International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 5 (2018) pp. 2192-2196

Early Effort Prediction for Agile Software Development Using Historical Data to Improve Productivity

[1]Mrs. R. Manjula and Dr. Mrs. R.Thirumalai Selvi[2]

[1]Research Scholar, Bharathiyar University, Coimbatore, Tamilnadu, India.
[2]Assistant Professor, Govt. Arts College, Nandanam, Chennai, Tamilnadu, India.
Email: cpmanjula76@yahoo.com

Abstract

Research on effort estimation in software development has been conducted for decades and has produced quantities of models and Tools. Unfortunately, reliable initial estimates are quite difficult to obtain because the lack of detailed information at an early stage of the development. Agile methods, which represent an emerging set of software development methodologies based on the concepts of adaptability and flexibility, are currently touted as a way to alleviate these reoccurring problems and pave the way for the future development. The estimation in agile software development methods depends on an expert opinion and historical data of project for estimation of cost, size, effort and duration. In case of the availability of historical data, project effort can be predicted. This paper focuses on the research work in Agile software development and effort estimation in agile. It also focuses the difference between effort estimation with historical data and without historical data.Section1 describes Introduction. Section 2 contain Related Work in this area. Section 3 explain correlation between Functional effort and development effort. Section 4 has limitation of story point. Section 5 include proposed method and its implementation.

Keywords: Agile Software Development, Functional effort, Effort Distribution, Effort summary

INTRODUCTION

The popular and most common approach for effort estimation in agile methodology is subjective estimation. Although this approach is simple and easy to apply, estimates are highly biased. In some agile teams, effort estimation is biased on their previous iteration actual effort and hence effort estimation is useful only for remained user stories. In addition, application of planning poker is one of the most popular practices for many agile teams in planning and predicting effort before starting each iteration. In this method, almost all developers collaborate in estimation. Thus, no one estimates for all and also everyone estimation is often. In this practice, each member gives a point to a story and the final point of that story is the mean of its assigned point. Nonetheless, User story point is not objective and cannot define a standard practice for estimation of software size and complexity. Moreover there is no evidence that estimation in this way is more adequate than the famous wideband Delphi, but at least it is funny for the team and motivates them in estimation practice. There are also many reports about using, usual software measurement practice which were used in some specific agile methods such as scrum and XP with appropriate estimation.

RELATED WORK

This section identifies some of the key research work in the area of Agile developments. Taghi et al., Investigated on Current measurement practices in agile software development methods and reviewing agile version of common software measurement International consortium[2]. It published a measurement guideline for ASD methods and in this guideline software size will be estimated. Rashmipopli and Nareshchauhan, focuses the problem in current agile practices thereby propose a method for accurate cost and effort estimation and calculate cost, effort and duration of small and medium size projects can be calculated officially and reduces the risk of falling project in chaos by providing realistic figures of estimation[4]. Florian Raith & Gudruti Klinker proposed a metrics to evaluate accuracy of estimates and the metrics are integrated into planning poker process. Computer aided tools are used to collect and evaluated necessary data. Finally prototype is developed but, there was a poor portability and necessary investment in technical equipment[3]. Dauh Nguyen-cong& De Tran-Cao published a Review of Effort Estimation studies in agile, iterative and incremental development. This model based on monitoring based and expert based estimation approaches are popular. There is lack of studies that focus on the impact of the properties of historical/current project data on estimation results, benchmark data and composite models[1]. Ishrar Hussain, Leila Kosseim, Olgaormadijiena discussed how user stories are classified into functional size with the use of Textmining approach and the functional size is measured by Cosmic[5].

The review of research estimation shows, majority of studies focus on estimation results for planning and tracking project. There is a lack of studies that focus on the impact of properties of historical current project data. This paper includes correlation between function effort and development effort and project scheduling, function point estimation in agile projects.

CORRELATION BETWEEN FUNCTIONAL EFFORT, DEVELOPMENT EFFORT AND PROJECT SCHEDULING:

Functional effort gives the amount of functionalities of requirement (Figure 3.1). A common example for an appropriate entity within traditional effort estimation is function points, which are measured with the function point analysis. Development effort represents the necessary amount of work to realize a certain quantity of functional effort, which is commonly measured in person days.

