

User Story Characteristics Affecting Software Cost in Agile Software Development: A Systematic Literature Review

Muhammad Ijaz Khan, Zia Ud Din, Muhammad Ali Abid, Tariq Naeem

Institute of Computing and Information Technology, Gomal University, Pakistan.

Abstract

Agile methodology was started in 2001 and since then, lot of companies have shifted towards Agile Software Development (ASD). Many researchers did a lot of work on Software Project Estimation in ASD. Many estimation techniques have been introduced to estimate Software cost. But no could introduce a mechanism that could be called conclusive. The recent CHAOS summary reports show 75% software projects still failed due to inaccurate estimation. In ASD requirements are collected in the form of user stories. Effort estimation techniques are mainly based on user stories and most of these techniques ignore user story characteristics. The current study aims to explore the characteristics of user stories that can affect effort estimation in ASD which is helpful to improve the efficiency of current estimation techniques.

1. Introduction

The software estimation is the process of forecasting the size of the software product, required development efforts, project schedules, and approximating overall cost of the project. It is the most critical and challenging task to accurately estimate the cost in the project management. For successful software development the required resources and schedules are needed to be accurately estimated [1][2]. It is an admitted fact that nearly 3 out of 4 projects overrun their budget or time or both as CHAOS summary reports continuously described decrease in success rate of projects since 2015 [3]. It is the most critical and complex issue in software development to predict the development cost, time and efforts accurately to make good management decisions required for both project managers, system analyst and developers otherwise it will lead to complete fiasco. It is believed that huge overrun occurs only due to inaccurate estimation.

The overall cost estimation process of software project management is not different from estimating the cost of any other engineering discipline but there are some aspects that are peculiar to software estimation due to the nature of software and software estimating methodologies. Every estimation method has different parameters for predicting the cost of software because the software is invisible, intangible and intractable which makes it more difficult to understand and forecast the cost of software. Furthermore, every software is somewhat different than any other software which leads to a different characteristic set.

2. Literature Review

Sungjoo Kang et al. [6] made an estimation model for agile projects. It is based on function points (FP). FP are generally utilized to estimate the expense and exertion that are required to build up a product. This methodology is generally utilized in conventional methodology. In Agile process, most broadly acknowledged estimation strategy depends on story points. They have fused the FP approach what's more toward story Point to accomplish the most abnormal amount of precision. The project position is progressively followed with the assistance of Kalman filter algorithm. The validation is achieved with the assistance of contextual investigation by contrasting the outcomes and the conventional methodology.

I. Hussain et al. [5] built up a strategy to evaluate the functional size of COSMIC standard. Where COSMIC is an ISO standard used to calculate the software functional size dependent on client needs. Be that as it may, this methodology isn't reasonable for agile process because it requires the client necessities to be formalized and deteriorated. This research addresses the issues by estimating the COSMIC functional size from in-formal literary necessities that outfits with the agile procedure.

The most effective assessment technique for ASD is Use case estimation (UCP). Parvez [7] built up another layer in the current UCP assessment technique. In this technique they have presented two contributing elements specifically: productivity and hazard for assessing the exertion required for testing. The current UCP strategy considers just the properties of the project yet this research centers around the team properties aside the project. The imperative factors to be focused in the new layer are resources of test team, span, testing weightage, Proficiency factor and hazard factors. The presentation of new layer in the current UCP enhances the adequacy and performance of the assessment.

T. Salinas et al [4] proposed a system for assessment and planning of online activities reasonable for Scrum based projects. This methodology depends on value-based point of view by consolidating various existing agile strategies. The proposed system is approved by real-life contextual analyses with the end goal to acquire the precise conclusion. This methodology is exceedingly appropriate for planning, managing, and evaluating web based agile projects.

S. Garg et al. [8] developed estimation model for ASD. S. Garg recognized highly correlated attributes. He presented Principle Component Analysis (PCA) for lessening the quantity of extensive attributes. This methodology is appropriate even without statistical information and expert opinion. The outcome from this methodology demonstrates to have a superior exactness and precision of cost assessment in ASD projects.

Story Point Approach (SPA) is the most generally utilized methodology in ASD assessment. A. Panda et al. [9] enhances the estimation exactness in Agile dependent on neural networks. This methodology considers distinctive sorts of neural systems like General Regression neural systems (GRNN), polynomial and probabilistic neural networks to enhance the exactness of the effort estimation. This strategy is good for effort assessment, anyway it ignores cost, schedule and risk.

