

COST AND EFFORT ESTIMATION FOR AGILE SOFTWARE DEVELOPMENT



Lindamaswana binti Masbahon adalah Pegawai Teknologi Maklumat, Agensi Pengurusan Bencana Negara (NADMA) di Jabatan Perdana Menteri. Beliau telah memulakan perkhidmatan pada tahun 2008 dan telah berkhidmat di pelbagai jabatan seperti Jabatan Kastam DiRaja Malaysia, Jabatan Kehakiman Syariah Malaysia dan terkini di NADMA, JPM. Beliau berkelulusan Ijazah Sarjana Muda Sains Komputer (Kejuruteraan Perisian), Universiti Putra Malaysia (UPM) dan pemegang Ijazah Sarjana Sains Komputer (Kejuruteraan Perisian), UPM.



Jamilah binti Din (Dr.) merupakan Pensyarah Kanan, Jabatan Kejuruteraan Perisian dan Sistem Maklumat, Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Putra Malaysia (UPM). Beliau berkelulusan Ijazah Sarjana Muda Sains Komputer, Indiana University, USA. Beliau juga pemegang Ijazah Sarjana Sains Kmputer, UPM dan serta Doktor Falsafah (Sains Komputer), Universiti Kebangsaan Malaysia.

ABSTRACT

Abstract—Kebanyakan punca biasa bagi pembangunan perisian gagal adalah kerana belanjawan yang tidak munasabah, kekangan masa, tiada kepakaran dalam bidang berkaitan dan sebagainya. Banyak syarikat perisian mencadangkan kaedah *agile* untuk digunakan di dalam pembangunan sesebuah sistem di dalam sektor perkhidmatan awam yang sedia maklum mempunyai bajet yang ketat dan perlu disiapkan dalam jangka masa pendek. Oleh yang demikian, anggaran kos dan usaha di peringkat awal menjadi peranan utama untuk memastikan bahawa pembangunan perisian berjaya dilaksanakan. Oleh itu, tujuan kajian ini diadakan adalah untuk mengatasi masalah tersebut dengan menggunakan sistem prototaip untuk membuat anggaran kos dan usaha di peringkat awal projek mereka secara sistematik. Untuk mencapai objektif kajian ini, tiga (3) langkah bermula dengan kajian teori. Dalam kajian teori, Fungsi Analisis Point (FPA) didapati kaedah yang paling sesuai untuk mengira kos dan usaha di peringkat awal pembangunan perisian. Langkah kedua adalah dengan membangunkan sistem prototaip selepas kriteria untuk mengira kos dan usaha ditentukan. Langkah terakhir adalah ujian ke atas sistem menggunakan dataset yang mempunyai kos sebenar. Hasil kajian ini menunjukkan kos menggunakan FPA mempunyai sedikit persamaan dengan keputusan sebenar manakala keperluan usaha untuk membangunkan satu perisian yang dikira adalah jauh lebih rendah daripada usaha sebenar untuk membangunkan projek tersebut.

Keywords—*cost & effort estimation, agile, FPA*

I. INTRODUCTION

The outgrowth of Information Technology (IT) in this twenty-first century has affected many sectors involved in world development which benefited them by accelerating the business purpose and increase the profit gain (Petersen, K., & Wohlin, C., 2010). The evolution of software development in the past decade has witnessed a rapid growth in most of the modern country that utilize most of their process computerize. In order to be one of the modern countries, the infrastructure used must be fully computerized for either citizen to easily access it or foreigner to utilize it. However,

to be one of them, it needs a huge numbers of investment. For example, by using computerized system, it will expedite the process of getting national identification, doing business transaction among countries or identifying process of getting in and out from country for the tourist. Moreover, the user itself will easily track down the record of every transaction from the system. With the use of IT, people around the world is globally connected with each other which also help in spreading knowledge without boundary.

Based on the observation, we can easily say that software development increase tremendously. Implementation of the software also changes from waterfall method into agile method where the users evolve in most of the development process (Lang, M., Conboy, K., & Keaveney, S., 2013). Even under the agile method umbrellas, there are few ways to implement agile in software development like Scrum, XP (Extreme Programming), DSDM, Crystal and it has been researched over decades about estimating cost and effort in agile manifesto. Agile methodologies have been found out that it bring a great values in software development, improving quality of the system and shortening the time to market the software (Hadar, I., & Sherman, S., 2012). There are a few case studies have been done by previous researchers that has been proven the good impact by using agile method in their software development. Laanti et al, (2011) concluded from their empirical study at well-known company in 2011, Nokia where most of the respondents agree with the usefulness of agile and they do not want turn back using old method they worked before. Meanwhile Seo et al., (2013) have done empirical study at one of the banking institution in Korea and conclude that to achieve the effectiveness in software project management, it necessary to predict the accuracy of software estimation. Dingsøy, Torgeir, et al., (2012) summarized countless implication of agile methodologies. They propose to dig more about effectiveness of agile methods and practices especially in different organizational context. Among the advantages of using agile methods in developing software is the regular interaction between developer and user, which lead to easy understanding of the requirements between both parties (Paasivaara, M., & Lassenius, C., 2006). The customer can get rapid satisfaction because the working software is delivered frequently to them. In agile method, customer, developer and tester constantly interact with each other in order to make sure the completeness of the requirements needed.

An accurate estimation for cost and effort estimation in developing software becomes critical issues in most of software houses. Inaccurate estimation may give a negative impact to the software houses whereby the system build may not function properly and end up with a bad quality system. This impact will not only affect the software house but also the developer's reputation and competitiveness (Popli et al, 2014). This situation also applies to the government agencies in Malaysia, as it is a main factor for them to estimate cost and effort accurately when analyzing and awarding tender to the suitable software houses that meets with the government requirement itself. It is still controversial to the government agencies since many of the IT project cannot meet the dateline when the agile software development become highly in demand and chosen by many software houses (Ayed, H., Vanderose, B., & Habra, N., 2014). This lead to skepticism in Government's IT sector that structure and culture of agile method and processes does not suit government project in order to acquire a computerize and systematic government business process. These issues become a critical question especially to the Government where it has been well known with their tight budget and short timeline to provide a better facility to the nations.

