International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 02 Issue: 09 | Dec-2015 www.irjet.net p-ISSN: 2395-0072

Evaluation of Software Hazard and Cost by Commercial Point-of-View

Ankur Srivastava[1] Mahesh Kumar Singh[2] , Abhimanyu Mishra[3]

[1 3] *Assistant Professor, Department of CSE, Jahangirabad Group of Institutions, Faculty of Engineering, Uttar Pradesh, India*

[2] *Assistant Professor, Department of CSE, Bansal Institute of Engineering & Technology, Uttar Pradesh, India*

Abstract - ***Function Point is a well known established method for commercializing cost of any software projects. There are several areas of the software engineering in which I can use the function point analysis (FPA) like project development, project construction, software execution etc.[1] This function point method is used for two different purposes, firstly, to estimation the risk in the software and secondly to estimate the cost of the software. In the literature of software engineering there are so many models to estimate the risk in the software like Soft Risk Model, SRAM, and SRAEM and so on. But in the proposed method we have used SRAEM i.e. Software Risk Assessment and Estimation Model, because in this model FP is used as an input variable, and on the other hand side, in sort to determine the cost of the software the International Software Benchmarking Standards Group Release Report (ISBSG) used.***

Proposed effort is helpful in cost estimation and risk analysis of any software. Broadly software can be divide into three category data processing software , examine software and system software On the basis of type of software cost and risk can be predicted. By by device guess of cost and risk can be done in well in advance, so that rejection and acceptance of projects are decided on the basis of results of cost and risk estimation models to estimate the risk in the software like Soft Risk Model, SRAM, SRAEM, SAEP and so on. As a background work SRAM and SRAEM are selected.

Key Words: Soft Risk Model, International Software Benchmarking Standards Group Release Report, Function Point, *etc…*

1. INTRODUCTION PROBABILITY STUDY

As a feasibility study in this project background and related work is chosen. As a part of cost estimation and risk assessment there is lot of work has been done till now. This section discussed about related work which has already been completed in past and there advantages and disadvantages.

To check feasibility it is best policy to analysis from previous work and find out opportunity of new form previous one. To estimate the risk in the software like Soft Risk Prototype tool ,SRAM, Software Risk Assessment and Estimation Model SAEP, and so on. As a background work SRAM and SRAEM considered. COCMO model and LINE OF CODE method are used as previous work taken for calculating cost of the software.

1.1 Assessment of the overall menace level of a task.

Let the nine risk element probabilities be denoted by **Rl, Rz, ..,Ri, ..,R9. [2]** As the nine risk elements have different degree of impact on different types of software project, different weights may be assigned to these elements when combining the risk element probabilities to derive at the overall risk value for the project.

Let the weights assigned to the elements be denoted by **Wl, W2,** .., **Wi,** .., **W9.** The risk level **R** of the project is then computed as **W1 Rl+ WZ R2+ ...+ Wi Rl+ ...+ WG R9.** If the maximum rating for all questions is **3** and the minimum rating is 1, then the maximum value of **R, R,,,,** is given by **R,,,** = **W1 3+ WZ 3+** ...+ **Wi 3+** ...+ **WG 3** = **3 (Wl+ WZ+** ...+ **Wi+ ...+ WG).** Similarly, the minimum **R, Rmin,,** is given by **Rmin = l*(Wl+ WZ+ ..+ Wi+ ...+W9).**

The overall risk level **R** may then be normalized as follows:

Normalized R

$$R, = (R - Rmin) / (Rmax - Rmin)$$

The normalized **R,** provides the risk level of the assessed project as a fraction between 0 and 1. **R,** for project with the lowest **risk** (no **risk)** is 0 and **R,** for project with the highest risk is 1.

1.2 Calibration of SRAM

Ten projects of a multinational company are used to calibrate the SRAM. For simplicity, these projects are named P1 to P10**.** The projects are all related to mobile software and point of impact of the risk basics on Quality, Schedule and Cost of the projects are similar. A common

set of values of weight **is** used. The values are those given in **Table 1** with HIGH=3, MEDIUM=2 and LOW=l.
The overall project risk is obtained by taking the average of the hazard values calculated for Quality, Schedule and Cost.

Table -1: Level of Impact of Risk Element

Risk Elements	Class	Schedule	Cost
Complexity	HIGH	HIGH	HIGH
Staff	HIGH	HIGH	HIGH
Requirement	HIGH	MEDIUM	MEDIUM
Reliability	HIGH	MEDIUM	MEDIUM
Development Process	MEDIUM	MEDIUM	MEDIUM
Estimation	MEDIUM	HIGH	HIGH
Monitoring	LOW	MEDIUM	LOW
Usability	MEDIUM	LOW	LOW
Tools	LOW	LOW	LOW

2. Opinion

The Customer Feedback Index is plot against the corresponding value of project risk for all 10 projects and is shown in Figure 2.1. A regression line is drawn through hall the ten points. This line then serves as calibration chart between overall project risk and Customer Feedback Index.
From data of the past software projects in the company, the Customer Feedback Indices for unaccepted projects are found to be below4.TheCustomer Feedback Indices for projects finished with extended schedule, low quality or cost overrun, are between 4 and 8. The Customer Feedback Indices for projects completed in time, within resources and according to specifications are between 8 and 12.