K. Moharrer et al. [10] gave the idea of automatic assessment technique called "Auto Estimate" for evaluating exertion for ASD. This methodology is supplementing to broadly utilized manual planning poker procedure. The best learning strategy is chosen automatically by carrying 1) Data accumulation by utilizing story cards, textual investigation, building the model with extracted features and performs analysis by estimating the performance. This model likewise furnishes promising outcomes regarding exactness.

3. Systematic Literature Review

A systematic literature review is conducted methodically by following a set of guidelines to collect and analyze all available evidence about a specific question in an unbiased and repeatable manner [16]. Following steps will be performed during this SLR.

3.1. Research Question:

What are the basic characteristics of user story that can affect effort estimation in agile software development?

3.2. Search Strategy:

In search strategy the electronic databases and manual conferences proceedings are searched. A search strategy starts with the identification of major key terms from PICOC and their alternatives and synonyms. These terms are used to form a query string that is used to derive the rest of the search process.

3.3. Query String

It is iterative process to form a string for searching. At first, I pursued the SLR rules [16] to make a basic string utilizing Boolean OR/AND. All synonyms of the terms and their alternatives are used with "OR" and then ANDed to create searching string. I applied the basic query string on search engine to get the pertinent studies. The basic search query was applied to well known search engines like IEEE explore, Scopus, Science Direct and google scholar. Catchphrases from known essential examinations and recently gotten ones were included in the string. Here, additionally examined the titles, summaries and author catchphrases from some known basic studies to distinguish seek terms.

3.4. Search terms:

The keywords used for search query are listed in Table 1.

Table 1: Keywords Extracted from Literature

SNo.	Keywords	References
1	Agile cost	[20,22,23,30,33,38]
2	Agile effort	[20,22,23-30,33,38,41]
3	Agile estimation, Agile estimating	[20,27,29,30,33,41]
4	Agile software development	[21,24,27,37,40,43]
5	User story size/sizing	[20-27,29,31,34,40-43]
6	User story metrics	[29,32,36]
7	User story complexity	[41,43]
8	User story characteristics	[22,26,35,39]
9	Good story quality	[21, 35, 39,]
10	Agile requirements	[23,29, 39, 40,43]

The expression "agile software development (ASD)" has a substantial number of equivalent words and exchange terms that are utilized in literature; few of them are listed in table 1. Given that as I studied more and more literature and included to my set of known studies more alternative terms for ASD were discovered. Single word (i.e. "Agile") has been selected to get majority of its conceivable interchange terms, then ANDed it with "Software" to sift through totally unessential investigations from different areas. The study is further filtered by ANDing the terms "User Story", "Story Size", "Story Characteristics". M. Usman, Dybå and Dingsoyr in their SLR [19] [13] on ASD have also utilized a comparable methodology for the Term "Agile". Another SLR on usability in ASD by Silva et.al [15] likewise utilizes the expression "Agile" in the pursuit string as opposed to endeavoring to include the majority of its alternative. Moreover, the set of known basic studies was likewise utilized as a quasi-gold standard as proposed in [17] to evaluate the exactness of the inquiry string. The final search string is displayed underneath. Note that this string must be altered in like manner for every one of the databases.

(Agile OR "extreme programming" OR "Scrum" OR "feature driven development" OR "dynamic systems development method" OR "crystal software development" OR "crystal methodology" OR "adaptive software development" OR "lean software development") AND (estimat OR predict* OR forecast* OR calculat* OR assessment OR measur* OR sizing) AND (effort OR resource OR cost OR size OR metric OR user story OR story characteristics OR story qualities) AND (software).*

3.5. Primary and Secondary Search Strategies:

In primary search strategy I utilized the search string on well-known databases. The date filter was applied to get literature since 2001. I picked 2001 as the beginning date since this was the point at which the Agile Manifesto was published. The search result from each source was kept and managed in separate Excel sheets. At the end results from all databases were combined and duplicates were removed. After removing duplicated I ended up with 273 primary studies. Databases and the search result (before & after duplicates) are listed in Table 2.