A. Problem statement:

Every year, the citizens will be hearing about government's budget allocation to each ministry in Malaysia. The budget enhancement that needs to be done in the ministry for that year and IT budget is one of the budgets included in the allocation. In order to smooth the government process when giving services to their nations, we need to computerize the process of working flow especially process that calls for many transactions per day like hospital, international business,

transportation and etc. Meanwhile when we are trying to convert from manual procedure into computerize system or to convert from legacy system due to their age and unable to quickly enhance to meet the business changes, there will be many problem and the most we will hear is about running out of budget, failure to deliver the system within the timeline and the worst case is the system fail to deliver for customer usage (Popli et al, 2014). Laird and Brennan (2006) ascertain that 23% of all software projects are cancelled before completion and from the completed projects, only 28% system delivered on time with the average project over-running the budget with 45%. In this case, which party will be responsible when the system fail to deliver? The government may already be investing half of the budget and same goes with the software houses. So, here is the question, does government make mistake when estimate cost and effort in software development using agile methodologies in tendering IT project? (Petersen et al, 2009) and (Popli et al, 2014). Based on survey by Peixoto, C. E. L., Audy, J. L. N., & Prikladnicki, R., (2010) about software estimation usage, the project has better results in terms of accuracy and the correctness when use the right technique in estimating cost and effort to avoid issues by poor estimation.

Hence, 2 research questions have been generated in this study:

RQ 1: What are the suitable methods and criteria to calculate cost and effort estimation in agile software development for government?

RQ 2: Does the government or vendor use any tools in other to estimate their cost and effort before submitting the tender?

For research question no.1, it needs to study from past research about methodology to estimate cost and effort estimation in developing software and this study will be specifically focusing on agile software development.

To answer second research question, this study will develop a system prototype to calculate cost and effort estimation based on criteria that has been chosen in earlier study. Then, the output will be compare with the actual cost to check the validity of the formula to calculate cost and effort estimation.

B. Objective of this study

In view of numerous study, this study propose a system prototype that may help in estimating cost and effort in developing software based on function need by user instead of based on the previous software houses historical such as data lines of code. The method that will be used in the system is called Function Point Analysis (FPA). FPA method has been chosen because we want to open the government tender not only for those company that have previous history only can submitted their application but also to those the new company that never involve in developing government system. This study will let judging process become fair to all software houses which will lead to competitiveness between them in order to show their ability and responsibility. So, here are the objectives of this propose study:

- To identify relevant criteria in estimating cost and effort for software development using agile method
- To develop a system prototype in estimating cost and effort for software development using identified criteria.
- To evaluate the identified criteria by comparing with actual cost.

C. Expected Result

The expected result of this study is to produce a system prototype that can estimate cost and effort for developing software for government more accurately based on the chosen method. Through the usage of the prototype, the users will have accurate evidence for every estimation cost and effort that they proposed later on.

D. Paper Organization

The rest of this paper is organized as follows: Section II present a review of literature review and past researcher works that associated with this study problem addressed, a better method to estimate cost and effort starting from early phase software development. Section III explain research methodology used in this study which consists of understanding the domain problem and process development to generate the result. Section IV describe about designing of the system prototype that will be as tool to answer the problem addressed earlier and its implementation using the real dataset. While Section V present evaluation of this study. This stage will evaluate the proposed system prototype using the real dataset to show whether the method chosen really help to get better result than the actual cost and effort. Last section is the conclusion of this study where it discuss lesson that being learnt in this study and recommendations for future works.

II. RELATED WORKS

This section provides related literature reviews that have been identified during this study that provide respectable understanding towards research topic. In this chapter, appropriate literature related to estimation cost and efforts for agile software development are presented. This review is organized in three (3) sections: (A) agile life cycle, (B) various types of cost and effort for agile software development and (C) function point analysis method.

A. Agile Life Cycle

Agile methodology is an iterative approach for software development that is performed in a collaborative environment by self-organizing teams. It propose an alternative from traditional approach which is waterfall where every aspect of development in agile is continually revisited. This methodology produces high-quality software in a cost-effective and timely manner to meet stakeholders' changing needs. Agile approaches are typically used in software development to help businesses respond to unpredictability. Iterations in agile are in a short time based and normally the duration among one to four weeks for one iteration to complete (Cockburn, 2007). Extreme Programming (XP), Scrum, Crystal are among of the main method that fall under agile umbrellas, sharing same philosophy but each of them have their own practices and terminology in implementing agile methods.

Table 1 describes by Boehm et al, (2003) the differences between plan-driven and agile methodologies in some parameter for better understanding.

Table 1: The Differences Between Plan-Driven And Agile Methodology

Parameter	Plan-driven	Agile
Process Model	Waterfall	Iterative/incremental
Environment	<ul style="list-style-type: none"> • Stable • Low change • Project and organization focused 	<ul style="list-style-type: none"> • Turbulent • High change • Project focused
Project Planning	<ul style="list-style-type: none"> • Detailed project from start to end created in early of the project 	<ul style="list-style-type: none"> • Coarse grained plan for overall project, detailed plans per iteration
Requirement Engineering	<ul style="list-style-type: none"> • Comprehensive requirement document which often part of contract 	<ul style="list-style-type: none"> • Easy access to customer rather than reliance on comprehensive requirements documentation
Testing	<ul style="list-style-type: none"> • Testing phase at the end of the project and execute by test specialist 	<ul style="list-style-type: none"> • Testing spread out over entire project and execute by end users

From previous study, there are a lot of methods that can be used in estimating cost and effort software development. Cost estimation is referred as cost of software development process workload. This cost does not include costing of material which means it differ with other regular industrial business while effort estimation is referred as cost of human labor consumption which directly involve in developing the system.

