

“An Expert Estimator Tool to Estimate Project Cost and Risk with early stage of function points”

¹Ajay Jaiswal, ²Meena Sharma

¹Asst. Professor, Department of Computer Science & Engineering
Chameli Devi School of Engineering, Indore (MP)-452020, India.

Email: jaiswal.ajay05@gmail.com

²Reader and HOD of Computer Engineering
Institute of Engineering & Technology, DAVV, Indore (MP)-452001, India.

Email: meena@myself.com

ABSTRACT

Software estimation provides an important tool for project planning; whose quality and accuracy greatly affect the success of a project. There are several area of the software engineering in which we can use the function point analysis like project planning, project construction, software implementation etc. In software development, accuracy and efficiency of cost estimation methodology for a web based application is very important. The proposed web based application (i.e. Expert estimator), is to produce accurate cost estimation and risk estimation throughout the software development cycle to determine feasibility of software project. Cost of the software projects depends on the project size, project type, cost adjustment factor, cost driven factors, nature and characteristics of the project. Software estimation needs to estimates or predict the software costs and software risk early in the software life-cycle.

The process of risk management embodies the identification, analysis, planning, tracking, controlling, and communication of risk. It gives us a structured mechanism to provide visibility into threats to projects success. Risk management is a discipline for living with the possibility that future events may cause adverse effects. Risk management partly means reducing uncertainty. The propose tool indicates the risk & estimates risk using risk exposure. Management team to estimates the cost & risk within a planned budget and provide a fundamental motivation towards the development of web based application project. Find heuristic risk assessment using cost factors, indicating product & project risk using some risk factors & check some risk management strategies in under estimation development time.

In this paper we proposed the Expert estimator in Java, this tool is used to two different purpose, first to estimate the cost of the software & secondly, to estimate the risk in the software. Most of the software's fails due to over budget, delay in the delivery of the software & so on. Function point is a well known established method to estimate the size of software projects. Its measure of software size that uses logical functional terms, business owners & user, more readily understand.

KEYWORDS

Cost estimation, Expert judgment, Risk management, Type of risks, Expert estimator, Risk exposure.

INTRODUCTION

Software estimation is a step by step approach to estimating the cost & risk for every project. Estimation area of the software development is size, effort invested, development time, technology used & quality. Web cost project or application is needed accurate cost estimation,

producing accurate result. The recommended steps are used to identify the risks, determine the risk exposure, develop strategies to mitigate the risks, handle the risks.

Improving the functions of project management is a main concern in software development organizations. Suitable budgeting & cost are essential ingredients of a successful project. In that essence the total cost of product must be known at the early design stage, with the maximum of accuracy in order to simplify the trial & error process [1]. The cost of software projects depends on the nature & characteristics of the projects and therefore, the accuracy of the estimation model rely on the data by some evaluation affected by high degree of imprecision & uncertainty.

Albrecht's model [5] of functional specification requires the identification of 5 types of components, namely External input, External output, External inquiry element processes, logical internal elementary processes & logical internal & external interface files. The actual calculation process itself is accomplished in 3 stages:

1. Calculate the unadjusted function point (UFP)
2. Calculate the value adjusted factor (VAF)
3. Calculate the adjusted function points (AFP), other details advised please refer to [2][3].

Estimated cost mapped to function points. FPA is a measurement of functional requirement in terms of business transaction & business data. Transaction can be classified into three types of external inputs, external outputs, & external inquiries [3,15]. The process of risk management embodies the identification, quantification, response & control of risk. We need to balance the cost & indicating risk & minimize or reducing that risk. There are two dimensions of software risk, project risk and product risk.

We have explained the Expert estimator is organized as follows. In section I, we present the background and related work. In section II, we have explained the Expert estimator. In section III, we have explained experimental work carried out. Finally we conclude the paper in section IV.

Section I

1. Background & related work

The objective of this paper is to produce nearly accurate cost estimation & risk minimization reliable and accurate estimation of software development cost and risk estimation are needed throughout the software development cycle to determine feasibility of software project.

