CHAPTER 8

IMPLEMENTATION ANALYSIS OF HYBRID ESTIMATION TOOL

8.1 AN OVERVIEW

One important problem with software development project is to get an early and nevertheless accurate estimation of the effort needed to complete the project within the schedule. Despite the increasing needs and available tools and methods, a satisfactory solution is still to be found.

From the literature survey, it has been noted that traditional Function Point Method does not cover any of the quality factors. But, quality parameters play an important role in the estimation process. Hence, the quality assurance factors are combined with the traditional functional point method and were implemented in the hybrid tool and the weighting factors were fixed based on the questionnaire.

The comparative analysis of the existing software estimation and the hybrid method is done by the tool. The effort estimation has then been integrated with risk assessment to make the effort estimation more accurate Stukes et al (1998) & Laird et al (2006). The hybrid tool has been developed with Java as Front end and MS _ ACCESS as back end.

8.2 TRADITIONAL FUNCTION POINT METHOD

Function-oriented software metrics is used to measure the functionality delivered by the application as a normalization value, Gray and MacDonnell (1997). The most widely used function-oriented metric is
function point (FP). The traditional Functional Point metric method does not take into account the quality parameters of the software and moreover the values assigned to the parameters will be decided by the project manager’s experience. So, the values for the parameters will vary depending upon the manager’s interest on the project. As a net result, it is very difficult to estimate the software by these varying parameters. So, it is very difficult to use this value for estimating, the schedule, effort needed, project time, cost estimation etc. Substantial past project data has been collected by means of questionnaire issued to managers working in various software companies. Figure 8.1 show the login screen of the hybrid model. Figure 8.2 show the validation of the user name with password in the login screen. Figure 8.3 show the front end page of the hybrid tool. Figure 8.4 show the entry details of the user which is to be registered in the database. Figure 8.5 show the database schema with tables. Figure 8.6 show the Updation of user details in the database. Figure 8.7 show the deletion of records from the database. Figure 8.8 show the estimation methods that are carried out in the hybrid model. Figure 8.9 show the manual calculation of UCP method in the hybrid tool. Figure 8.10 show the automatic calculation of UCP method in the hybrid tool. Figure 8.11 show the functional point method with quality parameters in the hybrid tool. Figure 8.11 – Figure 8.15 shows the weight entered for each parameter in the tool. Figure 8.16 show the measured complexity of the method. Figure 8.17 show the total estimated weights in the tool. Figure 8.18 show the front end screen of the COCOMO model version II. Figure 8.19 and Figure 8.20 shows the SLOC Input module. Figure 8.21 show the SLOC input module with values. Figure 8.22 show the effort adjustment factor value. Figure 8.23 show the effort estimation screen by using COCOMO model. Figure 8.24 show the overall project results. Figure 8.25 show the estimated schedule of the project.
Figure 8.1 Log-in Screen

Figure 8.2 Log-in Screen with username
Figure 8.3 Front Page

Figure 8.4 Entering User details
Figure 8.5 Database Schema with tables

Figure 8.6 Updation of details in database
Figure 8.7 Deleting the records from database

Figure 8.8 Estimation Methods
Figure 8.9 Use case Methods

Figure 8.10 Automatic estimation
Figure 8.11 Entering Weighting factors

Figure 8.12 Entering weights
Figure 8.13 Calculating weights

Figure 8.14 Total weights
Figure 8.15 Quality parameters

Figure 8.16 Measuring the complexity
Figure 8.17 Total estimated weights

Figure 8.18 Front End Screen – COCOMO II version
Figure 8.19  SLOC Input – Module 2

Figure 8.20  SLOC Input – Module 3
Figure 8.21 EAF Factor Screen

Figure 8.22 Estimation Screen
Figure 8.23  Project Overall Results

Figure 8.24  Estimating Schedule
8.3 ANALYSIS OF THE TOOL

In the hybrid tool, effort estimation for a software development process has been calculated by using the following methods:

- Automated Hybrid Tool (New Tool).
- Use Case Point Method (Developed to suit for English Statements).
- COCOMO Method.
- Function Point Method (Traditional Method).
- Revised Functional Point Method with Quality parameters (NewTool).
- Lines of Code Method

Software effort estimation has been calculated by using all these models. By using the results, a comparative analysis is done in the form of a graph. The output of the results are shown in the Figure 8.25, Figure 8.26, Figure 8.27 and Figure 8.28.
Figure 8.25 Net result of all the estimated parameters

Figure 8.26 Graphical representation of estimated models
8.4 INTEGRATION OF SOFTWARE RISK WITH ESTIMATION

Once the effort has been found by using one of the above said estimation method, the User can select the method needed for estimating the software manually using menu – driven option from the tool. Once the effort has been found, then the result will be integrated with risk.

8.4.1 Software Risk Assessment

Large software projects have a very high frequency of schedule overruns, cost overruns, quality problems, and outright cancellations, Demirors et al (2000).
Every software project is exposed to adverse external influences, the so-called project risks that affect the cost and the duration of the project and, possibly, the quality of the products. Risk analysis determines what the risks are for a specific project. 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.

