ISSN 2047-3338

Limitations of Function Point for Agile Software Environment: A Case Study

Hala Hamad Osman[1,2], Mohamed Elhafiz Musa[1], Abdelgaffar Hamed Ahmed[1]

[1]Faculty of Computer Science, Sudan University of Science & Technology (SUST), Sudan
[2]Faculty of Computer Studies, International University of Africa, Khartoum, Sudan
[1]laallaa83@yahoo.com

***Abstract*– Software measurement is an important area of research as it focused on the estimation of the cost and size of software. There are a different software estimation models that are used in industry to provide accurate and reliable estimates of the software costs and size. Despite their contribution towards software estimation, such models need to be standardized, validated, to incorporation estimation. Function Point (FP) metrics are used for studying software size, productivity, quality and costs. FP provides good result in traditional software estimation environment. But, it didn't give suitable results when used for measuring agile software. This paper investigates FP analysis in agile software development measurement through two case studies developed by scrum method. This study proved that FP in an original form is not a suitable metrics for agile software estimation. Hence, this paper states that the traditional FP needs more tangible enhancement to be used in agile software properly.**

***Index terms*– Software Estimation Models, Function Point, Agile Software, Story Point (SP), Using Function Point in Agile Projects and Agile Estimation Methods**

I. INTRODUCTION

SOFTWARE measurement is an important aspect for both software development methods (traditional and agile). There are a different software estimation models that used in industry for providing accurate and reliable estimates of the costs and size of software but there is not a general/ complete model to estimate different attributes of software (cost, size, quality). The resulting estimates are directly related to the other aspects and activities of the entire software project such as project planning, development and construction [1]. Software measurements methods s are growing and can be classified into three categories are: quantitative (formal), expert judgment-based (human-based) and ad hoc methods (others). [2] Agile software methodologies introduced some of models and techniques to estimate software sizing and many aspects of software development [3], [4]. So all these methods have drawbacks that challenge their ability to offer accurate and satisfactory results. Furthermore; the velocity that introduces the team's rate of progress is measured at the end of the iteration which makes it less dynamic [4], [5]. Also; there are no methods for estimating and monitoring the performance of agile projects based on a standardized procedure [6]. FP is one of the most common used to estimate software sizing in the early phase of the software development process, that the reason is low maturity of the software measurement practice.

The remainder of this paper is organized as follows: Section II is a summary of relevant theories, In Section III the research methodology is described. Section IV presents the results and Section V discusses the result's validity. Finally, section VI presents the conclusions.

II. RELATED WORK

Software measurement methods:

There are a different of software estimation methods that used in industry for providing accurate and reliable estimates of the costs and size of software but there is not a general/ complete model or method to estimate different attributes of software (cost, size, quality). The resulting estimates are directly related to the other aspects and activities of the entire software project such as project planning, development and construction [1].

A) Traditional software estimation methods

This category includes three models: Quantitative (formal) estimation methods [7], Ad hoc models [10] and Human – based models [11].

The limitations of these models include: high time consuming and expensive, difficulty of use by non-technical personnel, extensive training is required and the value will not be known until the software is fully developed [1], [8], [9] in addition are not satisfactory for the customers. [10] furthermore the Human-based model is that the estimates are based on an opinion and experience of the estimators which may not guarantee the accuracy of results [1], [12], [13].

B) Agile software estimation models

In Agile software estimation models expert opinion, analogy, or disaggregation are used to arrive at the estimates of the story points [11], [14], [15] There is no set formula for defining the size of a story that is a key problem in the agile software metrics [16], [17]. However, these models suffer from some drawbacks in somewhere and may results is not satisfactory, because story points' counting and velocity differ from a team to another, is difficult to use them to estimate the time duration objectively and it’s not sufficient to size measuring [5], [11], [17].