Table 2: Search Results

Database	Before Removal of Duplication	After Removal of Duplication
Scopus	20	5
IEEE Explore	199	110
EI Compendex	3	3
Web of Science	15	7
INSPEC	16	4
Science Direct	15	11
ACM DL	278	124
Springer Link	10	9
Total	556	273

Selected databases cover all significant areas of Software Engineering, giving thorough inclusion of current SLR's topic. Different SLRs, for example, [19, 11, 14, 18], also utilized these databases for seeking pertinent primary studies.

In the next search stage all the basic studies regained in the first phase, was examined.

3.6. Study Selection Criteria

I demarcated Inclusion and exclusion criteria according to research question and goals of SLR.

3.6.1. Inclusion Criteria

1. Studies based on Agile Software Development
2. Studies related to agile cost/effort/size estimation
3. Studies related to agile requirements
4. Studies related to user story characteristics

5. Studies related to user story quality
6. Studies describe in English
7. Studies reported in any workshop/conference/journal.

3.6.2. Exclusion Criteria

1. Studies not based on Agile Software Development
2. Studies not related to agile cost/effort/size estimation
3. Studies not related to agile requirements
4. Studies not related to user story characteristics
5. Studies not relate d to user story quality
6. Studies not describe in English
7. Studies not reported in any workshop/conference/journal.

3.7. Study Selection Process

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

3.7.1. Title and Abstract level screening:

In this phase the titles and abstracts of all 273 papers were studied. Inclusion/exclusion criteria was applied to titles and abstracts to decide their significance to the current review. At that point those studies were excluded which are clearly not relevant to ASD. For example, as the term "Agile" was used in search string so I got some hits on the publications about "Agile Manufacturing". Thus, all those articles were excluded whose titles clearly indicated that the articles were outside the scope of this systematic review.

Sometimes the author used such witty title that make it very difficult to guess about the actual contents of an article. In such cases, the abstract of article was reviewed to make it clear whether the article was out of scope or not. All those studies were excluded whose main focus were not ASD or they did not include empirical data.

In such cases, the articles were incorporated for audit and the abstract was studied to make it clear whether the article was out of scope or not. Studies were barred if their center, or fundamental center, was not ASD or on the off chance that they didn't present observational information. After titles and abstracts screening, I ended up with 37 papers. It was noted that abstracts were of variable quality. As few

abstracts were missing or misleading. similarly, few abstracts gave little sign of what was in the full article. Therefore, at this stage, all those studies were included that showed some type of involvement with ASD.

3.7.2. Full text level screening:

In this stage all 37 papers were studied in detail. Inclusion/exclusion criteria was applied to the contents of all 37 papers. Ten papers were excluded in this stage. In case I did not have access to the paper, I emailed the author, however 1 paper could not be accessed by all means.

3.8. Quality Assessment (QA)

All 37 papers were evaluated independently according to 13 criteria given in [16] as shown in Table 3. Other researcher [19,11,12] also followed these guidelines given by [16] to customize their work. Using 3-points scale, each question was answered by Yes (Y=1), No (N=0), Average (A=0.5). Each study could get 0-13 points. Using the first quartile ($13/4 = 3.25$) as the end point for including a study. If a study got equal or more than 3.25 it would be selected otherwise removed.

Table 3: Quality Assessment Checklist adopted by [19,12, 16]

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

4. Results:

This section describes the outcomes for the overall SLR process and for research question also. Table 4 shows the numbers of studies going through various phases of the SLR. Details of the rejected papers in various phases are:

ten papers were excluded because of not passing the inclusion criteria, one because of a low-quality score and one paper was rejected because the study was already included in another paper. Separation of the ten papers; excluded on inclusive/exclusive criteria is as bellow.

- 7 papers were not conducted in ASD (exclusion criteria 1)

- 3 papers were not related to user story characteristics (exclusion criteria 4)

Table 4: Papers in Study selection and QA

Database	Search
Search Result	273
After titles and abstracts screening	37
Inaccessible papers	1
Excluded on inclusive exclusive criteria	10
Duplicate study	1
Excluded on the bases of low-quality score	1
Final papers(b-c-d-f)	24

Table 5: Year wise summary of papers in study selection and QA

Year	Before Exclusion	After Exclusion
2001	15	0
2002	10	0
2003	13	0
2004	11	1
2005	18	0
2006	20	0
2007	10	3
2008	17	3
2009	14	1
2010	11	3
2011	15	3
2012	21	3
2013	16	2
2014	16	6
2015	20	5
2016	17	4
2017	14	1
2018	15	2
Total	273	37

4.1. RQ: User story characteristics affecting software cost in agile software development

The following table 5 presents the characteristics of user story that can affect software effort estimation in ASD.