Patil, Lalit V., et al., (2014) summarized that cost and effort estimations are divided into algorithmic and non-algorithmic method.

a. Algorithmic Method

Algorithmic method use mathematical formula to compute anticipated cost of project where the parameters are based on industrial experience COCOMO II model [17], Source Line of Code (SLOC) by Khatibi, V., & Jawawi, D. N., (2011) and Function Point by Albrecht, A. J., & Gaffney Jr, J. E., (1983) are some of the popular algorithmic method that use by previous researcher.

b. Non- algorithmic method

A non-algorithmic approach is based on expert method, machine learning and soft computing such as analogy, expert judgment, neural network and fuzzy logic. The analysis in this method based on chronological data and follow by human behaviour which based on information from previous projects of that software house (Attarzadeh, I., & Ow, S. H., 2010).

There are numerous advantages using algorithmic method to estimate cost and effort in software development such as it is easy to modify input data, refine and customize formulas when needed. This method calibrated to previous experience of the software houses. It gives advantages to those who have experience and inspiring to be competitive for new software houses in getting the tender. Besides that, algorithmic method able to generate repeatable estimation in another word, it is able to support a family of estimations technique (Sharma, N., Bajpai, A., & Litoriya, M. R., 2012).

Therefore, based on those advantages, this study will be focusing on numerous algorithmic models instead of non-algorithmic that available in estimating cost and effort in developing software using agile methods.

COCOMO (Constructive Cost Model) developed by Boehms, are well known over the world for method in estimation software. This method consists of three types of hierarchy which are:

a. Basic COCOMO

This phase is good for roughly and quick in estimating cost and effort based on source line of code (SLOC) but its accuracy is limited due to its lack of factors to account for difference in project attributes (*Cost Drivers*).

b. Intermediate COCOMO

This phase computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes.

c. Detail COCOMO

Detailed COCOMO incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

This model has been interpreted into a system namely as COCOMO 81. This system is developed on 1981 according to the software development practices on that era like C, Fortran or COBOL which also called as 3GL language. Then when the software development technique change dramatically, COCOMO II developed to meet the 4GL language (Visual Basic, Java, PHP etc) needed by offering increased the accuracy further along in the project planning and design process.

Another popular model in algorithmic method is Putnam Model by Lawrence H. Putnam published in 1978. Putnam model describes the time and effort required to finish a software project in a specified size. Putnam model is very sensitive to the development time: decreasing the development time can greatly increase the person-months needed for development (Jones, T. C., 2007) because it is based on knowing, or being able to estimate accurately the size (in lines of code) of the software to be developed. This model has been interpreted into proprietary software called SLIM (Software Lifecycle Management).

Function Point Analysis (FPA) (Ifpug, Function Points Counting Practices Manual (version 4.2.1), (2004) is another software measurement technique that is widely known and utilized created by Allan Albrecht at IBM in 1979. FP considers five basic parameters to measure a software system: External Input (EI), External Output (EO), External Inquiry (EQ), Internal Logic File (ILF), and External Interface File (EIF). FP is a little bit differ with other algorithmic model mention above where it only count based on function and not depends on lines of code.

This study will choose not to calculate cost and effort estimation using SLOC because it is difficult to estimate the SLOC in the early stage of software development lifecycle. Therefore, if the SLOC is inaccurate, the output of the cost and effort estimation will be inaccurate. In addition, project tendering in government will only list out the requirements for their new system. Based on experience, there was no guarantee that if the company have experienced developing government's system able to finish another system on time. Another reason is SLOC will differ by programming language that they used and skill expertise of the developer itself but by using FPA, it is free language meaning here by using FPA, the calculation in estimating cost and effort not relying on SLOC.

Table 2 below show the characteristics involve in each method to calculate cost and effort estimation in developing system.

Table 2: Criteria To Calculate Cost And Effort Estimation In Software Development

No.	Criteria	Model				
		Cocomo II	Putnam	Use Case Point	Function Point	Object point
1.	KLOC	✓	✓			
2.	Effort Applied (E)	✓	✓			
3.	Development Time (D)	✓			✓	✓
4.	People required (P)	✓				
5.	Time		✓			
6.	Productivity		✓			
7.	Size		✓			
8.	Quantity Of Use Case			✓		
9.	Use Case Type			✓		
10.	Unadjusted Function Point (UFP)				✓	✓
11.	Adjusted Function Point (Afp)				✓	
12.	Unadjusted Object Point					✓
13.	Adjusted Object Point					✓

This study is recommending delving more in calculative instead of predictive because of the easy understanding on what to be calculate based on requirement given. Function point has been used in a few estimation cost and effort of agile methodologies. There are some cost estimation methods which are based on a function point type of measurement, such as ESTIMACS and SPQR/20. SPQR/20 is based on a modified function point method. Whereas traditional function point analysis is based on evaluating 14 factors, SPQR/20 separates complexity into three categories: complexity of algorithms, complexity of code, and complexity of data structures. ESTIMACS is a propriety system designed to give development cost estimate at the conception stage of a project and it contains a module which estimates function point as a primary input for estimating cost.

2.1.1 Why choose function point in estimating cost and effort in agile software development?

Popli et al, (2014) described problem that may causes inaccurate estimating software and proposed algorithm to calculate the estimation. The algorithm is based on FP whereas in case there have not any historical data that might be related to refer. The researchers also prioritize the user stories based on the important client needs in a way to speed up the development process (Popli et al, 2014).