C) Function Point - enabled in agile projects

Function Point is an indirect quantitative measure of application software functionality and size; also for studying software productivity, quality, costs, risks, and economic value. Also; FP can be estimating cost software very well because it acts as a basis for software measurement, comparison, and analysis and its can provide a mechanism to track and monitor scope creep. [4] [17] many steps in [18]-[21] to calculate FP in accurate way especially in traditional software development. In [22], [23] [24] conducted it's difficult to count the functions of a software system when we try to use on modern software development.

In recent years, many researches on function point metric were proposed to improve software size estimation by redefine Unadjusted Function Point to be suitable with specific kind of application domain [22] or add new System General Characteristics (SGC) to be adopted with new field of software development. [25] Many difficulties exist when some are metrics to estimate the size and cost of agile software by using traditional metrics such as FP, COCOMO; due to the nature of agile software. While Story point is not sufficient alone to record an accurate result [17] FP can compatible with story points to estimate software size on agile projects. [4], [17] These studies concluded with theoretical relationship between Story Point and function Point but not empirical study. Also; the process of calculation FP from user story and story point need more details and analysis, and specific tasks for each story. [17], [19] The strengths of the FP include its increasingly wide use in software contracts, can be used to determine whether a tool, a language, an environment, is more productive when compared with others and can be used for measuring size software applications accurately and independent of languages or tools [19], [26].

On the contrary there are some weakness as: function points are almost never used on large systems > 10,000 function points in size, that is causes it counting is slow [4], [19], [27] According to [28] function point metrics can easily become a universal metric used for all software applications and for all software contracts in all countries. However, there are some logistical problems with function point metrics when used to estimate agile software sizing that need to understand, analysis, adaptation and overcome in order for function point metrics to become the primary metric for agile software measurement.

III. METHODOLOGY

A) Introduction

The methodology used in this study based upon constructive and action research method. This study uses case studies to reflect on the applicability of FP analysis in agile environments. The main questions of this study are:

i) Can a systematic metric be used for Cost, Size and Duration (CSD) estimation in agile software?

ii) Can function point analysis be used alone to estimate size for agile development projects?

B) Limitations of FPA in agile environnements

When we want to develop any system by agile method, organized the system requirements into story form to prepare product backlog that includes features, bugs, technical work [27] When wanted to express feature we use user story that is a high level definition of a requirement containing enough information [27], [28].

C) Case Studies

The used case studies follow these steps:

i) Collect all system requirements in user story form (Product backlog).

ii) Break down the user stories into small task (Sprint backlog).

iii) Mapping small tasks into corresponding FP’ elements to estimate software size and effort form small tasks (step 2).

iv) List the problems and difficulties faced the team when they built the above three steps

Step 1 and 2: user stories form and small tasks from stories:

Case study 1: taken from [16] about Deep Black and white game

Case study Two: about E- Registration System developed by T&M Company

Step 3: mapping small tasks into corresponding types of FP:

Table I: Mapping of small tasks into corresponding FP’ elements

No	Task	Types of FP	Case No
1	Draw empty board	External Input (EI)	1
2	Write automated test for ship win	ILF	1
3	Have move engine pursue an unblocked ring	External output (EO)	1
5	Automate test cases for making activated code	Unspecified	2
6	Identify test cases for payment process	Unspecified	2
7	Design errors massage.	EI, EO	2
8	Upload a lecture video (storage constraint).	Unspecified	2

IV. RESULTS

Step 4: List the problems / difficulties:

Some problems emerged as a result of implementing the above three steps, these are:

Problem 1:

It is difficult to estimate effort / size related to projects accurately, because at the planning phase of the project there is uncertainty about the project scope, due to the rapid change in the requirements in the agile software development (Case study 1).

Problem 2:

User story express customer's needs that involve more detailed in developing the features, risks, complexity and all quality and technical requirements. In addition, story point estimate complexity of the story, while function point does not cover non-functional requirements (Case study 1, 2).

