



ISSN 2047-3338

# A Survey of Agile Software Estimation Methods

Hala Hamad Osman<sup>1</sup> and Mohamed Elhafiz Musa<sup>2</sup>

<sup>1</sup>International University of Africa, Faculty of Computer Studies

<sup>2</sup>Sudan University of Science & Technology (SUST), Faculty of computer science, Khartoum, Sudan

<sup>1</sup>laallaa83@yahoo.com

**Abstract**– Software measurement which mainly includes software cost and effort estimation is one of the most important activities in the software development process and it is still a critical and long standing issue, especially in agile software. There are few agile estimation methods and techniques proposed in the last 10 years such as story point (the main metric used in agile), planning poker and expert opinion. Many researches estimated cost, size and duration (CSD) for agile project by using traditional methods such as: COCOMO II, COSMIC, Function Point, etc with a little modification. Other researches an attempt to combine traditional estimation methods and agile estimation method such as Story point and FP. The main contribution of this paper is a general overview about agile software cost/size estimation methods that can show us the evolution in this field and covers most of the famous available and practiced parametric models founded on the literature. Also, this paper includes comments and discussion on the performance about the estimation models and description of several newer approaches for agile software cost estimation.

**Index Terms**– Agile Software Estimation Method, User Story, Story Point, Planning Poker and Software Traditional Estimation Models

## I. INTRODUCTION

**M**EASUREMENT is the process by which numbers and symbols are determined to characteristics of objects in the real world—this allows us to identify such objects according to defined rules. In software development, measurements are performed by using metrics, which are experimental designations of a value to an object aiming to characterize a definite quality of this object.

Software metrics are used to measure both the process and the definitive product characteristics connected with software development [1].

Software measurement which mainly includes software cost and effort estimation is one of the most important activities in the software development process and it is still a critical and long standing issue, especially in agile software development because uncertainty in early phase of software development.

Software measurement is still a key factor in software estimation, even in modern development methodologies such as Agile Software Development.

Software estimation gives the approximate calculation of software size, software development cost and effort, and development schedule for a particular software project.

The most important aspects to be measured comes development cost, product size (length, functionality, complexity), and required duration (how long they will take to do the work) (CSD), monitor the quality and track the development progress (velocity (team's rate)).

Much research has focused on traditional methods, little is known about estimation in the agile method arena and these researches classified Agile Estimation Methods (AEMs) based upon analogy and expert opinion.

Other researches an attempt to combine traditional estimation methods and agile estimation method such as Story point and FP, All these methods has its own strengths and weaknesses, unfortunately, the accuracy of these models is not satisfactory.

This paper consists of five sections, section II presents agile estimation methods, the performance of agile methods was discussed in section III, section IV presented new approaches for agile estimation and discussion and result in section V.

## II. AGILE ESTIMATION METHODS

In this section three metrics considered to estimate size of agile project: user story, Ideal day and planning poker.

### A. Estimation by story point

On an agile project it is not uncommon to begin iteration with incompletely specified requirements, the details of which will be discovered during the iteration.

For Agile development, a project may consist of one or more “stories” or “Sprints” those are required to produce this final result.

A sprint is an incremental piece of work used by the Scrum methodology (which is closely related to Agile), as well as by the Agile methodology itself. In Scrum, It has fixed length (typically 30 days) and shall not be extended.

The team's members assign a sizing unit called Story Point to each User Story.

A user story is a high level definition of a requirement containing enough information so that the developers can produce an estimate of the effort to implement it.

Each story is formulated in one or two sentences in the language of the customer, and typically written on an index card [2], [3], [4].

The estimates are important, as they are the basis for planning the next release in terms of prioritizing features and staffing the development team.

Story points are a unit of measure for expressing the overall size of a user story, feature, or other piece of work.

The number of story points associated with a story represents the overall size of the story. There is no set formula for defining the size of a story that is a key problem in the agile software metrics. There are two approaches to estimate story point, the first one is to select smallest story depend on the team opinion, and say that story is estimated at 1 story point. The second approach is to select a story that seems somewhat medium-sized and give it a number somewhere in the middle of the range you expect to use [5], [6].

In recent years, some researches investigated story point in agile projects [3], [5], [7].

Another study about estimation in XP and concluded that estimation in the scrum the same with the XP but it differs while XP represent pieces of functionality to be estimated and this is done every 2 weeks [5], [7].

An overall expected time for each of these stories is estimated by the developers and the customers then prioritizes the stories based on these initial estimates and on the business value of each one [8].

