International Journal of Scientific Research & Engineering Trends
Volume 7, Issue 2, March-April-2021, ISSN (Online): 2395-566X

Analysis of Software Project Cost Estimation using Functional Point

Research Scholar Sunil Kumar, Prof. Dr. Rohit Kumar Singhal
Institute of Engineering and Technology, Alwar

***Abstract*- Effort estimation has been used for planning and monitoring project resources. As software grew in size and complexity, it is very difficult to predict the development cost. There is no single technique, which is best for all situations. A careful comparison of the results of several approaches is necessary to produce realistic estimates. The use of workforce is measured as effort and defined as total time taken by development team members to perform a given task. It is usually expressed in units such as man-day, man-month, and man-year, which is a basis for estimating other values relevant for software projects, like cost or total time required to produce a software product.**

***Keywords*- Effort estimation , man-month, and man-year** etc.

I. INTRODUCTION

The estimation of development effort is the key factor to manage and control of a software project management. The major cost factor in software industry is cost of manpower in most cases. The estimation of effort can be used for planning and monitoring project resources. In software industry, the critical issue is the accurate prediction of the software development costs.

Similarly the accurate determination of how much effort and time a project required is also important for project managers, system analysts and developers. Without a reasonably accurate cost estimation capability, it will be difficult for the project managers to determine how much time and man power is required for a project.

II. SOFTWARE PROJECT MANAGEMENT

Software project management is the art and science of planning and leading software projects.

Analysis of software project management shows that the following are the most common causes of failures:

- Insufficient end-userinvolvement
- Poor communication between customers, developers, users and projectmanagers
- Unrealistic or articulated projectgoals
- Inaccurate estimates of neededresources
- Badly defined or incomplete system requirements andspecifications
- Poor reporting of the project'sstatus
- Poorly managedrisks
- Use of immaturetechnology
- Inability to handle the project's complexity

Most of the software is produced to meet the client's requirements. Due to the frequent changes and progresses of the technology in the software industry, the experience of one product may not be applied to the other one. These are the main risk in the software development process hence it is crucial to manage software projects efficiently.

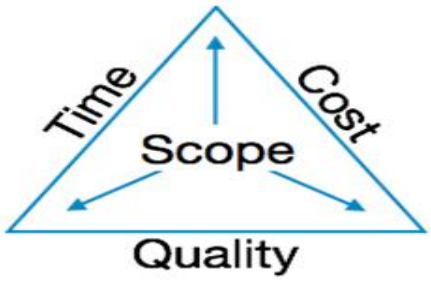

Fig 1. Constraints for the software Projects.

This figure 1.1shows the main constraints for the software projects. It is very vital for an organization to supply quality product, keeping the cost in client's budget constraints and delivers the project as per schedule. Several internal and external factors will be there to affect this constrain triangle. Any one of these factors can severely impact theothers.

Software project management is needed to incorporate user requirements together with budget and time constraints. It comprises of different activities, such as planning of project, scope of software product, cost estimation, scheduling of tasks and events, as well as resource management.

Project management include the activities such as,

- ProjectPlanning
- ScopeManagement
- ProjectEstimation

© 2021 IJSRET

1. Project Planning:
Software project planning is the phase, which is performed before the starting of actual software development process. It doesn't involve any concrete activity that has direct connection with software construction; but it is a set of multiple processes that facilitates software production.

2. Scope Management:
It defines the scope of project; which incorporates all the activities concerned about a deliverable software product. Scope management is important because it generates boundaries of the project by clearly specifying what should be done in the project and what should not be done. This form the project to contain limited and measurable tasks that can easily be documented and consecutively avoids cost and time overrun.

3. Project Estimation:
Accurate estimation of various measures is a must for an effective management. With faithful estimation, managers can direct and control the project more efficiently and effectively.

III. SOFTWARE ESTIMATION

Software estimation is the prediction of the most realistic value for different attributes/matrix of a software development such as size, effort, time of development etc. [45].

1. Software Sizee Etimation:
Size of the software can be estimated in terms of Line of Code as KLOC or the number of function points in the software or the number of use case points in the use case diagram of the software. KLOC depends upon the languages used for coding and function points may vary in accordance with the user or software requirement, Use case point may vary according to the use case diagram of thesoftware.