Problem 3:

Calculation of function point from a story is more difficult to apply, since:

a). It requires detailed analysis about the user story that it's not available on agile project due to the nature of documentation in this field.

b). It is written in natural language form, may lead to different results in the size measuring due to ambiguous and variation in the language and lead to the errors in case of inexperienced agile team

c). Some system holds huge volume of data. FPA rarely considers the data storage. In addition, it does not consider the size of simulations, animations and additional document effects (Case study 2).

Problem 4:

In some cases; the development team focuses on the tasks that help them to develop the system, regardless of classifying specific tasks into design, code, test, etc.

This may complicate the extraction of FP from stories too much. Since, all these tasks -which are the real work of agile development- are not related directly to any types of FP unadjusted counting. So, the elements of FP are not suitable when software development methodology become modern such as agile software (Case study 1).

Problem 5:

General System Characteristics (GSC) are used to adjust Function Points that are not sufficient for agile development, because there are many factors have influence on the agile project not shown in (GSC). FPA not much considering data transfer facilities. However; some systems have fund transfer facilities, which require high end security codes. Each line of this code has much weight. FPA does not reflect the importance to this code. Also, the team of agile development is one important factor to deliver useful product to the customer, (GSC) not include any factor or character related to the team' velocity.

V. CONCLUSION

Function Point is one of the most popular metrics widely used in traditional software measurement and it became a standard metrics in software industry, while story point considered as main metrics in agile software sizing, especially in scrum and XP methodologies.

Story point alone didn't give accurate results when used in agile projects; also FP in its original form didn't obtain suitable and successful results. Many studies investigated new approaches to produce best results in the field of agile software measurement that are combination between FP and SP.

Many publications addressed some factors that have direct impact on agile estimation process, but these factors in most researches proposed without any assigning of specific measuring value. Also, some researches proposed new form of FP which is a suitable to work in agile environment by extend and enhance one of the two categories of FP, the first is unadjusted FP, the second is adjusted FP but not both categories.

From above discussion and case studies, it's clearly that FP in its original form is not appropriate metrics to estimate agile software size. So; there is a real need to enhance FP by modifying and replacing the unadjusted and adjusted FP for mainstreaming with agile software environment.

Also; there is no single technique that is best for all situations. Therefore, story point is not sufficient alone, so combination between SP and FP may provide more accurate results.

Future Work: Future work on software measurement will encompass both theoretical and practical activities. On theoretical side, studies are needed to use multi_ measurement techniques that an important area to analyze and investigate. Many studies focused on using only one metric for estimating software size.

On the application side, measurement needs to be introduced in traditional applications environments and in new ones, such as Service Oriented Architecture (SOA) and cloud computing applications. Our future work will focus on the extension and improvement for Function Point metric to be adopted and used in reliable way in the agile environment.

Table II: Summary of problems for all case studies

No.	Problems	C 1	C2	Recommendation	Suggested Solution
1	requirements changing, scope creep	√	√	Nature of agile software	Iterative model
2	No relationship between real tasks and current FP unadjusted count,	-	√	-	Redefine factors of FP to compatible with agile concepts and practices.
3	Difficulty to apply	√	√	a). story card alone not sufficient to calculate FP b). not all tasks in template can apply to any system	Model for story C's Card, conversation, confirmation
4	General system characteristics (GSC) are used to adjust Function points are not sufficient for agile development,	√	√	-	Questionnaire To determine which factors influence on agile' team. Velocity

REFERENCES

[1]. Lafferty, M., T., "Software Effort Estimation Accuracy: A comparative Study of Estimations Based on Software Sizing and Development Methods", PhD thesis, Capella University, 2010.

[2]. J Leinonen, "Development effort estimation process in agile software development context", Master Thesis, 2016.