Cost of the software projects depends on the project size, project type, cost adjustment factor, cost driven factors, nature and characteristics of the project [1]. Software cost estimation needs to estimates or predict the project costs early in the software life-cycle. Many different models of the software cost & effort estimation have been developed and used during past decades, they are expressed such predefined functions like size of the product, level of reuse parameters etc., therefore an accurate estimation of the project cost of a software project will most likely lead to more successful results & on time completion.

Function point is a measure of software size that uses logical functional terms inputs and outputs. A risk is in exposure to loss or injury or a factor, thing, element or course that involves uncertain danger. Risk assessment involves risk identification, risk analysis, risk planning & risk controlling. Software risk can be internal or external; the internal risks come from risk factor within the organization. The external risks come from out of the organization & are difficult to control. Software risks can be grouped into project risks, process risks & product risks. [13]. The

proposed architecture gives the incremental risk as the software progress from phase to phase. In [4] the author have developed a software estimation tool based on software engineering metrics model, but in this tool there is no description regarding the costing of the software using ISBSG (International Software Benchmarking standard Group Release Report).

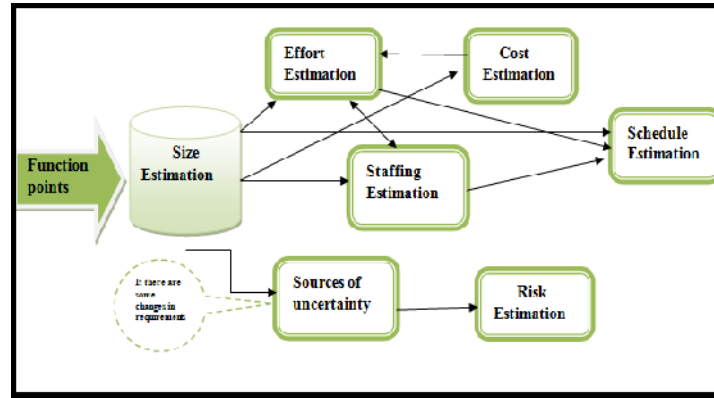


Fig.(1) Architecture of the proposed Expert estimator

In this paper we have developed the architecture of the Expert estimator tool in order to estimate the cost & risk of software. Architecture of the proposed Expert estimator is given in fig. (1).

Section II

2. Expert estimator

The proposed tool i.e. Expert estimator, indicate the risk of project & estimate the cost of project. A tool namely Expert estimator has been developed in the research by using java programming language and java eclipse based on techniques selected which are combination of applying function point & risk management process. Find out the cost of software & estimate the risk of the software.

“Estimation is a prediction that is equally likely above or below the actual results”. Estimation uncertainty occurs because an estimate is a probabilistic assessment of future condition. Risk assessment provides a snapshot of the risk situation & is part of a viable risk management program. There are four key factors of risk assessment & these factors are risk identification, risk analysis, risk planning & risk controlling. The first step of this tool is to calculate the function point of an input to the measurement error, model error & assumption error. The architecture of the proposed tool is given in fig.(1) adopted from [5].

2.1 Estimate the cost of project

In the proposed tool we have used the International Software Benchmarking Standards Group (ISBSG). The **ISBSG** delivers a database of software project history data that is used for estimation, benchmarking and project management. 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. Cost data is derived from 56 projects representing a broad cross section of the software industry. After going through these software projects, the ISBSG conclude that median cost to develop a function point is \$US 716, and the average cost is \$ US 849 per function point.

For more information about the ISBSG please visit: www.ISBSG.org.au [3]. In the calculation of the function point, calculating the value adjustment factor (VAF) is an indicate 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 14 GSCs show in table 1. The DI of each one of these characteristics ranges from 0 to 5 as follows:

- (i) 0 – no influence; (ii) 1 – incidental influence; (iii) 2 – moderate influence;
- (iv) 3 – average influence; (v) 4 – significant influence; and (vi) 5 – strong influence.