The other case study by Pow-Sang, J. A., & Imbert, R., (2012) present an approach called Incremental-FP that allows estimating effort for each increment using unadjusted function points (UFP) and COCOMO's effort adjustment factor (EAF). The experiment has been done towards undergraduate student at Pontificia Universidad Catolica del Peru where the researcher obtained the difference between estimated effort and real effort was less than 20% on the second increment. Hastings et. Al, (2001) indicates that a technique may be considered predictive and may be used with confidence, when the magnitude of relative error (MRE) is less than or equal to 20%.

The main cause FPA chosen to use in this study is because it can calculate the estimation from functional requirement analysis or any document related with specification. This method is the most suitable since government tender will only list out requirement that they need whereby the

software house need to develop system based on that requirement. In consequently, the tender only will be awarded to the success software house after getting approval from top management which most of them are not technical person. Besides that, FPA is more on technology independent whereas it not relying on lines of codes that generated by programmer. So, FPA is the best method since it is easy to understand by non-technical person. FPA also has been recognized by international body as per listed below:

- a. IFPUG v.4.1, ISO/IEC 20926:2003 [27]
- b. MK II v.1.3.1, ISO/IEC 20968:2002 [28]
- c. NESMA v.2.1, ISO/IEC 24570:2005 [29]
- d. FiSMA FSM v.1.1, ISO/IEC 29881:2008 [30]
- e. COSMIC v.3.0, ISO/IEC 19761:2011 [31]

Table 3 below shows the comparison between the three most popular used in previous researches.

Table 3: Comparison Model Of Cost And Effort Estimation

No.	Criteria	Model				
		Cocomo II	Putnam	Use Case Point	Function Point	Object point
1.	KLOC	✓	✓			
2.	Effort Applied (E)	✓	✓			
3.	Development Time (D)	✓			✓	✓
4.	People Required (P)	✓				
5.	Time		✓			
6.	Productivity		✓			
7.	Size		✓			
8.	Quantity Of Use Case			✓		
9.	Use Case Type			✓		
10.	Unadjusted Function Point (UFP)				✓	✓
11.	Adjusted Function Point (AFP)				✓	
12.	Unadjusted Object Point					✓
13.	Adjusted Object Point					✓

Methodology

This chapter discussed about methodology used in the research. There are 3 phases involved in this study which are (A) theoretical study involved, (B) system prototype development and (C) evaluation of the system prototype.

A. Theoretical Study

Firstly, theoretical study needs to be done in order to identify the relevant criteria that will be used to estimate cost and effort in developing the software. This phase has been done at the chapter 2 where FPA have been chosen to use in estimate cost and effort in this project. Five criteria of the FPA will be analyze and calculate in this study which are external input (EI), External Output (EO), External Inquiry (EQ), External Interface File (EIF) and Internal Logical Files (ILF). The guidelines to categorize the requirements documents for the project will be based on IFPUG manual counting procedure version 4.1. This study also will apply what FPA propose where it cannot calculate based on one method only. So, this study also will evaluate the value adjustment factor

(VAF) which is a 14 survey questions that will be answered by team member of the project that being as a dataset in this study.

B. Prototype Development

Secondly, the system prototype will be developing based on criteria in the method that have been chosen earlier. The system design developed is based on criteria and the user perception. This study focusing on agile methodology because of it produces high-quality software in a cost-effective and timely manner to meet users' changing needs. In this study, we developed the software using Microsoft product where the programming language used is Microsoft C# with framework .net and the database used is Microsoft SQL. Both of the software is licensed under student program. This language have been choose because it popularity in agile software development. At the same time, dataset is being prepared from system requirement specification (SRS) documents provided by selected government agencies that allow their documents to be used for this project. FPA is a combination of calculation and expert opinion. The expert opinion come out with 14 survey questions that given to the involved team.

C. Evaluation

Lastly, evaluation process will be done in this study where the dataset prepared earlier will be running using the proposed prototyped. Then the result of cost and effort calculated by the proposed system will be compared with the actual cost. The result need to be compared in order to show that the FPA is one of the better ways to estimate cost and effort in earlier requirements gathered especially where there is no historical calculation of estimation have been done on previous project. It also to prove that FPA is better than other method which need no of lines of code (SLOC) in estimating cost and effort to develop a system because SLOC may varies by programming language and it is a little bit late to estimate after get total SLOC involve.

DESIGN AND IMPLEMENTATION

This chapter describes the technical aspects involved in the proposed system. In order to achieve the objectives for this study, a system prototype namely as Function Point Calculator has been developed as the solution. This system prototype will act as a platform to enable the experiment being done to complete the second objective of this study.

System Design Operation

The architecture of the Function Point Calculator is using client server approach. A client computer, like a personal computer (PC) is the user tool for interfacing with the system and some or all application processing. A server is often the source for the application of client-users are executing to do work. In this prototype, the pc acted as a server. This approach use server to host delivers and manages most of the resources and services to be consumed by the client. The database layer is build using Microsoft SQL that provides the storage capability to the system. The database contains data that is used to serve the function of the system. The structure of database table is constructed so that it is easier to map the data. The key used for mapping from one table will become a foreign key in the other table. This is to ensure that the mapping of data will return correct output for analysis process to produce the result.

A. UML Diagram

i. Use Case Diagram

A use case diagram shows how system use cases are related with each other and how the user connects with the use cases. Each bubble represents the process that needs to be done consecutively by the user and the stick person represent user that involve in this system. The use case diagram in Figure 1 below shows the relationship among actor and use cases in the system. Currently, the system have two users which are admin that acting as administrator to control the user and another one is user of the system that want to do estimation of cost and effort. For example, the user need to key in all the five main categorization from his SRS documents and VAF value, and then the system will show the result to the user which is total cost and effort need to finish his project.