Table 5: User story characteristics

User story characteristics	Frequency
Independent	18
Negotiable:	17
Atomic:	6
Conflict free:	5
Valuable:	16
Estimable:	17
Testable:	16
Unambiguous:	14
Full Sentence	7
Unique:	6
Priority:	8
Flexibility:	6
Small	27

5. References

- [1] Z. Zia, A. Rashid, and K. uz Zaman, "Software cost estimation for component-based fourth-generation-language software applications," IET Softw., vol. 5, no. 1, p. 103, 2011.
- [2] B. W. Boehm, "Understanding and controlling software costs barry w. boehm, trw inc."
- [3] Hastie, S. and Wojewoda, S. (2015). Standish Group 2015 Chaos Report-Q&A with Jennifer Lynch. <https://www.infoq.com/articles/standish-chaos-2015>.
- [4] Torrecilla-Salinas, C J, Sedeno, J, Escalona, M J, Mejias, M, "Estimating, planning and managing Agile web development projects under a value-based perspective," Information and Software Technology, 2015; 61: 124-144.
- [5] Ishrar Hussain, Leila Kosseim, Olga Ormandjieva, "Approximation of COSMIC functional size to support early effort estimation in Agile," Data & Knowledge Engineering, 2013; 85: 2-14.
- [6] Sungjoo Kang, Okjoo Choi, Jongmoon Baik, "Model based dynamic cost estimation and tracking method for Agile Software Development," in Computer and Information Science (ICIS), IEEE / ACIS 9th International conference on, 2010: 743-748.
- [7] Parvez, A W M M, "Efficiency factor and risk factor based use case point test effort estimation model compatible with agile software development," Information Technology and Electrical Engineering (ICITEE), International Conference on, Yogyakarta, 2013: 113-118.
- [8] Garg, S, Gupta, D, "PCA based cost estimation model for agile software development projects," Industrial Engineering and Operations Management (IEOM), International Conference on, Dubai, 2015: 1-7.
- [9] Aditi Panda, Shashank Mouli Satapathy, Santanu kumar Rath, "Empirical validation of Neural Network Models for Agile Software Effort Estimation based on Story Points," Procedia Computer Science, 2015; 57: 772-781.
- [10] Kayhan Moharreri, Alhad Vinayak Sapre, Jayashree Ramanathan, Rajiv Ramnath, "CostEffective Supervised Learning Models for Software Effort Estimation in Agile Environments", IEEE 40th Annual Computer Software and Applications Conference, 2016.
- [11] Kitchenham, B. A., Mendes, E. and Travassos, G. H. Cross versus Within-Company Cost Estimation Studies: A Systematic Review. Software Engineering, IEEE Transactions on, 33, 5 (May. 2007), 316-329.
- [12] Azhar, D., Mendes, E. and Riddle, P. A systematic review of web resource estimation. In proceedings of the 8th International Conference on Predictive Models in Software Engineering (Lund, Sweden, 2012). 49-58.
- [13] Dybå, T. and Dingsøyr, T. Empirical studies of agile software development: A systematic review. Information and Software Technology, 50, 9-10 (Aug. 2008), 833-859.
- [14] Jalali, S. and Wohlin, C. Global software engineering and agile practices: a systematic review. Journal of Software: Evolution and Process, 24, 6 (Oct. 2012), 643-659.
- [15] Silva da Silva, T., Martin, A., Maurer, F. and Silveira, M. User-Centered Design and Agile Methods: A Systematic Review. In Proceedings of the Agile Conference (AGILE) (Salt Lake City, UT, 2011). 77-86.
- [16] Kitchenham B. and Charters S. Guidelines for performing systematic literature reviews in software engineering

