A NOVEL VARIANT OF FUNCTION POINT ANALYSIS FOR ACCURATE SOFTWARE SIZE ESTIMATION

Arpit Dhakad[*]

Anand Singh Rajawat[**]

ABSTRACT

Software cost estimation is the one of the most desired capability in software development. It helps customer to make investment but also helps to project manager to making appropriate plans for development. It is recognized that current estimation techniques does not cover estimation for expert user programming. We have proposed a new general system characteristic "Expert user Programming". Expert user programming also affects the size of a software. By adding it in the list of general System characteristics, we have created a provision for taking expert user facilities into account, while estimating the size of a project. After adding this General System Characteristic- Size, Time, effort will be increased. Hence to estimate these things, We have used this new GSC for expert user programming. It will help to software developers to create information systems which can be helpful for technically inexperienced users. This paper has include Expert User Programming as new General System Characteristic which will give accurate result of function Point Analysis.

Keywords- Function Point Analysis, General System Characteristic, Expert user programming

[*] MTech scholar, CSE (Software Engineering) Department, Shri Vaishanv Institute of Technology and Science Indore MP

[**] Assistant Professor, CSE Department, Shri Vaishanv Institute of Technology and Science Indore MP

I. Introduction

Software cost estimation is very necessary part in success of software development. In the beginning of computer era, Software cost estimation is very important task. when the software programs were less than 1000 machine instructions, only one programmer is require to write code. As software size grown, It is necessary to estimate accurate software cost. Software cost can be in terms of LOC (Line of code), effort, Schedule etc. Hence to calculate software cost, It is necessary to use efficient software cost estimation method. There are many methods available to estimate cost. But necessary thing is to estimate accurate software cost.

As we know, today software is the necessary part of modern business, government projects and Security (Military) operations. It means software company can produce hundreds of new software programs and modify existing applications every year. As a result, software cost estimation is a mainstream activity for every company. And as software users increased, software company provide a platform to develop software to non technical user and programming for non technical user is known as expert user programming. Expert user programming is done by end user who is not a professional developer. End user who wants to develop their own software for his personal use can develop software using online tools or small modification in existing code. Hence our system provide accurate software cost with expert user programming facility.

Most of the developer use Algorithmic and non algorithmic method to estimate software cost. Algorithmic model is based on mathematical functions or formula, results the accurate estimation. And Non algorithmic model is based on existing application. We predict the estimation by previously developed application which is same as application has to develop. Algorithmic models are COCOMO Model, Putnam's model, Function point analysis model etc and non algorithmic models are Analogy technique, Expert Judgment etc. Figure 1 illustrates the basic principles of modern

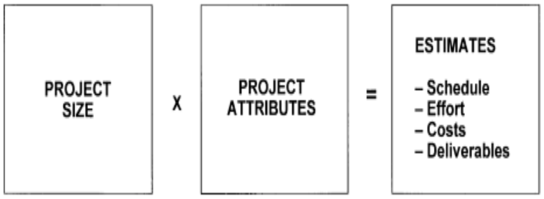

Figure 1: Software-estimating principles commercial software cost-estimating tools.

II. Related Work

1. P.K. Suri, Pallavi Ranjan, June 2012. Comparative Analysis of Software Effort Estimation Techniques. International Journal of Computer Applications.
2. Albrecht et al.,describes Function Point Analysis (FPA) method as alternative to code-based sizing methods.
3. Gaffney et al., illustrate international Function Point Users Group (IFPUG), a on-profit organization, which was later established to maintain and promote the practice.
4. IFPUG, describes extended and also published several versions of the FPA Counting Practices Manual to standardize the application of FPA.
5. Symons et al., describe other significant extensions to the FPA method have been introduced and widely applied in practice, such as Mark II FPA and COSMIC-FFP.
6. Abran et al., illustrate COSMIC-FFP which is also a extension to the FPA.
7. N. E. Fenton et al., proposed Function Point Analysis (FPA) model which consists of two main parts in the measurement. First is Data functionality and second is transactional functionality
8. Boehm et al., illustrate these characteristics which contribute to Value Adjusted Factor (VAF). The final function point count is obtained by multiplying the VAF times the Unadjusted Function Point (UAF).
9. Symons et al., describes 14 GSC's : Mentioned in Pre System Algorithm

III. Comparative Study

Author	Algorithm	Advantage	Disadvantage
Boehm	COCOMO Models	1. Clear results, very common	1. Much data is required.

		2. Very profound information is easy available	2. Duration calculation for small projects is unreasonable
Putnam	Putnam Model	1. The Putnam model is very sensitive to the development time: decreasing the development time can greatly increase the person-months needed for development	1. One significant problem with the Putnam model is that it is based on knowing, or being able to estimate accurately, the size (in lines of code) of the software to be developed. There is often great uncertainty in the software size. It may result in the inaccuracy of cost estimation.
Albrecht	Function Point Analysis	1. function points can be estimated from requirements specifications or design specifications, thus making it possible to estimate development cost in the early phases of development.	1. It needs subjective evaluations with a lot of judgment involved.
		2. function points are independent of the language, tools, or methodologies used for implementation.	2. Largely a manual process.

IV. Proposed Methodology

A. Pre System Algorithm

To measure the function point, we use two part. In first part, we use the software functionalities to measure the function point. This first part is known as Unadjusted Function Point. Unadjusted function point is counted by Data functionality and Transactional functionality. UFP estimation is arrive at a complexity level of high, medium and low and assigns a function point count for each subset.