[3]. Cohn, M," Agile Estimating and Planning", Addison-Wesley, Reading, 2005

[4]. Santana, C., Leoneo, F., & Vasconcelos, A, "Using function points in agile projects", Springer-Verlag Berlin Heidelberg. pp. 176–191, 2011

[5]. Hala, H., O, & Mohamed, E. M," Survey of Agile Software Estimation Methods: International Journal of Computer Science and Telecommunications IJCST, Volume 7, Issue (3), 2016

[6]. Radenko Corovic, "Agile Estimating and Agile EVM", unpublished

[7]. Fedotova, O., Teixeira, L., Alvelos, H, "Software Effort Estimation with Multiple Linear Regression: review and practical application", JOURNAL OF INFORMATION SCIENCE AND ENGINEERING (ISE), 2011

[8]. Touesnard, B," Software Cost Estimation: SLOC-based Models and the Function Points Model", UNB, 2004

[9]. Boehm, B., & Abts, C," Software development cost estimation approaches - A survey", Annals of Software Engineering. Vol. 10. pp. 177-205, 2000

[10]. Alverson," Software Development Lifecycle", Spring, CSE 403, 2007

[11]. Moløkken-Ø, Kjetil H, & Nils C," Combining estimates with planning poker - An empirical study", In Proceedings of the Australian Software Engineering Conference (ASWEC'07), IEEE, 2007

[12]. F.J. Heemstra, R.J. Kusters, "Software cost estimation and control : lessons learned", In European software cost modelling meeting : proceedings, 27-29 May 1992, Munich, Germany

[13]. Shepperd, M., Schofield, C. & B. Kitchenham," Effort Estimation Using Analogy", in Proceedings of International Conference on Software Engineering. pp. 170- 178, 1996

[14]. Ambily O. A, "Agile Software Development- An Approach to Light Weight From Heavy Weigh", International Journal of Engineering Science and Technology (IJEST). ISSN: 0975-5462. Vol. 3 No. 1 Jan, 2011

[15]. Taghi, J., Hazura, G., & Abdul, Z., "Obstacles In Moving To Agile Software Development Methods", At A GlancE. Journal of Computer Science. 9 (5): 620-625, 2013 ISSN 1549-3636, 2013

[16]. Cohn, M," User stories applied: For Agile Software Development", Addison RXesley, Doston, 2005

[17]. Sungjoo, K., Okjoo C., & Jongmoon, B, "Model-based Dynamic Cost Estimation and Tracking Method for Agile Software Development" Presented at International Conference on Computer and Information Science. IEEE/ACIS, 2010.

[18]. Albrecht, A. J," Measuring Application Development Productivity", IBM Corporation, 1979.

[19]. Dekkers, C, David Consulting Group, "Story Points or Function Points or both", 2015 Retrieved from \http://www.ifpug.org/Articles/Dekkers-CountingAgileProjects.pdf [accessed on 2017-05-30].

[20]. Pressman, R., S," Software Engineering A Practitioner's Approach", ISBN 978–0–07–3 37597–7 MHID 0–07–337597–7, 2010

[21]. Jones, C, "Estimating software cost", tata Mc- Graw -Hill Edition, 2007.

[22]. Bingchiang, J, "A Specific Effort Estimation Method Using Function Point", Journal of Information Science and Engineering 27, 1363-1376, 2011

[23]. Gianfranco Lanza, "Function Point: how to transform them in effort? ", Measurement European Forum, Milan, Itlay, pp 127-136, 2008.

[24]. Kemerer, C," An empirical validation of software cost estimation models", Communications of the ACM, Vol. 30, No. 5, pp. 416-429, doi: 10.1145/22899.22906, 1987.

[25]. Kanakalata & Mahadevan, 2012.

[26]. Scott G. Retrieved from http://www.qpmg.com/pdf/articles/Quantifying_the_Benefits_Using_Function_Points.pdf, 2016.

[27]. Jones, C, "Strengths and Weaknesses of Software Metrics", Version 5, March 22, 2007.

[28]. Jones, C, "Function Points as a Universal Software Metric", July 13, 2013.

[29]. Cohn, M, Retrieved from https://www.mountaingoatsoftware.com/agile/user-stories, 2017, [accessed on 2017-05-25].