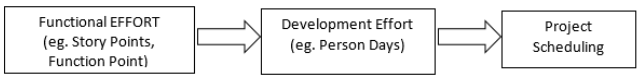

Figure 3.1

Productivity describes the relationship between functional effort and development effort

$$\textbf{Productivity} = \frac{FunctionalEffort}{Developmenteffort}$$

AGILE ESTIMATION CONCEPT BASED ON STORY POINTS (SP) AND ITS LIMITATIONS:

Some authors propose to use Story Points (SP) as the measure of software's functional size. Without performing an in-depth analysis, it is clear that SP do not provide an accurate measure of the software's functional size:

- Story Point calculation is not based on a standard method.
- There is no relationship between Story Points and Function Points.
- Story Point calculation is not based on the elementary processes, which are at the core of Function Point calculation.
- Story Points differ from one team/project to another and have meaning only to the members of the team who estimate them.

The fact that a Sprint refers to a period of time in which a set of functionalities has to be developed and not to functionalities that are already completed and delivered, which is standard in the calculation of Function Points, represents another difficulty for the application of Function Points in agile development. Indeed, the functionality developed in one Sprint can be improved in a subsequent Sprint in a way that the sum of the FP of all Sprints in one project is usually higher than the functional size of software measured at the end of the project.The current agile practices that meet the requirements of organizations in project estimation as well as in project performance monitoring are

- Obtain and Analyze the Estimate of Project Size and Efforts Early in Project Life Cycle
- Compare the Performance/Productivity - Internal and External Benchmarking
- Monitor the Progress of Agile Projects
- Collect Data on Past Projects in Order to Improve Future Projects Performance

Function Point measure is independent of language, development method; hardware/platform used and is a better candidate for benchmarking across organizations. Function Points can be used to effectively derive empirical formulae and pinpoint scope for improvement. Despite these benefits; due to its very nature, Function Points have to be counted manually. The counting process cannot be automated. A great level of detail is required to estimate the software size in terms of Function Points. Information on inputs, outputs, screens, database tables, and even logical records and fields are required to perform Function Point Analysis (FPA).

PROPOSED METHOD:

In this, historical data collected based on the criteria which should be given in the algorithm. The new project can be developed by using the historical data, and the extended features are later by introducing product backlog of new. Historical data are used to identify the functions easily, and predict the total effort.

Algorithm for Estimating Size of Agile Project Using Function Point

- For all(T, D, DT,WE,Ts) of OP = CP then Historical Data Is true(i.e Product Backlog is available)
- If (Funct. Specf. >15%) then
- Est. Function Point(FP)
- Use FP to Est. Size of CP
- Total Effort decreased
- Less effort for development and post productivity
- Else
- Total Effort Increased
- More effort for Development and post productivity
- End if
- End For.

[Note: T- Technology, D- Domain, DT- Development Team, WE- Working Environment, Ts – Tools, OP = Old Project]

EFFORT SUMMARY

Effort summary contain details about percentage of effort in hours of Requirement validation, design, Development unit testing, system testing, user acceptance testing and deployment etc. Formulas are given below to calculate the total effort (hrs).

$$\text{Total Effort (hrs)} = \frac{(EffortPostProductivity)*100}{\%\,ofEffortinDevelopment/unitTesting}$$

% of Effort in Development/unit testing = 100-(RV+SD+UT+ST+IT+UAT+G_L)

Effort (in hrs) of Development/unit testing = Effort Post Productivity

$$\text{Effort Post Productivity} = \left(\frac{RawDevEffort*(100+10)}{100}\right)$$

Raw Dev. Effort = $\sum_{i=1}^{N} AEi$(i.e, sum of Attribute Efforts (in hrs))

Efforts (in hrs) = Units of configuration items * complexity value

IMPLEMENTATION

Original data's are collected from the project development company XXX. This project based on Insurance domain. Using this, total efforts are estimated in two different cases. To calculate the total effort, Estimation guideline are prepared based on classification guide. This estimation guidelines describe list of use cases and its different complexity. It explain whether the use cases applicable to simple, medium and complex, or not applicable. Some use cases include simple, medium and complex. But some use cases has only simple complexity. Estimation Guideline descriptions are used to prepare Master Data Effort. Master Data Effort represent list of use cases and its required complexity values (in hrs). The complexity values are assigned by expert. Estimation guidelines (in hrs) are used to prepare the standard offering table. This table contains configuration items, no. of units, complexity, efforts, raw dev. effort, Effort post productivity and total effort. At the initial stage, standard offering table assign complexity value as simple to all the use cases. Effort for each use case is calculated with the help of Estimation Guideline (in hrs) which is in Master Data Effort table.