General System Characteristics(GSCs)	
1. General Data Communication Parameter	3
2. General Distributed Data Processing Parameter	0
3. General Performance Parameter	3
4. General Relatively Used Configuration Parameter	0
5. General Transaction Rate Parameter	2
6. General On Line Data Entry Parameter	0
7. General On Line Efficiency Parameter	2
8. General On Line Update Parameter	0
9. General Complex Processing Parameter	2
10. Scaled Reliability Parameter	0
11. Scaled Maintainance Parameter	2
12. Scaled Multiple Users Parameter	2
13. Scaled Hardware Charge Parameter	0
14. Scaled Operations Cost Parameter	1

Table 1: The general characteristics of a system.

The third and the last stage is the final calculation of the function points. With the help of the following equation we can get the total points of an application.

$$AFP = UFP * VAF \quad [3,9,10,\text{and } 16]$$

Where **AFP** = adjusted function points; **UFP** = unadjusted function points; and **VAF** = value adjustment factor. [1, 8, 9] Function points are computed by completing the table 1. Five information domain characteristics are determined and counts are provided in appropriate table location [9]. In table 2 ,the type of components, i.e. External Input (EI), External Output (EO), External Query (EQ), Internal Logical File (ILF), External Interface File (EIF) and UFP is the

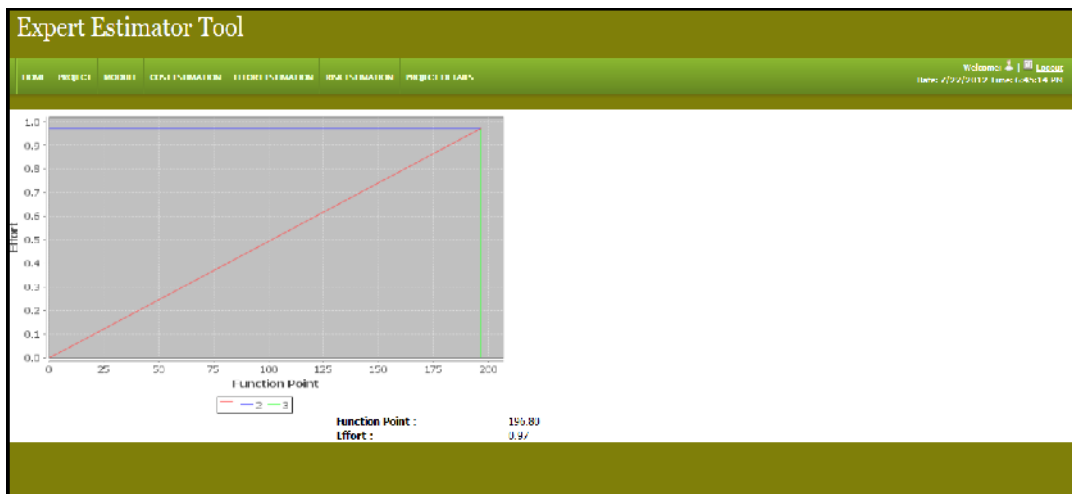


Fig.(3): Graphical representation function point verses effort calculation.

unadjusted function point. Once these data have been collected, a complexity value is associated with each count. Organization that use FP methods develop criteria for determining whether a particular entity is simple, average, or complex. To compute the AFP the following relationship is used [11, 12, and 13]. Fig.(2), show the size calculation using the function points & fig (3) show the chart of function point verses effort calculation.

$$AFP = UFP * [0.65 + 0.01 * \sum_{i=1}^{14} (DI)]$$

2.2 Estimation of the risk

Each software model has some weakness & also has some advantages. Every software project is exposed to adverse external influences, the so called project risks, which affect the cost and the duration of the project and, possibly, the quality of the products. With a risk analysis it can be determined for a specific project what the risks are. These risks then should be included in a systematic and formal manner in the project estimate in order to obtain a realistic and reliable project estimate and a realistic project plan. There are three dimensions of software risk i.e. technical risk, organization & environmental risk. Risk assessment provides a snapshot of the risk situation & is part of a viable risk management program. There are 3 key factors of risk assessment and these factors are risk identification, risk analysis & risk prioritization [14].