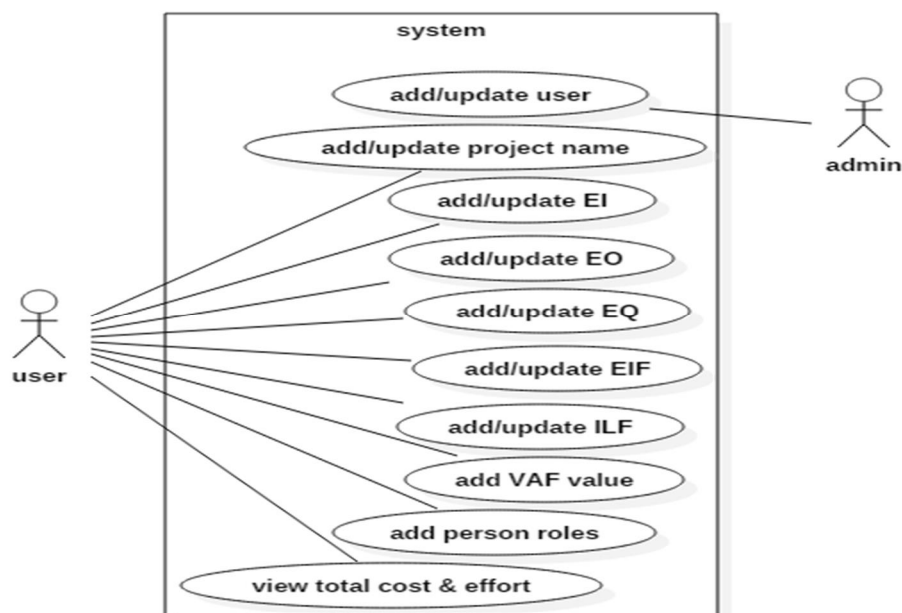


Figure 1: Use Case Diagram

ii. Class Diagram

The class diagram in figure 2 below shows the initial database structure of the Function Point Calculator prototype. There are 8 tables in the structure. The Project table handle all the tables in the database where the user need to add new project before begin key in the data. If there are existing project, user can search which project they need to update or view it. One project can have many function points. The Function Point table has another two tables related with it which are function point type and complexity type. One function point can only have one function point type which is external input (EI), external output (EO), external inquiry (EQ), external interface file (EIF) and internal logical file (ILF) while the complexity type is based on what type of function point choose by that user. The system will automatically calculate UFP of the project once user key in the five characters of FPA. One project can have only one VAF value to be key in by user.

To get value of adjusted function point (AFP), user needs to enter value VAF that answered by project team member. The system will show total effort after get AFP. From the diagram, one project can have many effort estimation if the user want to view effort involve if they choose difference language. While the total cost project will calculate after user entering person involve in that project with detail of their roles, man day price and estimation effort involve in that project.

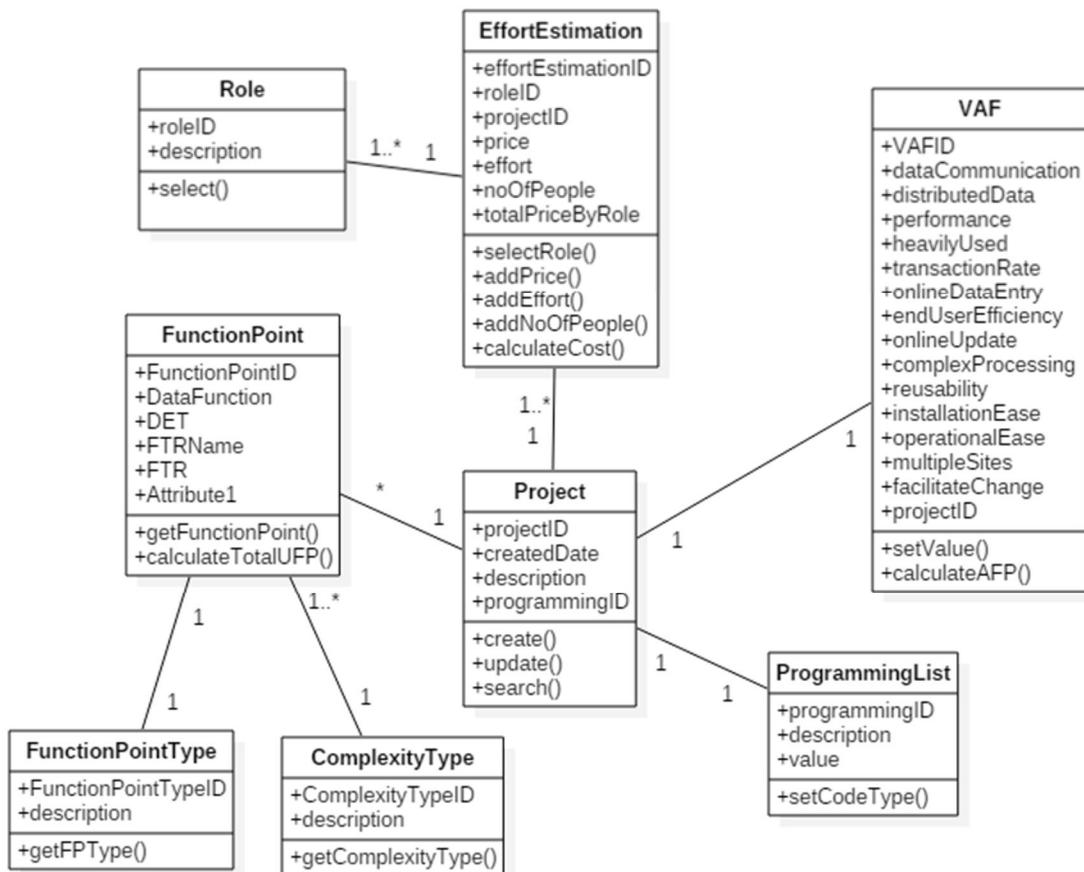


Figure 2: Class Diagram

iii. System process flow

The proposed system allow user to key in the five criteria need in FPA and value of the VAF. Then the user need to choose type of programming language that they use to develop the system to get total effort days need to finish the project. After that user need to key in team members that may involve in developing that particular system and this prototype will calculate estimation cost for the user. Figure 3 shown below is the process flow of the system prototype that has been developed.