Rough estimation in Master Data Effort can be calculated by

Raw. Dev. Effort = Sum of Use Cases Efforts

Effort Post Productivity = Raw. Dev. Effort * 1.2

Total Effort = Effort Post Productivity * 1.6

If historical data is present then the total effort prediction can be estimated easily (ie.) effort is reduced. Because the complexity of the project requirements are set into medium. But, in absence of historical data total effort required to develop a project need more, and the complexity is set as complex. Complexity of a project can be simple, medium and complex. In case of new project, complexity is always set as complex. Because it has very less details about the project requirements. Table 5.2.1. Shows the effort distribution of use cases.

Table 5.2.1: Effort distribution(medium)

CONFIGURATION ITEMS	EFFORTS (in Hrs)
Custom Objects (1 object)	9
Custom Fields (10 fields)	52
Custom Fields (10 fields)	4
Data Migration	250
VF Page	84
Apex Class	80
Triggers	32
Test Class	60
Email Templates (1 Template)	3
Report Type (1)	2
Work flow (5)	10
Validation (5 Validation)	32
Security	32
Page Layout (1 Layout)	72
Record Type (2)	4
Profile(1)	96
Web Service	48
Report (1)	12

Each use cases efforts are estimated by multiplying the configuration item unit with the complexity values. The chart1 represents if complexity medium, then assume historical data is available. X axis represents use cases configuration items and Y axis represents Efforts(in hrs) needed for each use cases.

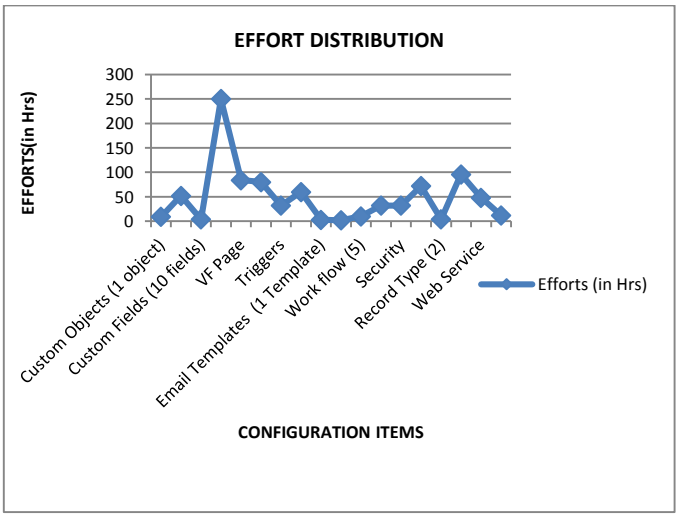

Figure 5.2.1

Table 5.2.2 has list of use cases configuration items and its Efforts(in hrs). The complexity of the use cases are set into complex and the requirement validation between the historical data and the current project are set into <15%. Table 5.2.1 is differed from Table 5.2.2, Some of the use cases are set into

complex value, few of the use case are in simple value. So, the Effort spent for the complex use case also increased.

Table 5.2.2: Effort Distribution (Complex)

CONFIGURATION ITEMS	EFFORTS (in Hrs)
Custom Objects (1 object)	9
Custom Fields (10 fields)	78
Custom Fields (10 fields)	6
Data Migration	350
VF Page	168
Apex Class	128
Triggers	48
Test Class	90
Email Templates (1 Template)	12
Report Type (1)	2
Work flow (5)	16
Validation (5 Validation)	48
Security	32
Page Layout (1 Layout)	144
Record Type (2)	4
Profile(1)	96
Web Service	64
Report (1)	24

Figure 5.2.2 reperesents Effort Distribution between use cases configuration items and Efforts needed in each use cases