- I. **Risk identification** produce list of projects specific risk items, likely to compromise a project success. A typical risk identification technique includes examination of drivers, assumption, analysis & checklist.
- II. **Risk analysis** assesses the loss probability & loss magnitude for each identified risk item and it access compound risk in the risk interaction [11]. Typical technique include performance models, cost models network analysis etc.
- III. **Risk prioritization** produces a ranked ordering of the risk items identified and analyzed. Typical techniques include risk exposure analysis, risk reduction leverage etc.
- IV. **Risk planning** helps to prepare you to address each risk items including the coordination of individuals risk item plans with each other & with the overall project plans.
- V. **Risk monitoring** involves tracking the project's program towards resolving its risk items & tracking corrective action where appropriate.

Estimating uncertainty risks

The most critical aspect to any long-term risk management or estimation plan is to recognize and communicate the fact that it is something that will become more exact, or accurate, over time. Initially the model estimates the source of uncertainty using measurement error, model error & assumption error. We have considered the concept of function points to explain the measurement error, model error & assumption error. Function point is an important software metrics which is used to calculate the approximate LOC, cost & effort of software.

A. Measurement Error (MeE)

For measurement error this is the recognition that some of the input variables in an estimation model might have inherent accuracy limitations. (This is definitely the case with function point-type metrics, which are generally going to be about 12% inaccurate). As a result of Chris F. Kemerer [6], function points are assumed to be at least 12% inaccurate. Thus if we estimate a product size of 2000 points, measurement error could mean that the read size is anywhere between 1760 and 2240.

B. Model Error (MoE)

For model errors, this is the recognition that the model used for estimation cannot generally include all the factors that affect the effort required to produce a software product. Factors that affect effort but are not included explicitly in the model contribute to the model error and this needs to be recognized. This error occurs because all empirical models are abstraction of reality. For eg. Such as 0.5 person-days per function point is usually obtained from results observed for recalled from previous projects. Model is expected to all right on average. If you have a base model on past project data, you should calculate the associated inaccuracy by using the mean magnitude relative error. Thus if you have an estimation model with an inherent 20% inaccuracy & your product is 1000 function points in size, your estimate is likely to be between 400 & 600 person-days.

C. Assumption Error (AsE)

The assumption error happens when someone makes incorrect assumptions about the model's input parameters. So what we try to show management is that 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 just basic risk analysis but sort of gussied up to make it seem more technical than it is.) For eg. Your assessment that a product size is 1300 function points rests on the assumption that you have correctly identified all the customer requirements. If you can identify your assumption, you can investigate the effect of their being invalid by assessing both the probability that an assumption is incorrect & the resulting impact on the estimate. This is a form of risk analysis.

Section III

Experimental work

In this section we have presented how the proposed Expert estimator would be useful to estimate the risk in software and also to estimate the cost of software. For more detail about the Software requirements elicitation and prioritization, please refer to [6, 7, 8, 9 and 10]. According to the measurement error the actual size of the function point varies from 44-56. In order to find out the model error we have assumed that the 0.2 person-days per function point. No estimation model can include all factors that affect the effort required to produce a software product. Suppose we have an estimation model with an inherent 20% inaccuracy and our product is of 50 function points in size, our estimation is likely to be between 8-12 person days. The assumption error occurs when we have some incorrect assumption about the models input parameter. Our assumption is that the product size is of 50 function points rest on the assumption that we have correctly identified all the requirements. If we can identify our assumption, we can investigate the effect of their invalid by assessing both the probability that an assumption is incorrect and the resulting impact on the estimate. This is the called the risk analysis. If we assume that there is 0.4 probabilities that the requirement complexity has been underestimated, so we estimate another 2 function point.

$$RE = (E2 - E1) * P2, [11, 13]$$

Here RE means "Risk Exposure", E1 is the effort value if your original assumption is true (1000), E2 is the effort value if your new assumption is true (1100) and P2 is the probability that the new assumption is valid (0.1). So let E1 will then be the most likely estimate originally made (200 person-days) and E2 is the most likely alternative to that (220 = 1,100 * 0.2 person-days). So you

get: $RE = (220 - 200) * 0.1$. This gives you 2 person-days. That is your risk exposure. So this risk exposure of two person-days corresponds to the contingency you need to protect yourself against regarding the assumption error.