2. Effort Estimation:
The software effort is estimated in terms of person-hour required to produce the software. For estimating the effort, the software size is needed. This software size is derived either by managers' experience or by the historical data of organization. This software size can be converted into efforts by using the prescribedformulae.

3. Time Estimation:
The time of development of the projects can be estimated from the size and efforts of software. The different tasks of the software are divided into smaller tasks, activities or events by Work Breakthrough Structure (WBS)[45]. The tasks are scheduled on day-to-day basis or in calendarmonths.The total of time required to complete all these tasks in hours or days is the total time needed to complete the project.

4. Cost Estimation:
This is the most difficult task of all because it depends on more components than any of the previous ones. The following components are required for estimating project cost [45].

- Size ofsoftware
- Softwarequality
- Hardware
- Additional software or tools, licensesetc.
- Skilled personnel with task-specificskills
- Travelinvolved
- Communication
- Training andsupport

IV. FUNCTION POINT ANALYSIS

Function point analysis is a method for estimating size of the software from the user's point of view. According to Roberto Meli, Luca Santillo[37] FP (Function Points) is the most widespread functional type metrics which is suitable for quantifying a software application.

This metrics is based on a 5 user identifiable logical "functions". These functions can be divided into 2 data function types and 3 transactional functiontypes.

In this approach effort can be estimated as follows:
- Estimate the number of external inputs, external interface files, external outputs, external queries and logical internal tables (files).
- Use the Function Point Conversion Factor table to get total initial functionpoints.
- Initial function points are adjusted via 14 complexity factors to obtain final (adjusted) functionpoints.
- Use adjusted function points to obtainKSLOC.
- Use KSLOC to estimateefforts

1. Methodology:
There are different methods to estimate the development effort. Estimation of a software project development mainly implies the values of effort (in Person hours), time taken for development, and cost of development. For each software project, an estimation method should be selected so that it should comply with the actual values of these attributes.

2. Problem Statement:
Software Estimation is an important factor in the software development life cycle, it help managers tender on projects and allocates resources efficiently. So many methods exist to estimate the software efforts. Function point analysis and use case point analysis are two important methods among them.

We can identify the approximation of the methods to the actual from the comparison of these methods.

© 2021 IJSRET

Our work tries to answer the following questions:

- How much the software development projects deviate from the original plan, with respect to cost, schedule andfunctionality?
- Which methods are used to estimate software effort, and whether these systematically differ inaccuracy?
- How important is accurate effort estimation perceived to be, and how much is the level of accuracy considered a problem in the softwareindustry?
- What are the main causes for software projects to deviate from their originalplan?

3. Objective:

Comparison of function point method with use case point method of software effort estimation in order to make the decision process easy by performing case study on ten applications.

It requires the following steps:

- Study the software effortestimation
- Study the function point method and use case point method of effortestimation
- Select 10 applications for ourwork.
- Calculate the attributes of each application in both the method.
- The comparison of these attributes among both the methods of all applications

The following steps are involved for comparisons,

- Do until end ofapplication
- Take oneapplication
- Determine the external input, external output, external inquiries as well as the general system characteristics of theapplication
- Calculate the function pointcount
- Estimate the effort, LOC, Product Delivery Rate (PDR), Productivity and time taken for development from the function pointcount.
- Draw the use casediagram
- Determine the complexity of actors and usecases
- Determine the technical and environmental complexity factors of theapplication
- Calculate the use case pointcount
- Estimate the effort, LOC, Product Delivery Rate (PDR), Productivity and time taken for development from the use case pointcount
- Compare the values of attributes in both the methods with statisticalmethods
- Draw the graphical plot of theattributes

V. LIBRARY MANAGEMENT SYSTEM

This system gives the details of the books available in various departments, the transactions of books and about the book holders. It enables a fully automated library service, which makes the transactions easier andsystematic.

1. Use casediagram:

The use case diagram of library management application is shown in the Fig 3.1 and the complexities of different use cases present in the system are in the Table 3.1

Table 1. complexity Of Use Cases: Library Management Systems

Sr. No	Use cases	No. of transactions	Complexity of use case
1	Registration	3	Simple
2	Enquiry	5	Average
3	Verification of Member	1	Simple
4	Issue of Book	>7	Complex
5	Availability of Books	4	Average
6	Return Book	6	Average
7	Fine Calculation	4	Average
8	Maintenance of Books	5	Average

Function point analysis

Formula for counting function point is

FP=UFC * [0.65+(0.01*□Fi)] **Eqn (1)**

The general characteristics [5] of the application library management system is shown in the Table.