Project risks affect all aspects of a software project: the organization, the personnel, the technology etc. One can distinguish between two types of risk: direct risks – risk over which a project has a large degree of control – and indirect risks – risk over which a project has little or no control. Risks can be described by various characteristics. Two of them are the probability of occurrence and the cost of mitigating the problem in case the risk fires. A detailed risk analysis reveals the risks that threaten a specific project, determines the strategies on how to mitigate the project risks in case they fire and ranks their characteristics.

Risk assessment estimates the needed contingency due to the impact of anticipated risky events. The project manager can confidently commit to the aggregated result of effort estimation and risk assessment. Risk Method Tool was developed to test this approach.

Risk assessment Method process covers risk identification, analysis, prioritization, and monitoring. The tool does not support risk management planning and risk resolution. The basic version of the model supports two risk components – risk – to – effort. The Risk Exposure formula is given in equation 8.1.
\[ RE_j = P_j \times M_j + P_j \times \sum_{k} P_{j,k} \times M_{j,k} \]  

(8.1)

where

- \( P_j \) (event) - Probability of a risky event
- \( M_j \) (event) - Magnitude of that event’s direct impact
- \( P_{j,k} \) (outcome) - Probability of potential outcome k and
- \( M_{j,k} \) (outcome) - Magnitude of impact for outcome k

There may be several risk events for any risk item. Risk item with the largest absolute \( RE_j \) will be the company’s generic risk drivers. Risk Prioritization deals with risk ranking. The risk drivers are ranked according to the magnitude of \( RE_j \).

The objective of the questionnaire in Appendix – 2 is to determine the potential risk drivers for risk assessment. Six experienced project managers were then asked to go through the risk drivers. They were requested to comment on the completeness of the cost drivers in each category (i.e., are there any missing), their relevance (i.e., should this cost driver be considered at all), whether they ought to be further refined, and on any overlaps (i.e., cost drivers that ought to be combined). The risk driver definitions were subsequently revised, leaving a total of 10 risk driver categories. Then, the risk drivers were ranked according to the magnitude of their impact on effort. The six project managers were requested to rank the risk drivers within each category.

The output of the questionnaire analysis in Appendix – 2 is a minimal set of risk drivers that have the largest impact on the effort of projects. Based on this final ranking, twelve risk drivers were retained.
It can be seen from the following example and in Figure 8.28; integration of effort estimation with risk assessment gives a more accurate estimation. Assume that the initial estimated effort of a project is 50 person-months. Then the project manager hears there is a 50 percent probability that senior analyst AA may resign. The manager identifies the corresponding risk item as analyst capability (ACAP), which is one of Intermediate Cocomo’s cost drivers. The ACAP rating of AA is “high,” with a corresponding effort multiplier value of 0.86. The direct impact of AA’s resignation would be that any replacement must spend two person months reading and understanding the project results to date. Two potential replacement candidates, BB and CC, are available and have an equal probability ($P(outcome) = 0.50$) of being hired. BB’s ACAP rating is “nominal” and CC’s is “low,” which results in corresponding effort multiplier values of 1.00 and 1.19 respectively. Thus, the project’s revised, realistic estimate becomes 57.8 person months, because

$$RE = 0.50 \cdot (2\text{ pm}) + 0.50 \cdot (0.50 \cdot 50\text{ pm} \cdot (1.00 - 0.86)/0.86) + 0.50 \cdot (0.50 \cdot 50\text{ pm} \cdot (1.19 - 0.86)/0.86)$$

$$= 1.0\text{ pm} + 2.0\text{ pm} + 4.8\text{ pm}$$

$$= 7.8\text{ pm}$$

Fig 8.28 Risk Integration Form
8.5 HYBRID TOOL LAYOUT DIAGRAM

![Diagram showing Hybrid Tool layout]

Figure 8.29 Architecture of Hybrid Tool

8.6 RESULT ANALYSIS

The existing results for the Tested application in the software industry are given below:

- Based on the automated hybrid Tool, estimated effort is 63 person months.
- Based on the use case point model (automatic), the estimated effort is 64 person months.
- Based on the use case point model (manual), the estimated effort is 63 person months.
- Based on the COCOMO model estimate, the estimated effort is 49 person months.
Based on the Function Point metric estimate, the estimated effort is 62 person months.

Based on the derived Function point metric, the estimated effort is 64 person months.

Based on the LOC estimate, the estimated effort is 61 person months.

The average estimate (using all seven approaches) is 61 person months. The proposed tool automates complexity classification in the UCP method. It combines quality factors with functional point estimation and it also integrates risk with the estimated effort to give more accurate estimation. Additionally, the hybrid tool also estimates effort using traditional estimation methods and provides a graphical comparison between all estimation techniques.

8.7 TRAINING OF HYBRID TOOL

The Hybrid tool has been trained with a minimal set of keywords and parameters for a specific project and the architecture of the proposed model is shown in Figure 8.29.

8.8 ADVANTAGES OF CUSTOMIZED HYBRID TOOL

The main advantages of this developed hybrid tool are given below:

- This tool uses seven different cost estimation techniques.
- This tool integrates the software effort estimation with the risk assessment strategy.
- This tool gives a detailed explanation of the obtained output with the relevant graphical explanation.
• The user can select the estimation technique which he/she is interested.

• The calculated effort from this hybrid tool has been extended to various results which are used for analyzing the risk strategy of the software.