend to take long time [S16], affected by dominant characters [S20], Political pressure [S27].

Interpretations of size measures. In [5] Ozkan & Demirors observed and discussed the misconception that practitioners in the field frequently refer to functional size as an effort measure. This is also observed in agile software development, not limited to size and effort but also seen in size and duration. This confusion between size and effort is also mentioned in the guidelines for functional size measurement in agile projects published by COSMIC [14].

This misinterpretation of size and effort results from the fact that what a "story point" denotes is often ambiguous in the agile software development literature. In our SLR, we observed that a story point was presented as a size estimate in some articles but as an effort or duration estimate in others. The representative studies for example [S20] and [S7] generally referred to Cohn [15], who related a story point with size, and described it as "a pure measure of size" (p.71). In [S5] Kang, Choi and Baik also claimed that "In an agile project, the story points are measured to estimate the size of the work for each user story" (p.744). Similarly, in [S37] Bhalerao and Ingle suggested a size-based estimation of stories using story points. Furthermore, in [S31] story point is also considered as a size measure.

On the other hand, authors in Satapathy, Panda and Rath [16] interpreted a story point as an "effort measure" by stating that "In the case of agile projects, story point is used to measure the effort required to implement a user story."(p.304).

In [S24], by referring to Cohn [15], Miranda, Bourque and Abran drew attention to the fact that there is a proportion between the story point and the efforts required to implement them, and exemplified this situation as "A 6-point user story is expected to require about twice as much effort as a 3-point user story" (p.1327). These definitions were encountered in other studies, but here only some have been discussed to present the two main views of story points; i.e., size and effort.

Additionally, size was expressed in terms of duration by Cohn in [15], who proposed to measure the size of software based on ideal days by stating that it is possible to consider ideal days as size estimate like story points when the organizational overhead is not taken into account and further stating that "Then, an estimate of size expressed as a number of ideal days can be converted into an estimate of duration using velocity in exactly the same way as with story points." [15] (p.46). In addition, in [S30] Popli and Chauhan related story points with duration as follows: "The size of user story is defined in terms of number of days i.e. time needed to complete the user story but this time estimation contains uncertainty" (p. 1358).

In another study [S39], Hohman defined story points as "an arbitrary, indivisible unit of story 'complexity', intended to map linearly to the actual, calendar time necessary for completion." By 'map linearly' I mean that two story points are supposed to take twice as much calendar time to complete as one story point." ("Story points, para. 1).

In [17] it is reported that "the team defines the relationship between story points and effort. Usually 1 story point is equal to 1 ideal working day." (p. 773). In [18] Goswami, Jasuja and Dhir pointed to the drawback of this view by referring to [19] who stated that estimating a user story with ideal time concept can be perplexing for developers which may lead them to assume that there is a relation between story points and time that is not valid because story point corresponds to a distribution of time.

It should also be noted that story points cannot be considered as an objective measure [S4], [S8] because they are relative [S4]; [S5]; [S20], subjective [S4, S18], and unreliable [14], and they have a meaning limited to the project and the team they belong to [S8]. They are assigned by using the experience of measurers, with the help analogies with previous cases or user stories the measurers have at hand at the time of estimation. Therefore, the way they are assigned to stories is not compliant with the conventional measurement procedure, described by McDermid in [20] as "…measurement is the means of recording this property with integrity: objectively, reliably, repeatably, efficiently, and without bias or ambiguity." (p.12/3).

The reviewed articles revealed that neither the exact meaning of a story point is clear nor its relationship with effort and duration. Since the story point concept is already troublesome because it is relative and subjective, conflicting interpretations make the issue even more problematic. The varying meaning of size, which is sometimes confused with effort and duration, is emerging from the lack of a stable size measurement method being adopted by the agile community.

There is a clear need for well defined, concrete, objective size measurement methods applicable in agile software projects. The size obtained through such methods can be used as an input to estimate the effort, duration, cost and scope by appropriate techniques. In addition, objective size measures can establish a baseline for benchmarking, portfolio management, and normalization of indirect measures, including defect density.

Difficulties in application. In this part, we present the difficulties reported in the literature related to the application of software size measurement and estimation in agile software development. In [21] Javdani et al. emphasized that there was no commonly accepted, standardized practice for agile software measurement. Although story points are a highly adopted estimate in the agile literature, it does not fulfill the estimation needs. In addition to the confusion and misinterpretations concerning the story point concept as represented in Section 3.1, there are also problems with its usage.