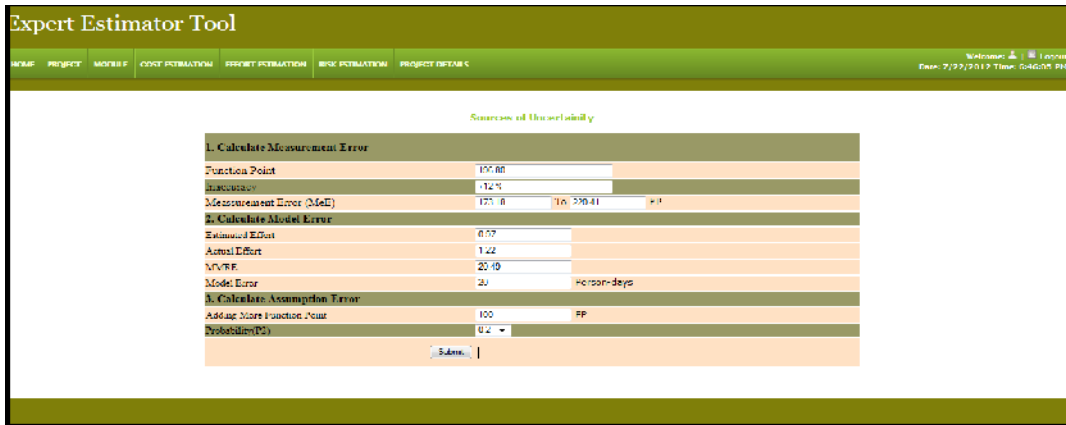


Fig.(4): Graphical representation of calculation of sources of uncertainty.

S.No.	Project with Function points	Measurement Error (MeE)	Model Error (MoE)	Assumption Error (AsE)	Risk Exposure (RE)
A	Project A	88 FP to 122 FP	16-24 Person-Days	Addition of 4 more FP	11 Person-Days
B	Project B	132 FP to 168 FP	28-32 Person-Days	Addition of 60 more FP	42 Person-Days
C	Project C	176 FP to 224 FP	38-42 Person-Days	Addition of 100 more FP	60 Person-Days

Table 2: Approximately results from Expert estimator

where **MeE** is Measurement error, **MoE** is Assumption error, **AsE** is Model error & **RE** is Risk Exposure. The results for the given software that we have got from the proposed Expert estimator are summarized in table 2.

So this risk exposure of two person-days corresponds to the contingency you need to protect yourself against regarding the assumption error. All in all this is not a bad contingency and is easily placed in the buffer of most project plans. However, the probabilistic nature of risk means that the allowed contingency cannot protect a project if the original assumption is wrong to begin with.

Section IV

CONCLUSIONS

In this proposed Expert estimator focuses on developing estimation tool for web based application. This tool namely (i.e. Expert estimator) is developing by using java as development

language & java eclipse as the development tool. Proposed tool easily estimate the risk in software & also to estimate cost of the software.

The cost estimation depends on the calculation of function points, cost adjustment factors & reuse. Function point approach as an input parameter into the "Expert estimator". This information is needed in the calculation of effort, schedule & total cost for the project.

The risk estimation based on the risk assessment of software projects. Risk identification, risk analysis & risk prioritization are the main subparts of risk assessment. From proposed model it is easy to calculate the risk at different phase as the software projects progresses from phase to phase. From further research, it is highly recommended that other cost estimating method is considered such as Price-to-win as an added method to cost estimation for web based application & software requirements after adding the thread in to it & then we will prioritize it using analytic hierarchy process & quality function deployment, & after this we will generate the results of that software using the proposed Expert estimator.