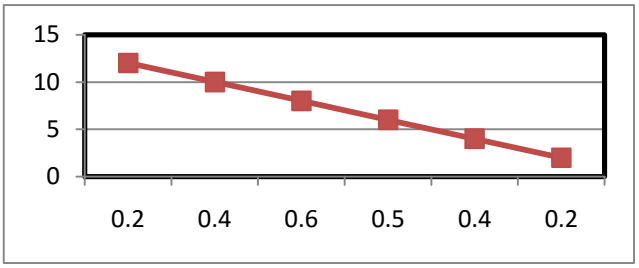

Chart -1: Feedback Index
Customer feedback index

By the next diagram we can see the risk involved in the factors that affect error but are not included clearly in the model contribute to the model error. For example such as 0.5 person-days per function point is usually obtained from results observed for recalled from previous projects. It is unlikely that any future projects will achieve the same ratio, but the model is expected to all right on average. If you base a model on past project data, you should calculate the associated inaccuracy by using the mean scale relative error. [5]

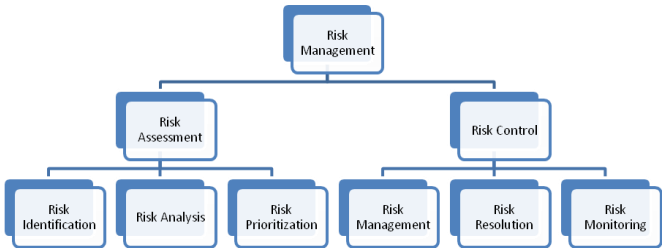

Fig 1.Steps Involved in Risk Management

The Customer Feedback Index is plotted against the corresponding value of project risk for all 10 projects and is shown in Chart1. [3] A regression line is drawn through hall the ten points. This line then serves as calibration chart among overall project risk and Customer Feedback Index as shown in the chart 1.

3. Hypothesis Fault

This occurs when we make incorrect assumptions about a model's input parameters. For example, your assessment that a product size is 1300 function point rests on the assumption that you have correctly identified all the customer requirements. If you can identify your assumptions, you can investigate the effect of their being invalid by assessing both the probability that an assumption is incorrect and the resulting impact on the estimate. This is the form of risk analysis. For example you believe that there is a 0.3 probability that the requirement complexity has been underestimated and, if it has, you estimate another 390 function point. At this point the concept of risk exposure is used to calculate the effective present cost of a risk and can be used to prioritize risk that requires countermeasure. Mathematically it can be written as Probability of risk occurring * Total loss if risk occur. These three errors are used to estimate the risk exposure. We have also included the MCRSRM in the proposed tool.

3. Software Cost Estimation Model

There are various existing techniques for calculating cost of the cost on the basis of size and human effort involve in developing the software like Line of Code and Constructive Cost Model.

3.1 COCOMO (Constructive Cost Model) model

The COnstructive COst MOdel (COCOMO) was launched in 1981 by Barry Boehm. It is often mentioned as COCOMO 81 to discern from its follow up, COCOMO II, discussed below. The model assumes that the size of a project can be estimated in thousands of delivered source instruction and then uses a non-linear equation to determine the effort for the project. The formula is for projects that can be achieved with 2-3 people teams. For "large" projects and

Effort = a * size

Effort = a * size + b

The parameters a and b are those that hold the interest.

3.2 Lines of Code (LOC)

This is the digit of lines of the delivered source code of the software, exclusive of clarification and blank lines and is generally recognized as LOC. [4] Although LOC is programming language dependent, it is the most widely used software size metric. Most models relate this dimension to the software cost. On the other hand, exact LOC can only be obtaining after the project has completed. Estimating the code size of a program before it is actually built is almost as hard as estimating the cost of the plan.

The traditional size metric for estimating software development effort and for measuring productivity has been lines of code (LOC).A large number of cost estimation models have been proposed, most of which are a function of lines of code, or thousands of lines of code (KLOC). [8] Generally, the effort estimation model consists of two parts. One part provides a base estimate as a function of software size and is of the following form:

$$E = A + B \times (KLOC)\ C.$$

where E is the estimated effort in man-months; A. B and C are constants; and KLOC is the estimated number of thousands of line of code in the final system.

The general characteristics of a system are:

(i) Data connections;
(ii) Distributed data processing;
(iii) Performance;
(iv) Heavily used Configuration;
(v) Transaction charge;
(vi) Online data entry;
(vii) End user efficiency;
(viii) Online update;
(ix) Composite processing;
(x) Reusability;
(xi) Equipment ease;
(xii) Operational ease;
(xiii) Multiple sites;
(xiv) Facilitate transform.

4. Results
Experimental Work

How the proposed -Tool would be useful to estimate the risk in software and also to estimate the cost of software presented in this section. In this work considered the projects developed in C/C++.