Story point has been frequently criticized for its relativity [S5] and subjectivity [S4]. In [S21], Hamouda commented on the varying nature of story points across the teams for similar features and emphasized that this situation can create a challenge for organizations in maintaining consistency across projects. Adoption of FSM methods is required especially for benchmarking purposes [S8]. Popli and Chauhan in their study [S27] pointed out that since the estimation methods used in agile projects are not based on mathematical formulas for effort and cost calculation, they are not effective. To overcome this, some researchers; e.g., Khatri, Malhotra and Johri [S34] integrated technical and environmental complexity factors into use case points. However, the complexity perception is also a relative concept.

Commeyne, Abran and Djouab [22] criticized subjective estimations in agile projects being based on a consensus, and thus varying from one team to another even for the same story set, depending on estimators' background in terms of abilities and past experiences, rather than the sizing of the product. In [22], the authors also emphasize the inability of re-using past estimates for iterations, assessing the quality of estimates, and taking related actions for improvement.

In Kang, Choi and Baik [S5] described the following problem that the team faced when assigning story points to stories-: The agile team selects the simplest user story to assign a point to it, and then only they can compare other user stories with it and assign story points. The same authors drew attention to the need for story points to be reassigned when the referenced user story is changed. Another similar way of assigning story points is presented in [14], where the team first selected an average user story, and then assigned story points to other user stories, comparing them with the average. Here, it can be inferred that software size estimation is restricted to the perception of estimators at the moment of estimation and to the scope of the user stories at hand. Moreover, when the story used as a reference is changed, the estimate should be updated, which leads to overheads and extra effort.

As a response to the problems related to subjective estimation techniques, functional size measurement methods are being adopted by the agile community. COSMIC [14] recently published a guideline on the usage of the COSMIC functional size measurement (COSMIC FSM) method for agile user stories. This was followed by attempts to apply COSMIC FSM method to agile user stories [S9] [S10] [S11] and observe its usability [S13], [S16], [22] in Scrum projects.

In [S7], Santana et al. detected a positive correlation between functional size in terms of IFPUG function points [23] and the number of story points. This study was replicated by Huijgens and Solingen [S4], who assessed the relationship between NESMA [24] for functional size measurement and using story points, and found a negative correlation. On the other hand, Lenarduzzi and Taibi in [S2] conducted research using Simplified Function Points (SiFP) [25] and IFPUG Function Points [23] in Scrum projects and concluded that these methods did not have any contribution to estimation accuracy. Since there has been only a handful studies on the topic to date, it is too early to draw a conclusion on the usability of functional size measurement in agile contexts. However, there are clear difficulties in relation to maintaining a balance between the need to create well defined requirements for measuring functional size and minimizing overhead arising from welcome the changing requirements in agile projects. Further case studies are required to explore the usability and acceptability of functional size measures in agile projects. Furthermore, new measurement approaches or new extensions of current approaches that might be more suitable for measuring size of user stories should be explored

Measurement/estimation process acceptability. As mentioned in Section 3.2, there are studies on the successful application of the functional size measurement methods in agile contexts. However, these methods are not widely used in practice by agile teams. In a survey [S15] by Huijgens et al., it was reported based that the majority of the participants acknowledged the importance of functional size measurement (FSM) in decision making, and a minority of the participants thought that in agile "developers do not like disturbance for functional size measurement" (p.60). The ratio of this minority group was low but still noteworthy to depict the need that adopting functional size measures in the agile world should be considered as a social challenge. This finding was implicitly supported in [8], where it is indicated that subjective estimation techniques were the most used. In addition, as stated in [S8] agile teams resist replacing their already adopted sizing or estimation methods. Furthermore, in [S15] it is found that the participants who were against the usage of FSM complained about the incompatibility of the FSM techniques with the poor, open scope and changing requirements, and short-cycle characteristics of agile software development.

Additionally, in [S39] Hohman stated that instead of being told what to do, developers appreciate taking over the responsibility about the share of their existing work. In the line of this obtained responsibility, developers benefit the authority to foresee the duration of the implementation rather than being told how long it is supposed to be and claimed that the estimates were more realistic when performed through comparisons.

Hussain, Kosseim, and Ormandjieva in [S12] emphasized that agile processes result in time efficiency by eliminating the necessity to adjust requirements to make them formalized and decomposed in a visible granularity, a task that is required by COSMIC method and offered an approach to use COSMIC where the formalism in requirements is not needed. The authors also supported the findings of [S15] emphasizing that a FSM technique COSMIC is considered as time-consuming for agile.