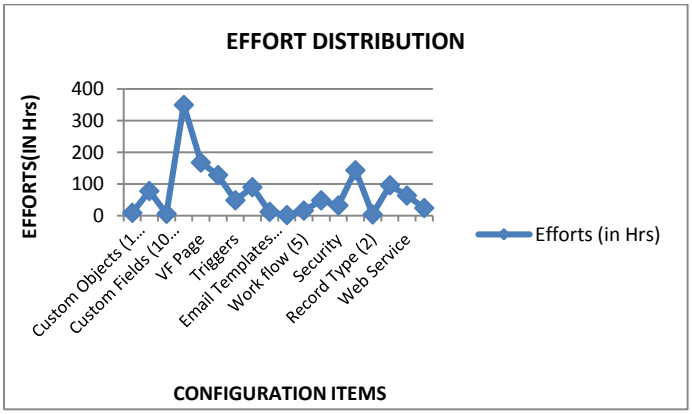

Figure 5.2.2

COMPARISION BETWEEN THE EFFORT SUMMARY

Table 5.3.1 shows the Effort Summary between with and without Historical Data. If the Historical Data is available then the Requirement validation is map to the current project, So, the Effort(%) is set as >15% and the solution design has the same value as the Requirement validation. In case, Historical Data is not available, then the Requirement validation and the solution design is set as less than 15 %,

Table 5.3.1: Effort Summary Comparison

Work Title	Effort (In hrs>15%)	Effort (In hrs<15%)
Requirement Validation	336.6	351
Solution Design	336.6	351
Development/ Unit Testing	1009.8	1491
System Testing	224.4	292
User Acceptance Testing/ Deployment	224.4	292
Go-Live	112.2	146
Total Effort	2244	2923
Project Management Effort	2244	2923

Figure 5.3.1

Figure 5.3.1 represent the variation between the efforts in Effort Summary.

A Project with historical data and without historical data comparison is displayed in the above chart. Which shows the effort spent for the project. Effort can be reduced if historical data is available. With respect to the algorithm, function points of the new project are calculated with the help of historical data and extended features of the new project effort are calculated from the product backlog.

CONCLUSION AND FUTURE WORK

Estimations are usually made to support for planning and tracking project. The model-based, monitoring based and expert-based estimation approaches are popular. The estimation models regularly use the current project data rather than the historical project data to improve the accuracy of estimation over iteration/releases. Most of the paper describes effort estimation with story point approach and use people, project related and critical factors to calculate the effort of a project. This paper identifies the functional specification of the new project which is map into the Historical project. These functional specifications are used to estimate the efforts needed for the project. This paper shows, how efforts are distributed to

various use cases and the estimation of effort summary. More than 500 hours are differed between with historical data and without historical data. Future studies could do more on: 1) Size of the project is calculated from the efforts 2) Time estimation and cost estimation with the help of historical data.

REFERENCES

[1] Valentina Lenarduzzi, Davide Taibi**,**Can Functional Size Measures Improve Effort Estimation in SCRUM**?** October 2014.

[2] David Consulting Group, September 2014, "What other types of function points exist other than IFPUG function points And what are the pros and cons of different function type.

[3] Rashmipopli and Naresh Chauhan, 2014,IEEE "Cost and effort estimation in Agile software development",

[4] IshrarHussian, Leila Kosseim, olgaormadijeva , .May 2013,"Approximation of Cosmic Functional size to support early effort estimation in Agile".

[5] Florian Raith and Gudrun Klinker, IEEE 2013,"identification of inaccurate effort estimates in agile Software development.

[6] Zhamri Che Ani, ShuibBasri, Malaysia 2013. "A Case Study of Effort Estimation in Agile Software Development using UseCase Points" Special Issue: Agile symposium,

[7] Daugh Nguyen-long and De Tran-Cao, IEEE 2013, "A review of Effort Estimation studies in Agile, iterative and Incremental Development",

[8] Taghi et all, IJCSI, July 2012, "On the Current Measurement Practics in Agile software Development".

[9] Tsun Chow,Dac-Buu Cao, Science Direct 2007, "A Survey study of Critical Success factors in agile software projects".

[10] Cockburn, 2007, "Agile Software Development," Pearson Education, Asia Low Price Edition.

[11] Mike Cohn, ,2005 **"**Agile Estimating and Planning", Addison-Wesley