The complexity of different components of library management system and its unadjusted function count is shown in the Table 4.6

Table 2. Unadjusted Function Count: Library Management Systems.

Type of components	No of Components	Complexity of components	Value
External inputs	3	3	17
	2	4	
External outputs	1	4	14
	2	5	
External inquiries	1	3	11
	2	4	
Internal logical files	2	7	14
External interface files	1	5	5
	Total (UFC)		61

© 2021 IJSRET

Function point count for Library Management System= 58.58

FP = UFC * [0.65+ (0.01*ΣFi)] Eqn(2)

= 61*[0.65+ (0.01*36)]

= 58.58

Function point count for Library Management System= 58.58

The effort is calculated based on Jones first order estimate [23]

It gives an estimate for optimal schedule months from the function point count. According to C. Jones [23] the appropriate exponent, j, should be chosen according to the type of system and the general capability of the development team as per Table 4.7.

Table 3. Exponents For Jones's First-Order Estimate[23]

Kind of software	Best in class	Average	Worst in class
Systems	0.43	0.45	0.48
Business	0.41	0.43	0.46
Shrink-wrap	0.39	0.42	0.45

The following formula converts function points into total man-months.

Effort = f3*j/27 Eqn (3)

Where, f -> function point j -> Jones first order estimate So, effort =58.583*0.43 / 27 = 7.06 person- months

VI. CONCLUSION AND FUTURE WORK

A number of different models and effort estimation methods have been developed in the past four decades. This clearly indicates the awareness among the researchers to improve effort estimation in software engineering. There are many factors have impact on the software development process. These factors can be human, technical and their impact can never be fully predicted. We have studied the different estimation techniques and illustrated two approaches for measuring the size in the estimation process in this paper.

If the estimation is done accurately, it results in error decrease. The Estimation process reflects the reality of project's progress. It avoids cost/budget or schedule overruns. This process is quite simple which takes a few inputs. This assessment framework helps inexperienced team to improve project tracking and estimation. The effort estimation still remains unreliable. Too many techniques are developed including use case point, story point etc. to overcome this inability. In the future work, we can compare these techniques for their ability.

REFERENCES

[1] Boehm, B. Software Engineering Economics. Englewood CI@, NJ Prentice-Hall (198 1).
[2] Kemerer, C.F. An empirical validation of software cost estimation models. Communications of the ACM vol. 30, no. 5 (May 1987) 416-429.
[3] Albrecht & Gafney. Software function, source lines of code, and development effort prediction: a software science validation, IEEE Transactions on Software Engineering 9, 6 (1983) 639-648.
[4] Mukhopadyay, T., Vicinanza, S.S., Prietula, M.J. Examining the feasibility of a case-based reasoning model for software effort estimation. MIS Quarterly (June 1992) 155-171.
[5] Srinivasan, K., Fisher, D. Machine learning approaches to estimating software development effort. IEEE Transactions on Software Engineering, VOI. 21, no. 2 (February 1995)
[6] 126-137.
[7] Briand, L.C., Basili, V.R., Thomas, W.M. A pattern recognition approach for software engineering data analysis. IEEE Transactions on Software Engineering, vol. 18, no. II (1992) 931-942
[8] Allan J. Albrecht May 1984. AD/M Productivity Measurement and Estimation Validation, IBM Corporate Information Systems. IBM Corp.
[9] Barry W. Boehm, Bradford dark, Ellis Horowitz, Chris Westland, Ray Madachy and Richard Selby. Cost Models for Future Software Lifecycle Processes: COCOMO 2.0 Annals of Software Engineering. Volume 1, pp., 57-94, 1995.
[10] Magne Jorgensen. May 2007 Forecasting of Software Development Work Effort: Evidence on Expert Judgment and Formal Model.
[11] Parvinder S. Sandhu, Porush Bassi, and Amanpreet Singh Brar. 2008. Software Effort Estimation Using Soft Computing Techniques. World Academy of Science, Engineering and Technology 46 2008.

© 2021 IJSRET