In addition to the FSM techniques, it is pointed out in [S20] that although planning poker, an expert estimation technique, is a prevalent technique to size user stories, it suffered from being time consuming and the probability of becoming involved in unnecessary discussions. Salmanoglu, Hacaloglu and Demirors in [S16] also noted that story points-based estimation took far more effort than COSMIC-based measurement for the cases they studied.

In [26] Mahnič & Hovelja stated that in contrast to the studies in psychology claiming that group processes are risky for effort and schedule estimates, agile methodologies recommend the adoption of planning poker for user story size estimation and developing release and iteration plans. In [S1] Ali, Shaikh, and Ali criticized the group activity such as incorporating all team members to the estimation process, being a potentially costly activity.

In [S20], Power drew attention to another social point: Challenge of planning poker caused by dominant characters when sizing user stories. With this aim, in their study, the authors tried to reduce unnecessary long discussions and save time with the help of silent grouping. As a result, they found that the participants acknowledged the effect of silent grouping positive as they were able to think and express their opinions freely. In [S27] Popli and Chauhan explored the causes for inaccurate estimates in agile development and addressed from the political pressure which may prevent a team to express their real estimate, and rather lead them to make an estimate that would satisfy the manager or customer.

The SLR undertaken in this study revealed that agile projects suffer from the relative and subjective estimation techniques, such as story points, and various studies have drawn attention to the benefits and problems of performing software size estimation as a group activity. It is also noteworthy that developers generally resist adopting a formal functional size measurement technique, such as COSMIC since it requires a pre-study and preparation of requirements before measurement.

4 Conclusion

Considering the popularity of agile methodologies among the practitioners, measurement, estimation, and implicitly the size concept has come into special prominence. However, what software size is and how it should be measured in agile software development projects is not clear. It seems that the most frequently referenced story point concept does not fulfill the need of a size measure. It is treated fundamentally differently in various studies; sometimes referring to effort and time while other times to size based on the viewpoint of the authors. There is no doubt that a well-defined and accepted scope is a prerequisite for a concept to be measurable. If we do not know what we are measuring, it is not likely that the discussion of the results will be fruitful.

Relativity and subjectivity of estimation techniques, such as story points is the main criticism expressed by the researchers. The measurement and estimation process of these measures not only involve software artifacts but also human measurers, as well as application teams as the parameters of the process. The resultant estimations are project-specific and can be used for effort prediction at best. Teams need to utilize other measures for other common usages of size; e.g., benchmarking, portfolio management, and normalization of other outputs like the number of defects. In addition, the efficiency of the estimation process as a group activity is highly questionably, and thus needs further research for improvement.

It is observed from the reported studies that functional size measurement methods have the potential to resolve some of these issues. However, practitioners appear to be reluctant to adopt these methods probably due to three main reasons: First, they find the application-related procedures cumbersome to follow; second, they think there is a mismatch between agility and the need for detailed requirements; and finally, teams prefer to be somewhat free to make decisions about software development.

In conclusion, the literature survey demonstrated the significant need for new size measurement techniques or modifications of the existing method that are applicable to agile contexts. Such new technique should consider all related components, including the artifacts to be measured, the measurement process to be applied, and the challenges discussed in this paper. It should also first start with a clear definition of the concepts on which it is built and resolve the confusion about the concepts in the literature. In future work, we plan to extend this study by performing case studies to provide a deeper understanding of the usability of the common size measurement techniques in practice. We believe that the development of a new size measure will complement our umbrella work on measurement in agile contexts [27].

This study has certain limitations. For example, the search was performed in a limited number of databases, excluding grey literature and books and theses, and the synonyms of the search keywords were not used. In our future work, we hope to extend the literature review considering these limitations and also do a deep analysis of the selected papers in this review.

Acknowledgements

We would like to thank to Asst. Prof. Dr. Bilge SAY for her valuable contributions in reviewing the article and for her precious comments which improved the article.