ACKNOWLEDGEMENTS

The authors would like to thank **Dr. CNS Murthy**, Professor and Dean, Chameli Devi School of Engineering, Indore, India; **Dr. Rajeev G Vishwakarma**, Professor at SVITS Indore, India; and **Mr. Preetesh Purohit**, Associate Professor, Chameli Devi School of Engineering, Indore, India, for his valuable support, guidance and encouragement.

REFERENCES

- [1] Khalid nadan, UAE University" Practical software project total cost estimation method, MCIT 2010 IEEE, page 7-10.
- [2] Zuse H. 1991., Software Metrics-Methods to Investigate and Evaluate Software Complexity Measures, Proc. Second Annual Oregon Workshop on Software Metrics, Portland.
- [3] International Function Point User Group (IFPUG), Function Point Counting Practices Manual, Release 4.0, IFPUG, Westerville, Ohio, April 1990.
- [4] 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
- [5] Function Point Analysis Difficulties and Improvements. Charles R.Symons, IEEE 1988, IEEE Transaction on software engineering, vol no. 14,1 January 1988.
- [6] D. Firesmith, "Prioritizing requirements", Journal of Object Technology, Volume 3, No.8, September 2004
- [7] J. Karlsson, "Software Requirements Prioritizing", Proceedings of the International Conference on Requirement Engineering, 1996.
- [8] LI Zong-yong, WANG Zhi-xue, YANG-ying, WU Yue, LIU Ying, " Towards multiple ontology Framework for Requirements Elicitation and Reuse", 31st IEEE Annual International Computer Software and Application Conference, 2007.
- [9] 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.
- [10] Mohd. Sadiq, Shabina Ghafir, Mohd. Shahid," A Framework to Prioritize the software Requirements using Quality Function Deployment", National Conference on Recent Development in Computing and its Application, 2009, organized by Jamia Hamdard, Delhi, India.
- [11] <http://www.aw.com/cseng>
- [12] Improving Estimation Practices by Applying Use Case Models, Bente Anda , Endre Angelvik , and Kirsten Ribu Simula Research Laboratory, Norway M. Oivo and S. Komi-Sirviö (Eds.): PROFES 2002, LNCS 2559, pp. 383-397, 2002. Springer-Verlag Berlin Heidelberg 2002
- [13] Improving Cost Estimation with Quantitative Risk Analysis Be More Precise by Employing Uncertainty by Gregory Nolder, Vose Consulting, www.voseconsulting.com.

- [14] "Classification and Analysis of Risks in Software Engineering" Hooman Hoodat, and Hassan Rashidi, World Academy of Science, Engineering and Technology 56 2009 .
- [15] "Function point estimation methods:A comparative overview", Roberto Meli, Luca Santillo Data Processing Organization.
- [16] Function Point Measurement Tool for UML Design Specification, Roberto Meli, Luca Satillo Data Processing Organization.

BIBLIOGRAPHY OF AUTHORS

¹**Ajay Jaiswal** joined in the Chameli Devi School of Engineering as an Lecturer in 2008 and became a Asst. Professor in 2012. He is currently working in the department of Computer Science Engineering. He received B.E. in Computer Science in 2006 from RGTU University, and M.E. in Software Engineering from IET, DAVV University. His current research interests include software engineering, soft computing, and natural language processing.



²**Meena Sharma** joined in the IET, DAVV University as an Associate Professor in 2007 and became a full professor in 2012. In addition, she is Head of Department of Computer Engineering. She received B.E. in Computer Engineering in 1992, M. Tech. in Computer Science in 2004 and Ph.D. in 2012. She serves as a Trainee Engineer in Carbon Corporation Ltd. Nashik, Software Engineer, Senior Software Engineer in Pertech Computers Ltd. Delhi, Assistant Manager in Hindustan Motors, Pithampur, Indore, and continues working for the IET, DAVV since 1998. Her current research interests include Software Engineering, Software Quality Metrics, Object Oriented Modeling and Design.