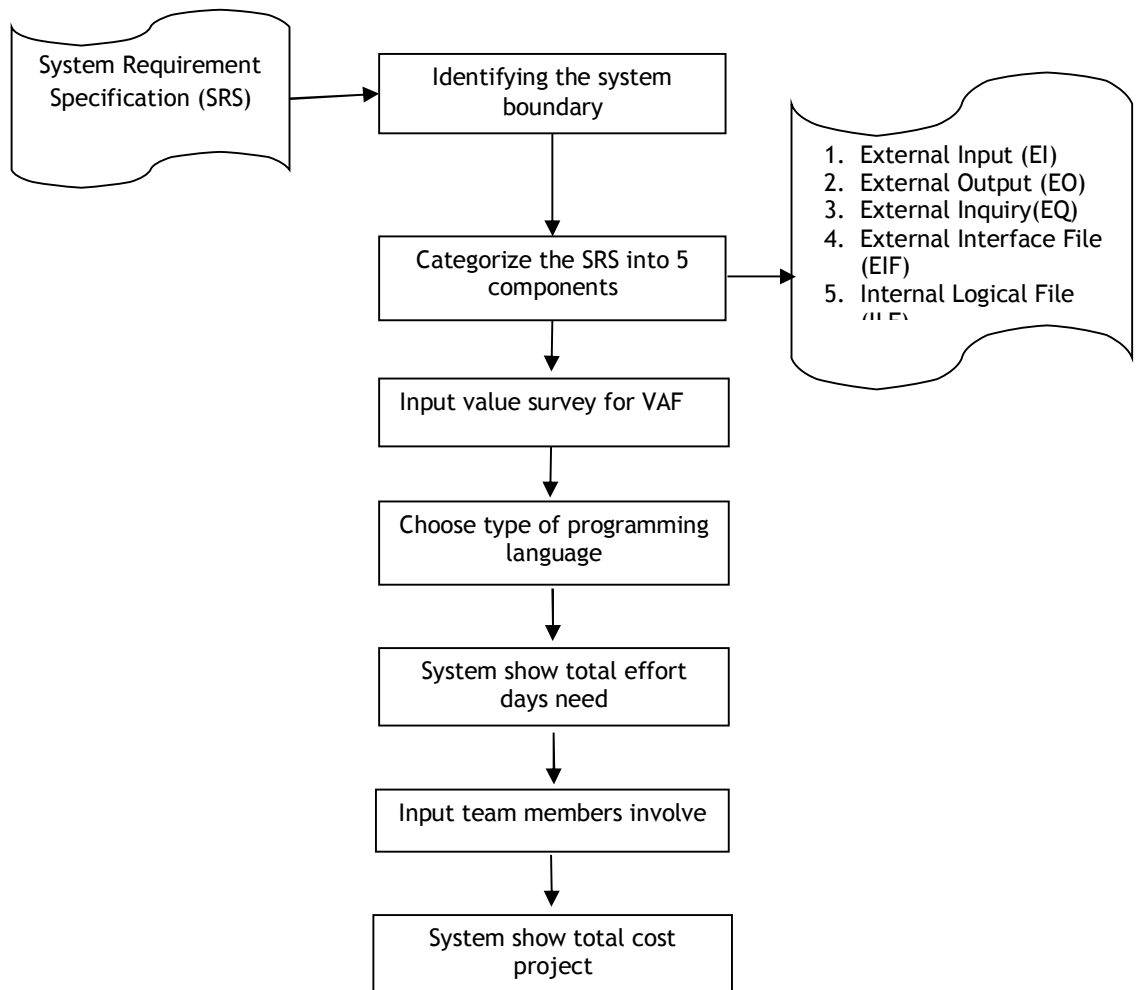


Figure 3: Process flow in Function Point Calculator

B. Implementation

First of all, the prototype need user to key in name of the project if the project is the new one before begin key in the data. The system will show pop-up message if user success creating new project. If the project already exists, the user can search the project name and update it. Then, the user starts to key in the data that have been categorized into five categories which are EI, EO, EQ, EIF and ILF. Based on Counting Process Manual (CPM), (2003), external input (EI) is an elementary process that processes data or control information that comes from outside of the system boundary. The primary intent of an EI is to maintain one or more internal logical files (ILF) and/or to alter the behavior of the system. This prototype need user to identify data element type (DET) which is data entry that might be get from user or external input and may change the database which called as file type referenced (FTR). The combination of the DET and FTR will decide the complexity of the EI. The complexity table taken from FPA guideline as table 4 below.

Table 4: External Input (EI) Complexity

FTR	Data Elements		
	1-4	5-15	>15
0-1	Low	Low	Medium
2	Low	Medium	Complex
>3	Medium	Complex	Complex

EO is an elementary process that sends data or control information outside the system boundary. The primary intent of an EO is to present information to a user through processing logic. An EO also maintain one or more ILF and/or to alter the behavior of the system (27). The combination of the DET and FTR will decide the complexity of the EO. EQ is an elementary process that sends data or control information outside the application boundary. The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information from an ILF of external input file (EIF). The processing logic contains no mathematical formulas or calculations, and creates no derived data. No ILF is maintained during the processing, nor is the behavior of the system altered. (27). For EQ, the user need to identify DET which is the data that user request to query the database (FTR). The combination of the DET and FTR will decide the complexity of the EQ. The complexity table for EQ is same with the EO based on FPA guideline as table 5 below.