References

1. Fenton N. E., Pfleeger, S., L.: Software Metrics A Rigorous and Practical Approach. PWS Publishing Company, (1997).
2. Abran, A., Symons, C., and Ebert, C., Vogelezang, F., Soubra, H. Measurement of Software Size: Contributions of COSMIC to Estimation Improvements, (2016).
3. Gencel, C., Demirors, O.: Functional size measurement revisited. ACM Transactions on Software Engineering and Methodology (TOSEM) 17(3), 15 (2008).
4. Trendowicz, A. Jeffery, R. Appendix A: Measuring software size. Software Project Effort Estimation. 433-449, (2014).
5. Özkan, B., Demirors, O.: On the Seven Misconceptions about Functional Size Measurement. In: 2016 Joint Conference of the International Workshop on Software Measurement

and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 45-52. Berlin (2016).
6. Bilgaiyan, S., Mishra, S., Das, M.: A review of software cost estimation in agile software development using soft computing techniques. In: Computational Intelligence and Networks (CINE), 2nd International Conference on. pp. 112-117, IEEE (2016).
7. Nguyen-Cong, D., Tran-Cao, D.: A review of effort estimation studies in agile, iterative and incremental software development. In: Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2013 IEEE RIVF International Conference on, pp. 27-30. IEEE, (2013, November).
8. Usman, M., Mendes, E., Weidt, F., Britto, R.: Effort estimation in agile software development: A Systematic Literature Review. In: Proceedings of the 10th International Conference on Predictive Models in Software Engineering - PROMISE '14, pp. 82–91. (2014).
9. Schweighofer, T., Kline, A., Pavlic, L., Hericko, M.: How is Effort Estimated in Agile Software Development Projects?. In: SQAMIA, pp. 73-80. (2016).
10. Osman, H. H., Musa, M. E.: A Survey of Agile Software Estimation Methods. International Journal of Computer Science and Telecommunications 7(3), 38-42 (2016).
11. Kitchenham B., Charters S. Guidelines for performing systematic literature reviews in software engineering (version 2.3). Technical report, Keele University and University of Durham, (2007).
12. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th international conference on evaluation and assessment in software engineering, pp. 38. ACM, (2014).
13. Garousi, V., Petersen, K., Ozkan, B.: Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. Information and Software Technology 79, 106-127 (2016).
14. COSMIC, The Guideline for the use of COSMIC FSM to manage Agile Projects (2011).
15. Cohn, M.: Agile estimating and planning. Pearson Education, (2005).
16. Satapathy, S. M., Panda, Rath, S. K.: Story point approach based agile software effort estimation using various svr kernel methods (2014).
17. Panda, A., Satapathy, S. M., Rath, S. K.: Empirical Validation of Neural Network Models for Agile Software Effort Estimation based on Story Points. Procedia Computer Science 57, 772-781 (2015).
18. Goswami, R., Jasuja, M., Dhir, S.: Impact of Different Estimation Approaches in Traditional and Agile Development. In: Computational Intelligence & Communication Technology (CICT), 2016 Second International Conference on, pp. 679-682. IEEE, (2016).
19. Simon Baker, Ideal time vs. story points, http://www.energizedwork.com/weblog/2006/02/ideal-time-vs-storypoints
20. McDermid, J. A.: (Ed.), Software engineer's reference book. Elsevier (2013).
21. Javdani, T.Zulzalil, H., Ghani, A.A.A., Md Sultan, A.B. Parizi, R. M.: On the Current Measurement Practices in Agile Software Development. CoRR abs/1301.5964 (2013).
22. Commeyne, C., Abran, A., Djouab, R.: Effort Estimation with Story Points and COSMIC Function Points - An Industry Case Study. Software Measurement News 21(1), 25–36 (2016).
23. IFPUG, IFPUG FSM Method: ISO/IEC 20926 - Software and systems engineering – Software measurement – IFPUG functional size measurement method, New York: International Function Point User Group (IFPUG), (2009).
24. NESMA, NESMA functional size measurement method conform ISO/IEC 24570, version 2.2, Netherlands Software Measurement User Association (NESMA), (2004).

25. Lavazza L., Meli, R.: An Evaluation of Simple Function Point as a Replacement of IFPUG Function Point, IWSM - Mensura 2014, Rotterdam, October (2014).
26. Mahnič, V., & Hovelja, T.: On using planning poker for estimating user stories. Journal of Systems and Software, 85(9), 2086-2095, (2012).
27. Salmanoğlu, M., Coşkunçay, A., Yildiz, A., Demirörs, O.: An Exploratory Case Study for Assessing the Measurement Capability of an Agile Organization. Software Quality Professional 20 (2), 36-47 (2018).
28. Reifer, D. J.: Web development: Estimating quick-to-market software. IEEE Software, 17(6). 57-64. (2000).
29. Albrecht, A.J., Gaffney, J.E.: Software function, source lines of codes, and development effort prediction: a software science validation, IEEE Trans Software Eng. SE-9, 639-648 (1983).
30. The COSMIC Functional Size Measurement Method, Version 4.0.1, Measurement Manual, The COSMIC Implementation Guide for ISO/IEC 19761: 2011, COSMIC Group, April (2015).