4.1 Evaluation of Cost

In the proposed tool we have used the International Software Benchmarking Standards Group (ISBSG). It is an international Group of representatives from international metrics organizations who collect project data from countries like, India, Hong Kong Germany, Japan, and USA. ISBSG Release 6 Report provides the cost value for the software projects. [6]

In the calculation of the FP, calculating the cost adjustment factor (VAF) is an earmark of the general functionality provided to the user. The VAF is derived from the sum of the degree of influence (DI) of the 14 general system characteristics (GSCs). The DI of each one of these characteristics ranges from 0 to 5 as follows:

(i) 0 – no influence;
(ii) 1 – minor influence;
(iii) 2 – reasonable influence;
(iv) 3 – standard influence;
(v) 4 – major influence;
(vi) 5 – tough influence.

The general characteristics of a system are : (i) data communications; (ii) distributed data processing; (iii) performance; (iv) heavily used configuration; (v) transaction rate;(vi) online data access; (vii) end user efficiency (viii) online update(ix) complex processing; (x) reusability; (xi) installation ease; (xii) operational ease; (xiii) multiple sites; (xiv) facilitate change. The third and the last stage is the ultimate calculation of the function points. With the help of the following equation we can get the total points of an application.

AFP = UFP * VAF.

Where AFP = adjusted function points;
UFP = unadjusted function points;
VAF = value adjustment factor. [1, 5, 8, 9]

Function points are computed by completing the TABLE 4 1. [7] Five information area characteristics are determined and counts are provided in appropriate table location [19].

In table 1 the MP is the measuring parameters, i.e. External Input (EI), External Output (EO), External Query (EQ), Internal Logical File (ILF), External Interface File (EIF) and UFP is the unadjusted function point.

Table 2. Calculation for adjusted function points

MP	Count	Simple	Average	Complex	Result
EI	X	3	4	6	
EO	X	4	5	7	
EQ	X	3	4	6	
ILF	X	7	10	15	
EIF	X	5	7	10	
UFP	**TOTAL**				

Once these data have been composed, a complexity value is associated with each count. Organizations that use FP method develop criteria for determining whether a particular entity is simple, average, or complex. To compute the AFP the following relationship is used:
Adjusted Function Points= Unadjusted Function Points *[0.65+0.01*(∑E_i)]

Where E_i is the summation of the total factors given from 0 to 5.

5. CONCLUSION

From the proposed device it is easy to estimate the risk in the software and also to estimate the cost of the software. The expenditure of the software depends on the value of the function point. In this effort we have applied the function point approach as an effort factor into the Tool. From the future tool it is easy to find out the outlay of software and estimation the risk of that software's project that were designed and developed by in C and C++ programming languages. In future we will elicit the software requirements after adding the threat in to it and then we will prioritize it using analytic ladder process and quality function deployment, and after this we will generate the results of that software with the proposed Tool.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Abdul Rahman, Head of Department, Jahangirabad Group of Institutions, Faculty of Engineering, Barabanki; and Mr. Mahesh Kumar Singh, Assistant Professor, Bansal Institute of Engineering & Technology, Lucknow, for his valuable support, guidance and encouragement.

REFERENCES

[1] Chris F. Kermerer, "Reliability of Function Points Measurements, A Field Experiment, Communication of the ACM, pp. 85- 97, Vol.36, February 1993.

[2] Daya Gupta, Mohd Sadiq, "Software Risk Assessment and Estimation Model", International Conference on Computer Science and Information Technology, IEEE Computer Society, Singapore, 2008. pp 963-967.

[3] Joseph S. Sherif, "Metrics for Software Risk Management",ISMN#0-7803-3274-1, pp.507-513.

[4] Mohd. Sadiq, Shabbir Ahmed,"Relationship between Lines of Code and Function Point & its Application in the Computation of Effort and Duration of a Software using Software Equation", International Conference on Emerging Technologies and Applications in Engineering, Technology and Sciences , ICATETS2008, Rajkot, Gujarat.

[5] Tom Demarco, Tim Lister, "Risk Management during Requirements", IEEE Computer society, IEEE Software, pp.99-100, 2003.

[6] Roger L. Van Scoy, "Software Development Risk: Opportunity, not problem", Technical report, September 1992.

[7] Low G.C. and Jeffery D.R.1990, Function Point in the Estimation and Evaluation of the Software Process, IEEE Trans Software Engineering, Vol.16, no.1.

[8] Mohd. Sadiq, Abdul Rahman Shabbir Ahmad , Mohammad Asim , Javed Ahmad ," esrcTool: A Tool to Estimate the Software Risk and Cost ", 2010 IEEE International Conference on Computer Research and Development ," Kuala Lumpur, Malaysia (Accepted for publication in the proceeding).

BIOGRAPHIES

Ankur Srivastava, B.Tech in Information Technology from Azad Institute of Technology. (29/08/1985), Assistant Professor, JETGI, Barabanki.

Mahesh Kumar Singh (14/02/1981), Assistant Professor, BIET, Lucknow.

Abhimanyu Mishra (01/06/1987), Assistant Professor, JETGI, Barabanki.