Table 5: EO and EQ Complexity Table

FTR	Data Elements		
	1-5	6-19	>19
0-1	Low	Low	Medium
2-3	Low	Medium	Complex
>3	Medium	Complex	Complex

ILF is to hold data maintained through one or more elementary processes of the application being counted (Counting Process Manual (CPM), 2003). In other words, the ILF is data that stored in database where it allow user to store and maintain within the boundary of the system. FTR in ILF is record that acting as child to the parent (parent-child relationship). While EIF is to hold data referenced through one or more elementary processes within the boundary of the application counted. This means an EIF counted for an application must be in an ILF in another application (Counting Process Manual (CPM), 2003). In other words, the EIF is a source data from outside of the system that user will use as referenced but the user cannot maintain the data. The user to identify DET which is the data function that will use in the system with related file type referenced (FTR). The complexity table for ILF refers to table 6.

Table 6: ILF and EIF Complexity Table

RET	Data Elements		
	1-19	20-50	>50
0-1	Low	Low	Medium
2-5	Low	Medium	Complex
>5	Medium	Complex	Complex

After user key in all the value for 5 characteristics of the FPA, the system will calculate unadjusted function point (UFP) by add all the total of five criteria that has been input earlier. The next step is user need to key in VAF that have been answered by the team member that involve in developing the project. The answer of the VAF questionnaires designed using Likert scale to measure team members respond towards the system that will be develop by them (refer Table 7).

Table 7: Likert Scale Type

0	No influence
1	Insignificance influence
2	Moderate influence
3	Average influence
4	Significant influence
5	Strongly influence

Then, the VAF value will add on to the UFP to get value of adjusted function point (AFP). AFP is the total function points for the particular project that needs to be develop by the team member (refer Figure 4). The VAF calculation formula is like below:

$$\text{VAF} = 0.65 + 0.01(\sum \text{Dol}) \text{ where Dol} = \text{degree of influence for each factor}$$

Figure 4: VAF interface

In order to know the total effort days need to develop the software, user need to choose type of programming language that will be use later to develop the system while for the total cost, it needs user to input every person that involve in developing the software with their specific roles and man day salary to develop that project. The effort for each person roles is based on expert judgments because there was no specific guideline or past research elaborate effort based on roles of team members. While the rate effort function point per day is taken from ISBSG guideline produced on 2009 since there was no historical study collected at that agencies.

IV. RESULT AND DISCUSSION

This section will discuss the result from evaluation process using the proposed prototype whether it meets the objective of this study. Based on the objectives of this study, the proposed system must able to (1) identify relevant criteria in estimating cost and effort for software development using agile method, (2) develop a prototype system in estimating cost and effort for software development using identified criteria and (3) evaluate the identified criteria by comparing with the actual result.

Evaluation of the test data

The system prototype have been running using dataset that being created using SRS that we received from one of the government agencies in Malaysia. The SRS has been finish write up on 2012. The system generally is about registering the government's client and process of the approval of the client project will store in the system. So, the first stage to evaluate the system prototype is to key in all the input data need which are the five characteristics of FPA called it as EI, EO, EQ, EIF and ILF in the system to get UFP value. The system will calculate total effort need to develop the system. Here, the system prototype calculates effort needs per function point based on ISBSG outlined on 2009 as per below;

Table 8: Productivity Hour Per FP

Programming Language	Hour/FP
ASP	06.1
VB	8.5
Java	10.6
C++	12.4
C	13
C#	15.5
COBOL	16.8

$\text{Effort} = \text{AFP} / (\text{programming language})$ <p>Where programming language = FP/day</p>

From the table given, we manipulate the calculation into FP per day that can be finish by programming language. Working day in Malaysia is normally eight hour per day. Since the AFP value calculates into two decimal points, the effort also will take the same decimal point and the result become days in decimal point.

Table 9: The Differences Cost And Effort Between FPA And Actual

	FPA	Actual Cost
Cost	RM 90350	RM 95466
Effort	394.8 days	132 days

Table 9 shows the result that generated from the Function Point Calculator. Based on number of person involved in the actual project which is seven people, we use it to calculate cost of the project using FPA method. The result show that the cost calculated using FPA method is slightly same with the actual cost. The man-day salary calculated based on statistic salary in Malaysia IT industry but the effort per role is based on expert judgment. However the total effort is higher than the actual effort. The effort calculated by FPA is counted for one person only. So, in order to get the actual effort that may be need for one team to finish the project, we need to divide total days into total person involved in that project. For example, this project involves seven (7) people in a one team to finish the project. So, if 394.8 days divide by seven will get 56.4 days effort per person. It means here, the project can be finish earlier from 120 days given in the tender.

CONCLUSION

Overall, purpose of this study is to introduce a platform or medium that can support project manager in any agencies to estimate their project cost and effort accurately. In order to fulfill the objectives of this study, a system prototype called Function Point Calculator has been developed. The system prototype will assists any project manager in better estimation of their cost and effort that may involve in early of the development of the system. The significance impact of this proposed study will help any IT project manager especially in government agencies to do a better estimating cost and effort of every tender that they receive in order to evaluate their proposal to meet the requirement. Instead of that, the judgment of tender will be done fairly and equitable

which end up the result tender without prejudice. At the same time, it will help both parties which are government and software house getting their general idea when doing the proposal for IT tender which they can get their return of investment after release the system. The development of the prototyped has successfully grab the user's attention and finally the test has proven that implementation of this estimation in this study has successfully accepted by the user.