According to [8], the nature of agile methods often results in fixed budgets and a fixed schedule, and it is the scope of the project that remains flexible throughout.

On the other hand report in [9] announced that companies using agile methods usually lean towards "flexible contracts instead of fixed ones that predefine functionalities, price, and time".

The nature of agile may cause scope creeping due to changes in requirements which increase in cost for software project [10].

Agile methods aim to reduce the cost of changes throughout the development of a system, but not necessarily to reduce the occurrence of changes [11].

The authors in [12] identified the effort estimation within XP-projects is a part of the so called planning game. The recommended duration of the planning game can be assumed with 4 hours. Within this time, customers and developers work together.

### *B. Estimation by ideal day*

Another metric to estimate agile software size, Ideal time is the amount of time that something takes when stripped of all peripheral activities.

Elapsed time, on the other hand, is the amount of time that passes on a clock.

It is almost always far easier and accurate to predict the duration of an event in ideal time than in elapsed time. A team

member may spend time answering email, making a support call to a vendor [2].

When we estimate the number of ideal days that a user story will take to develop, test, and accept, it is not necessary to consider the impact of the overhead of the environment in which the team works.

If you choose to estimate in ideal days, assign one aggregate estimate to each user story. Some teams are tempted to estimate a number of ideal days for each individual or group who will work on a story.

There are many different between Story point and ideal day, firstly, estimating in story points is often faster than estimating ideal days. Finally, unlike ideal days, story points can be compared between team members. On the other hand, Ideal days more easily explained to those outside the team, easier to get started with, and it easier to predict initial velocity.

The author in [2] Prefers story points to estimate agile software size due to more powerful than ideal time.

### *C. Estimation by planning poker*

In the software industry, various techniques are used to combine estimates. One of the most recent additions is planning poker, introduced by James Grenning in 2002 [2].

Planning Poker method is the most commonly used method in Agile Estimation Methods; it combines expert opinion from different areas of software development, analogy and disaggregating into an enjoyable approach to estimating that results in quick but reliable estimates [12].

At the start of planning poker, each estimator is given a deck of cards. Each card has written on it one of the valid estimates. The cards should be prepared prior to the planning poker meeting and the numbers should be large enough to see across a table. Cards can be saved and used for the next planning poker session.

After the discussion, each estimator re-estimates by again selecting a card. Cards are once again kept private until everyone has estimated at which point they are turned over at the same time [2], [12], [13].

The goal in planning poker is not to derive an estimate that will withstand all future scrutiny.

## III. PERFORMANCE FOR EXISTING AGILE ESTIMATION METHODS

All these above methods have drawbacks in somewhere and may results is not satisfactory, For example: story points' counting and the velocity differ from team to team, so, it cannot be easy to estimate the time duration and the results can be subjective rather being objective.

Story point is relative values which have different values by different teams, in addition; there is no fixed formula for defining the size of the story. Also; story point alone is not sufficient to size measuring.

Furthermore, the velocity which a measure of the team's rate of progress is measured at the end of the iteration which makes it less dynamic.

There are no papers found accurate results for actual size/cost and cost estimated by story point.

*Planning poker:* it depends on existence of experts in agile project and that is may not available in that field since it new method and technique.

#### IV. NEW APPROACHES FOR AGILE ESTIMATION

In recent years, many empirical studies were proposed to present accurate and precise cost and size estimation for agile software project.

Most of these studies an attempt to combine the main method with other techniques such as story point and FP, planning poker and Delphi technique and using soft computing in agile software estimation.

In the following section, we overviewed some recent approaches in agile software estimation.

##### A. Combination FP and story point

The author in [6] states that one of the agile weaknesses is the widespread failure to measure projects using standards metrics, such as function points and concluded both Function Points and story Points measure the size of the software and are based on the count of logical expressions, function points address functional size while story points represent the business value of one user story.

Product Size = Functional Size + Non-Functional Size + Environments Variables Size,

Story Points = Function Points + Non-Functional Size + Environments Variables Size.

The limitation with this study This “formula” is not intended to be shown mathematically correct, but that represent that the functional size is part of the product size and finding a correlation between the whole (product size) and the piece (functional size) represents a valid proportion.

But a few studies [4] refute that states while they proposed correlation between story point and function point and presents conceptual differences between function points and story points. Furthermore; Agile teams separate estimates of size from estimates of duration [6], [14] while function points are complemented by other methods when it comes to effort and cost estimate such as COCOMO II.