- (version 2.3). Technical report, Keele University and University of Durham, 2007.
- [17] Zhang, H. and Babar, M. A. On searching relevant studies in software engineering. In Proceedings of the 14th international conference on evaluation and assessment in software engineering (EASE) (Keele, UK, 2010), 1-10.
 - [18] Schneider, S., Torkar, R. and Gorschek, T. Solutions in global software engineering: A systematic literature review. *International Journal of Information Management*, 33, 1 (Feb. 2013), 119-132.
 - [19] 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, PROMISE '14, pages 82–91, New York, NY, USA, 2014. ACM
 - [20] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper, "Improving agile requirements: the Quality User Story framework and tool," *Requir. Eng.*, vol. 21, no. 3, pp. 383–403, 2016.
 - [21] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper. The Use and Effectiveness of User Stories in Practice. In Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), pages 205–222, 2016a
 - [22] I. ul Hassan, N. Ahmad, and B. Zuhaira, "Calculating completeness of software project scope definition," *Inf. Softw. Technol.*, vol. 94, pp. 208–233, 2018.
 - [23] A. Kajirunga and K. Kalegele, "International Journal of Computer Science & Information Security," October, vol. 9, no. 10, pp. 49–52, 2015.
 - [24] M. Daneva and O. Pastor, "Requirements engineering: Foundation for software quality: 22nd international working conference, REFSQ 2016 Gothenburg, Sweden, march 14–17, 2016 proceedings," vol. 9619, pp. 171–187, 2016.
 - [25] A. T. Raslan, "Towards a Fuzzy based Framework for Effort Estimation in Agile Software Development," vol. 13, no. 1, pp. 37–45, 2015.
 - [26] A. E. D. Hamouda, "Using agile story points as an estimation technique in CMMI organizations," *Proc. - 2014 Agil. Conf. Agil.* 2014, pp. 16–23, 2014.
 - [27] Patanakul, P., & Rufo-McCarron, R. (Accepted/In press). Transitioning to agile software development: Lessons learned from a government-contracted program. *Journal of High Technology Management Research*. <https://doi.org/10.1016/j.hitech.2018.10.002>
 - [28] L. Williams, *Agile Software Development Methodologies and Practices*, 1st ed., vol. 80, no. C. Elsevier Inc., 2010.
 - [29] E. Miranda, Sizing User Stories Using Paired Comparisons, *Information and Software Technology (IST) Journal*, Vol. 51, Issue 9, September 2009, pp.1327-1337
 - [30] O. Liskin, R. Pham, S. Kiesling, and K. Schneider, "Why We Need a Granularity Concept for User Stories," pp. 110–111, 2014.
 - [31] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in agile software development," *Proc. 10th Int. Conf. Predict. Model. Softw. Eng. - PROMISE '14*, vol. 3, no. 7, pp. 82–91, 2014.
 - [32] S. Measurement and E. Forum, "Smef 2007 9 - 11," 2007.
 - [33] L. Buglione and A. Abran, "Improving the User Story Agile Technique Using the INVEST Criteria," pp. 49–53, 2013.
 - [34] R. Meli, "SOFTWARE MEASUREMENT EUROPEAN FORUM," no. May 2008, 2014.
 - [35] S. K. Tipu and S. Zia, "An Effort Estimation Model for Agile Software Development," vol. 2, no. 1, pp. 314–324, 2012.
 - [36] M. Cohn, "Writing Stories," *User Stories Appl. Agil. Softw. Dev.*, pp. 17–29, 2004.
 - [37] P. Heck and A. Zaidman, "A Quality Framework for Agile Requirements : A Practitioner ' s Perspective."
 - [38] N. Khanh, J. Daengdej, ... H. A.-I. C. on S., and U. 2017, "Human stories: a new written technique in agile software requirements," *Int. Conf. Softw. Comput. Appl.*, 2017.
 - [39] G. Lucassen, ... F. D.-R., and undefined 2015, "Forging high-quality user stories: towards a discipline for agile requirements," *Ieeexplore.Iee.org*.
 - [40] C. Patel and M. Ramachandran, "SoBA: A tool support for story card based agile software development," *Int. Conf. Softw. Eng. Theory Pract. 2008, SETP 2008*, vol. 2, no. 2, pp. 17–23, 2008.
 - [41] J.-M. Desharnais, L. Buglione, and B. Kocatürk, "Using the COSMIC method to estimate Agile user stories," *Proc. 12th Int. Conf. Prod. Focus. Softw. Dev. Process Improv. - Profes '11*, p. 68, 2011.
 - [42] J. Hyvönen, "Creating shared understanding with Lego Serious Play," *Proc. Semin. 58314308 Data- Value-Driven Softw. Eng. with Deep Cust. Insight*, no. 58314308, pp. 36–42, 2014.
 - [43] M. Ali, Z. A. Shaikh, and E. Ali, "Estimation of Project Size Using User Stories," no. Racs 2015, pp. 54–60, 2016.