- Data Functionality

  Data functionality represents the requirement (Functionality provided by user).

  I. Internal Logical File (ILF)

  II. External Interface File (EIF)

- Transaction Functionality

  Transactional functionality is an elementary process to move data.

  I. External Inputs (EI)

  II. External Outputs (EO)

  III. External Queries (EQ)

Table1: Complexity Value for each functionality

Complexity	ILF	EIF	EI	EO	EQ
Simple	7	5	3	4	3
Average	10	7	4	5	4
Complex	15	10	6	7	6

General System Characteristics (GSCs) is the second part. GSCs are measured from 0 to 5 nominal scales. These characteristics are used to calculate Value Adjusted Factor (VAF) . We use TDI to calculate VAF. TDI is the sum of the influence of all General System Characteristic. The final function point count is the multiplication of Value Adjusted Function Point (VAF) and Unadjusted Function Point (UAF) . 14 GSC’s are : Data communication, Distributed functions, performance, heavily used configuration, transaction rate, online data entry, End user efficiency, Online Update, complex processing, Reusability, installation ease, multiple sites, facilitate change. Influence rank of these 14 GSC's should be within o to 5. The standard equation for estimation is:

FP = UFP * VAF

Where UFP = Unadjusted Function Point

UFP is the sum of Data functionality and Transactional functionality

UFP = EI + EO + EQ + ILF + EIF.

VAF = 0.65+ TDI (Total degree of influence)/100

where TDI is the sum of the influence of 14 GSCs.

These 14 GSCs can affect the length of the project and rank of each can be " 0 " to "5".

VAF = 0.65 + TDI/100

B. Proposed System Algorithm

Software size estimation is necessary and critical activity. If the software contains new facility then the size of a software increases. Hence Expert User Programming also increase the software size. In the standard model, expert user facilities are not taken, while estimating the size of a software. we can introduce new characteristic to estimate software size. Here we have introduced a new general System Characteristic, and can be measured as:

VAF = 0.65 + TDI/100

Here TDI is the sum of all standard General System Characteristic and Expert User Programming (Total 15 GSC's).

Hence the VAF of all 15 GSC's will be

VAF=0.65+TDI (Sum of all 15 GSC) / 100

Here TDI means Total degree of influence.

our new general system characteristic is classified as

we have defined our new General System Characteristic, for Expert User Programming into five different Phases.

Steps of Process Model	**Expert User Ingredients**	**Degree of Influence**
Planning	Expert User Requirements	3
Designing	Design of expert user features	4
Coding	Coding of Expert User tool and review	1
Testing	Testing & Review	2
Implementation & Installation, maintenance	Installation and monitoring	5

we take average of Influence of these five phases and add to the influence of 14 standard General System characteristic. so we can calculate the new function point count of all 15 GSC's.

we take average of Influence of these five phases and add to the influence of 14 standard General System characteristic. so we can calculate the new function point

V. Conclusion

In this paper we have proposed a new variant of the function point analysis. It provides more accurate software size estimation.

Proposed (General System Characteristic) FPA provides more accurate size estimates with existing GSC. Expert user programming environments will definitely have impact on software size and effort estimation for software projects. It was difficult for Software professionals to estimate the function point value for non Professional user who does not have any programming Knowledge and develop their own software. Expert User Programming focuses on enabling end users to create software, with the quality of software across software lifecycle from beginning to deployment. Proposed Solution helps to software professional to estimate accurate Function Point for expert user programming.

References:

[1] Albrecht A.J. and Gaffney J. E. "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," IEEE Transactions on Software Engineering, vol. SE-9, no. 6, November

[2] P.K. Suri, Pallavi Ranjan, June 2012. Comparative Analysis of Software Effort Estimation Techniques. International Journal of Computer Applications

[3] Tharwon Arnuphaptrairong, " Early Stage Software Effort Estimation Using Function Point Analysis: An Empirical Validation" IJDATICS, VOL. 4, NO. 1, DECEMBER 2013

[4] IFPUG, "IFPUG Counting Practices Manual - Release. 4.1," International Function Point Users Group, Westerville, OH

[5] IFPUG, "IFPUG Counting Practices Manual - Release. 4.2," International Function Point Users Group, Princeton Junction, NJ.

[6] Albrecht A.J. “Measuring Application Development Productivity,” Proc. IBM Applications Development Symp., SHARE-Guide, pp. 83-92.

[7] M. Pauline, P. Aruna and B. Shadaksharappa, “Software Cost Estimation Model based on Proposed Function Point and Trimmed Cost Drivers Using Cocomo II”, International Journal of Engineering Research & Technology (IJERT), Vol. 1 Issue 5, July - 2012.

[8] Symons C.R.. "Function Point Analysis: Difficulties and Improvements," IEEE Transactions on Software Engineering, vol. 14, no. 1, pp. 2-11

[9] http://www.slideshare.net/hemanthraj5439/introduction-to-software-cost-estimation

[10] http://www.devshed.com/c/a/Practices/An-Overview-of-Function-Point-Analysis/3

[11] B.W. Boehm, “Software engineering economics,” IEEE Transaction of Software Engineering.

[12] Jyoti G. Borade and Vikas R. Khalkar, " Software Project Effort and Cost Estimation Techniques " ijarcsse Volume 3, Issue 8, August 2013

[13] C.V.S.R.SYAVASYA, " An Approach To Estimate Function Point Analysis Using Unadjusted Function Points And Value Adjustment Factor " International Journal of Engineering Research and Applications Vol. 2, Issue 5, September- October 2012