In addition, function point not cover nonfunctional requirements while story point estimate is an amalgamation of the amount of effort involved in developing the feature, the complexity of developing it, and the risk inherent in building it.

Another study in [11] provides a systematic estimation and dynamic tracking methodology for agile projects. To estimate the effort of a project development, Function Points are used in addition to the story point.

The function points are determined based on the user stories of desired features of the product. Author in [6] introduced good and clear example for that combination.

##### B. Combination planning poker and other techniques

There are few empirical studies on the combining of estimates in software engineering, but there are some indications that combination may reduce the bias towards optimism in software estimates.

Reference [12] studied using planning poker for user story estimation and concluded Group estimation of user stories is an important part of Extreme Programming (XP), used for both planning releases and iterations.

User stories are commonly estimated using group processes, user story estimation may enjoy improved estimation accuracy compared to individual estimation processes .(however, with unstructured group estimation processes , factors such as company politics, group pressure and dominant personalities, may reduce estimation performance.

The author investigated two approaches planning poker and unstructured group estimation and approved planning poker is more accurate when the team has previous experience from similar tasks and planning poker is possibly less accurate when there is no previous experience from similar tasks. However this study needed more research since is limited to one team within one company.

Reference [12] combine planning poker with other techniques like Delphi (This method may be considered as a subset of expert judgment method, where group of experts are asked for the estimates until the consensus is reached) [12].

That study concluded planning poker may reduce optimism when compared to the mechanical combining of individual estimates, and may also, in some cases, be more accurate than unstructured combining of estimates in a group.

In [13] stated that techniques used to estimate agile development projects have typically been expertise-based, where the developers look to past projects or iterations, and draw on their own experiences to produce estimates for the stories.

##### C. Factors - based estimation in agile software

Many studies investigated factors that affected in agile cost estimation processes.

The authors in [15], [16] proposed a model for cost, size and duration for agile projects and it incorporates various vital factors namely: performance, complex processing, configuration, project domain and multiple sites.

After that, graded the intensity of these factors on the scale of low, medium and high based upon the complexity of the project. The limitation in that work Categorized all project domain (web application, MIS, military project) into 8 categories based on level of intensity for vital factors using square series (1 for low, 4 medium, 9 for high).

Another study by [15], [17] addresses some factors that direct impact on agile estimation process, but these factors proposed without any assigned to specific value.

##### D. Soft computing techniques in agile estimations

Recently; many researches attempt to automate the estimation process by building computerized models that can learn from previous estimation experience.

Soft computing consists of many techniques such as fuzzy logic, neural network, genetic algorithm, etc.

One of these studies in [18] uses a Neuro network to discuss and calibrate complexity weights of function point.

The proposed use the concepts and properties from fuzzy set theory to extend FP analysis into a fuzzy FP analysis, a

prototype that automates the calculation of FPs using the fuzzified model was created, but again the calibration was done using a small database comprised of legacy systems developed mainly in Natural 2, Microsoft Access and Microsoft Visual Basic, which compromises this work's generality [19].

Another study [20] combines the Neuro-fuzzy technique with different algorithmic models such as SEER-SEM.

Also; authors in [21], [22] state soft computing is very useful especially in handling uncertain and imprecise inputs and collinear characteristics.

Soft computing targets to use the tolerance for uncertainty, imprecision and partial truth in order to get low cost solutions, robustness and resemblance to human-like decisions [19], [23].

## V. DISCUSSION AND CONCLUSION

Estimation may use any of several methods. One of the most rigorous and accurate employs an estimate of the number of function points to be delivered and the expected productivity, effort, cost, schedule and staffing. For both Waterfall and Agile projects, these estimation metrics would typically be determined prior to the start of project development. For Agile projects, however, this means that objectives are determined at the beginning of a story or sprint, rather than at the beginning of all development. The reason is that too little is known about the requirements until each story/sprint begins.

Software development has been the subject of much criticism. Its critics point to the contention that it is difficult to estimate effort related to projects accurately, because at the early phase of a project there is uncertainty about the project scope. Moreover, there are inherently many requirements changes associated with agile software development which increase the risk of scope creep.

So; Customer changes requests and unclear requirements as the main causes of the failure of agile software estimation.

The Main metrics in agile such as story point that used and practiced only in scrum and XP methodologies.

Story points are faster and most used in agile size estimation than ideal days.

Ideal day easier and accurate to predict the duration than in elapsed time estimation by user story have some limitation as there is no set formula for defining the size of a story, it failure to measure projects using standards metrics, such as function points, not sufficient alone to measure size, but when that combine with FP may obtain a good and an accurate result.