Primary Studies

[S1] Ali, M., Shaikh, Z., Ali, E.: Estimation of Project Size Using User Stories. In: The International Conference on Recent Advances in Computer Systems, pp.54-60. (2015).
[S2] Lenarduzzi, V. and Taibi, D. : Can Functional Size Measure Improve Effort Estimation in SCRUM?. In: ICSEA - International Conference on Software Engineering and Advances, pp.173-178. Nice, France (2014).
[S3] Lenarduzzi, V., Lunesu, I., Matta, M.,Taibi, D.: Functional Size Measures and Effort Estimation in Agile Development: A Replicated Study. In: International Conference on Agile Software Development, pp. 105-116. Springer International Publishing, (2015-may).
[S4] Huijgens, H., Solingen, R. V.: A replicated study on correlating agile team velocity measured in function and story points. In: Proceedings of the 5th International Workshop on Emerging Trends in Software Metrics, pp. 30-36. ACM, (2014).
[S5] Kang, S., Choi, O., Baik, J.: Model-Based Dynamic Cost Estimation and Tracking Method for Agile Software Development. In: 2010 IEEE/ACIS 9th International Conference on Computer and Information Science, pp.743-748. Yamagata (2010).
[S6] Soni, D., Kohli, P. J.: Cost estimation model for web applications using agile software development methodology. Pertanika Journal Of Science & Technology, 25(3). 931-938. (2017).
[S7] Santana, C., Leoneo, F., Vasconcelos, A., Gusmão, C.: Using function points in agile projects. In: International Conference on Agile Software Development, pp. 176-191. Springer, Berlin Heidelberg (2011).
[S8] Buglione, L.,Trudel, S.: Guideline for sizing agile projects with COSMIC. In: Proceedings of the IWSM/MetriKon/Mensura, Stuttgart, Germany, (2010).
[S9] Desharnais, J. M., Kocaturk, B., Abran, A.: Using the cosmic method to evaluate the quality of the documentation of agile user stories. In: Software Measurement, 2011 Joint Conference of the 21st Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA), pp. 269-272. IEEE, (2011).
[S10] Desharnais, J. M., Buglione, L., & Kocatürk, B.: Using the COSMIC method to estimate Agile user stories. In: Proceedings of the 12th International Conference on Product Focused Software Development and Process Improvement, pp. 68-73. ACM, (2011).

[S11] Desharnais, J., Buglione, L., & Kocaturk, B.: Improving Agile Software Projects Planning Using the COSMIC Method. In: workshop on Managing Client Value Creation Process in Agile Projects, pp.1-11. Torre Cane, Italy (2011).
[S12] Hussain, I., Kosseim, L., Ormandjieva, O.: Approximation of COSMIC functional size to support early effort estimation in Agile. Data & Knowledge Engineering 85, 2-14 (2013).
[S13] Ungan, E., Çizmeli, N. and Demirörs, O.: Comparison of Functional Size Based Estimation and Story Points, Based on Effort Estimation Effectiveness in SCRUM Projects. In 40th Euromicro Conference on Software Engineering and Advanced Applications Comparison, pp. 77-80. (2014).
[S14] Ochodek, M.: Approximation of COSMIC Functional Size of Scenario-Based Requirements in Agile Based on Syntactic Linguistic Features—A Replication Study. In: Software Measurement and the International Conference on Software Process and Product Measurement (IWSM-MENSURA), 2016 Joint Conference of the International Workshop on, pp. 201-211. IEEE, (2016).
[S15] Huijgens, H., Bruntink, M., van Deursen, A., van der Storm, T., and Vogelezang. F.: An exploratory study on functional size measurement based on code. In: Proceedings of the International Conference on Software and Systems Process (ICSSP '16), pp.56-65. ACM, New York, NY, USA, (2016).
[S16] Salmanoglu, M., Hacaloglu, T., and Demirors. O.: Effort estimation for agile software development: comparative case studies using COSMIC functional size measurement and story points. In: Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement (IWSM Mensura '17), pp.41-49. ACM, New York, NY, USA (2017).
[S17] Dumas-Monette J, Trudel, S.: Requirements Engineering Quality Revealed through Functional Size Measurement: An Empirical Study in an Agile Context. In: 2014 Joint Conference of the International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement, pp.222-232. Rotterdam, (2014).
[S18] Berardi, E., Santillo, L.: COSMIC-based project management in agile software development and Mapping onto related CMMI-DEV process areas. In: International Workshop on Software Measurement-IWSM, (2010).
[S19] Fehlmann, T., Santillo, L.: From story points to cosmic function points in agile software development–a six sigma perspective. In: Metrikon-Software Metrik Kongress, pp. 24. (2010).
[S20] Power, K.: Using silent grouping to size user stories. In: International Conference on Agile Software Development, pp. 60-72. Springer, Berlin Heidelberg (2011).
[S21] Hamouda, A. E. D.: Using agile story points as an estimation technique in cmmi organizations. In: Agile Conference (AGILE), pp. 16-23. IEEE, (2014).
[S22] van Valkenhoef, G., Tervonen, T., de Brock, B., Postmus, D.: Quantitative release planning in extreme programming. Information and software technology 53(11), 1227-1235 (2011).
[S23] Bhalerao, S., Ingle, M.: Incorporating Vital Factors in Agile Estimation through Algorithmic Method. IJCSA 6 (1), 85-97 (2009).
[S24] Miranda, E., Bourque, P., Abran, A.: Sizing user stories using paired comparisons. Information and Software Technology 51(9), 1327-1337 (2009).
[S25] Kumar, C. S., Kumari, A. A., Perumal, R. S.: An Optimized Agile Estimation Plan Using Harmony Search Algorithm. International Journal of Engineering and Technology (IJET) 6(5), 1994-2001 (2014).