A. Future Works

At present, this study is just providing the basic part which laid an important foundation for future development. There are still much room for system improvement and perfection to perfectly achieved its objective and provides powerful system information which can help the agency to better estimate their cost and effort to develop any IT project. The missing parts that might be do after this are:

- The system may be extend to read from the document itself without human intervention (ex.: read from use case diagram/entity-relationship diagram)
- The sample data may compare with another type of function point analysis like NESMA, Mark-II, FiSMA and COSMIC.

REFERENCES

- Albrecht, A. J., & Gaffney Jr, J. E. (1983). Software function, source lines of code, and development effort prediction: a software science validation. *Software Engineering, IEEE Transactions on*, (6), 639-648.
- Attarzadeh, I., & Ow, S. H. (2010). Improving the accuracy of software cost estimation model based on a new fuzzy logic model. *World applied sciences Journal*, 8(2), 177-184.
- Ayed, H., Vanderose, B., & Habra, N. (2014, June). Supported approach for agile methods adaptation: An adoption study. In *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering* (pp. 36-41). ACM.
- Boehm B. W. (Jan.1984) "Software Engineering Economics", IEEE Trans Software Engg., Vol.SE-10, No.1, pp.4-21.
- Boehm, B., & Turner, R. (2003). Using risk to balance agile and plan-driven methods. *Computer*, (6), 57-66.
- Cockburn, 2007 "Agile Software Development," Pearson Education, Asia Low Price Edition.
- Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.
- Laird, L. & Brennan, C. (2006). Software measurement and estimation: A practical approach. New York: IEEE Computer Society/Wiley.
- Lang, M., Conboy, K., & Keaveney, S. (2013). Cost Estimation in Agile Software Development Projects. In *Information Systems Development* (pp. 689-706). Springer New York.
- Hadar, I., & Sherman, S. (2012, June). Agile vs. plan-driven perceptions of software architecture. In *Cooperative and Human Aspects of Software Engineering (CHASE), 2012 5th International Workshop on* (pp. 50-55). IEEE.
- Hastings, T. E., & Sajeev, A. S. M. (2001). A vector-based approach to software size measurement and effort estimation. *Software Engineering, IEEE Transactions on*, 27(4), 337-350.
- Ifpug, Function Points Counting Practices Manual (version 4.2.1), (2004) Ifpug: International Function Point User Group, USA
- ISO, Software Engineering IFPUG 4.1 Unadjusted Functional Size Measurement Point User Group Method. Counting Practice Manual. 2003, ISO, Geneva.
- ISO, Software Engineering MK II Function Point Analysis. Counting Practice Manual 2002, ISO, Geneva.
- ISO, Software Engineering NESMA Functional size Measurement Method, version 2.1, Definition and counting guidelines for the application of Function Point Analysis 2005, ISO, Geneva.
- ISO, Software Engineering - Fisma Functional Size Measurement Method, version 1.1. 2008, ISO, Geneva
- ISO, Software Engineering COSMIC-Functional Size Measurement Method 2011, ISO, Geneva.
- Jaiswal, A., & Sharma, M. (2013). Expert Webest Tool: A Web Based Application, Estimate the Cost and Risk of Software Project Using Function Points. In *Advances in Computing and Information Technology* (pp. 77-86). Springer Berlin Heidelberg.
- Jones, T. C. (2007). *Estimating software costs*. McGraw-Hill, Inc.
- Khatibi, V., & Jawawi, D. N. (2011). Software Cost Estimation Methods: A Review 1.
- Laanti, M., Salo, O., & Abrahamsson, P. (2011). Agile methods rapidly replacing traditional methods at Nokia: A survey of opinions on agile transformation. *Information and Software Technology*, 53(3), 276-290.
- Ning, J., Yan, J., Yu, H. (2014) Research and application of estimation method for software cost estimation based on putnam model. [Journal of Chemical and Pharmaceutical Research](#). Volume 6, Issue 7, Pages 2617-2626

- Paasivaara, M., & Lassenius, C. (2006, October). Could global software development benefit from agile methods?. In *Global Software Engineering, 2006. ICGSE'06. International Conference on* (pp. 109-113). IEEE.
- Patil, L. V., Waghmode, R. M., Joshi, S. D., & Khanna, V. (2014, February). Generic model of software cost estimation: A hybrid approach. In *Advance Computing Conference (IACC), 2014 IEEE International* (pp. 1379-1384). IEEE.
- Peixoto, C. E. L., Audy, J. L. N., & Prikladnicki, R. (2010, May). The importance of the use of an estimation process. In *Proceedings of the 2010 ICSE Workshop on Software Development Governance* (pp. 13-17). ACM.
- Petersen, K., & Wohlin, C. (2009). A comparison of issues and advantages in agile and incremental development between state of the art and an industrial case. *Journal of systems and software*, 82(9), 1479-1490.
- Petersen, K., & Wohlin, C. (2010). The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empirical Software Engineering*, 15(6), 654-693.
- Popli, R., & Chauhan, N. (2014, February). Cost and effort estimation in agile software development. In *Optimization, Reliability, and Information Technology (ICROIT), 2014 International Conference on* (pp. 57-61). IEEE
- Popli, R., Chauhan, N., & Sharma, H. (2014, February). Prioritising user stories in agile environment. In *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on* (pp. 515-519). IEEE.
- Pow-Sang, J. A., & Imbert, R. (2012). Effort estimation in incremental software development projects using function points. In *Computer Applications for Software Engineering, Disaster Recovery, and Business Continuity* (pp. 458-465). Springer Berlin Heidelberg.
- Seo, Y. S., Bae, D. H., & Jeffery, R. (2013). AREION: Software effort estimation based on multiple regressions with adaptive recursive data partitioning. *Information and Software Technology*, 55(10), 1710-1725.
- Sharma, N., Bajpai, A., & Litoriya, M. R. (2012). A comparison of software cost estimation methods: A Survey. *The International Journal of Computer Science and Applications (TIJCSA)*, 1(3).