Planning poker may causes more inaccurate results in size estimation, despite its most method used in agile software, since it depends on expert opinion and analogy which are both may not available in agile project. Many factors affected the size, cost and velocity for agile projects and may present precise result when we build an algorithmic model based on those factors.

Many publications addressed some factors that direct impact on agile estimation process, but these factors in most researches proposed without any assigned to specific value.

Estimation-base on soft computing as neuro fuzzy, more accurate and recent but it need to real and huge dataset from industry to implement a model for agile software estimation.

Software cost estimation needs more efforts from the researcher either in academic or in industry field to cooperate together to enhance the estimation process and produce the best results, software cost estimation currently need a fully systematic review for the available research and classify it according to its content and trends, this will ease the research process and can clarify the shortcomings in this field. From above, it's clearly there is need systematical approach for software cost estimation may not depend on software size only. Also; is that no single technique is best for all situations and story point is not sufficient alone. So; there is a great need to combination between more than two models to ensure the accurate estimation such as: FP with story point, neuro fuzzy with COCOMO and neuro network by using any repository for real dataset.

## REFERENCES

- [1]. Manfred Bundschuh, Carol Dekkers, "The IT Measurement Compendium Estimating and Benchmarking Success with Functional Size Measurement, 2008.
- [2]. M Cohn," Agile Estimating and Planning", Addison-Wesley, (2005).
- [3]. M Cohn,"user stories applied: For Agile Software Development", 2004, AddisonWesley, Doston
- [4]. S. Kang, O. Choi, J.Baik, "Model-based Dynamic Cost Estimation and Tracking Method for Agile Software Development", 9th IEEE/ACIS.2010, IEEE
- [5]. Jones, C.: Applied Software Measurement. McGraw Hill, New York (2008)
- [6]. C. Santana, F. Leoneo, A. Vasconcelos and C. Gusmão, "Using Function Points in Agile Projects". (Eds.): XP 2011, Springer-Verlag Berlin Heidelberg
- [7]. Schwaber, Ken and Mike Beedle, "Agile Software Development with Scrum", 2002, Prentice Hall.
- [8]. Lovaasen, G, "Brokering with eXtreme Programming. In XP Universe 2001. Raleigh, North Carolina
- [9]. Highsmith, J, "Agile Project Management: Principles and Tools: Cutter Consortium, 2003
- [10]. Siobhan Keaveney, "Cost Estimation In Agile development Projects",
- [11]. Highsmith, J., and A. Cockburn. 2001. Agile Software Development: The Business of Innovation. IEEE Computer 34 (9):120- 127.
- [12]. Fuqua, A., "Using Function Points in XP –Considerations", In: Proceedings of Extreme Programming and Agile Processes in Software Engineering, Springerlink.
- [13]. N.Christian,"An Empirical Study of Using Planning Poker for User Story Estimation", Proceedings of AGILE-2006 Conference (AGILE'06), 2006.
- [14]. K. Moløkken, "Combining Estimates with Planning Poker an Empirical Study",
- [15]. Highsmith, Jim, "Agile Project Management: Creating Innovative Products", Addison-Wesley Professional, 2004.
- [16]. Shilpa Bhalerao1, Maya Ingle, "Incorporating Vital Factors in Agile Estimation through Algorithmic Method", 2009.
- [17]. S. Ch, R. Lavanya, V. Kanchana, "Multi-Criteria Approach For Agile Software Cost Estimation Model".
- [18]. Dac-Buu Cao, "A survey study of critical success factors in agile software projects", 2007.

- [19]. U. R. Saxena, "Software cost estimation using a Neuro-Fuzzy algorithmic approach", IEEE, 2013.
- [20]. Vikas Kumar, Sweta Pandey, "Accurate Software Size Estimation Using the Updated Function Point Analysis Model", 2010.
- [21]. Ch.Satyananda Reddy and KVSVN, "An Optimal Neural Network Model for Software Effort Estimation", 2010
- [22]. Wei Lin Du<sup>1</sup>, Danny Ho<sup>2</sup>, Luiz Fernando Capretz, "Improving Software Effort Estimation Using Neuro-Fuzzy Model with SEER-SEM", 2010.
- [23]. C. S. Reddy, K. Raju, "An Improved Fuzzy Approach for COCOMO's Effort Estimation using Gaussian Membership Function", Journal of Software, Vol. 4, No. 5, pp. 452-459, 2009.