[S26] Aslam, W. A. Q. A. R., Ijaz, F. A. R. A. H., Lali, M. I., Mehmood, W. A. Q. A. R.: Risk aware and quality enriched effort estimation for mobile applications in distributed agile software development. Journal of Information Science and Engineering, 33(6). 1481-1500. (2017).

[S27] Popli, R., Chauhan, N.: Cost and effort estimation in agile software development. In: Optimization, Reliabilty, and Information Technology (ICROIT), 2014 International Conference on, pp. 57-61, IEEE (2014).

[S28] Popli, R., Chauhan, N.: Agile estimation using people and project related factors. In: Computing for Sustainable Global Development (INDIACom), 2014 International Conference on, pp. 564-569. IEEE, (2014).

[S29] Popli, R., Chauhan, N.: Estimation in agile environment using resistance factors. In: Information Systems and Computer Networks (ISCON), 2014 International Conference on, pp. 60-65. IEEE, (2014).

[S30] Popli, R., Chauahn, N.:Managing uncertainty of story-points in Agile software. In: Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on, pp. 1357-1361. IEEE, (2015).

[S31] Choudhari, J., Suman, U.: Story points based effort estimation model for software maintenance. Procedia Technology, 4, 761-765 (2012).

[S32] Choudhari, J., Suman, U.: Phase wise effort estimation for software maintenance: an extended SMEEM model. In: Proceedings of the CUBE International Information Technology Conference, pp. 397-402. ACM, (2012).

[S33] Zahraoui, H., Idrissi, M. A. J.: Adjusting story points calculation in scrum effort & time estimation. In: Intelligent Systems: Theories and Applications (SITA), 2015 10th International Conference on, pp. 1-8. IEEE, (2015).

[S34] Khatri, S. K., Malhotra, S., and Johri, P.: Use case point estimation technique in software development. In: Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), 2016 5th International Conference on, pp. 123-128. IEEE (2016).

[S35] Nunes, N. J.: iUCP–estimating interaction design projects with enhanced use case points. In: International Workshop on Task Models and Diagrams for User Interface Design, pp. 131-145. Springer, Berlin Heidelberg (2009).

[S36] Parvez, A. W. M. M.: 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, pp. 113-118. IEEE, (2013).

[S37] Bhalerao, S., Ingle, M.: Agile estimation using caea: A comparative study of agile projects. In: Proceeding of International Conference on Computer Engineering and Application (ICCEA09), pp. 78-84. Philippines, (2011).

[S38] Satapathy, S. M., Rath, S. K.: Empirical assessment of machine learning models for agile software development effort estimation using story points. Innovations in Systems and Software Engineering 13(2-3), 191-200. (2017).

[S39] Hohman, M. M.: Estimating in actual time [extreme programming]. In: Agile Conference, pp.132-138. IEEE, 2005.

[S40] Celar, S., Seremet, Z., Marusic, Z., Turic, M.: Using of Web Objects method in agile web software projects. In: Telecommunications Forum (TELFOR), 2013 21st, pp. 873-876. IEEE (2013).