



RESEARCH ARTICLE

PRAGMATIC APPROACH ON COST ESTIMATION TOOLS

Gurpreet Singh¹, Shashi Kant Rathore², Ranjit Kaur³

¹Lovely Professional University, Punjab, India

²Lovely Professional University, Punjab, India

³Lovely Professional University, Punjab, India

¹ gurpreet619@gmail.com; ² shashi.mnit@gmail.com; ³ geet18@live.com

Abstract— Estimating the cost is one of the most important requirements of any of software project. We can't deliver according to expectations if we won't plan and we cannot plan until we know the underlying dependencies and estimates. So we need sort of cost estimation tools that are using the various available cost estimation techniques to estimate the overall or the individual module cost for each and every software project when it comes to development of any project in the industry. In this paper I'm going to study and discuss few of the available tools for estimating the cost for any software project to reduce the chances of failure that may happen because of inappropriate cost estimation. So if, one won't take appropriate care then one may have various uncertainties in developing any kind of software project in the industry. In this paper, a sort of comparative study on those available cost estimation tools in the market, based upon certain factor is done by me. You can use any one of them to estimate the cost of software project according to the feasibility and budget management of your organization.

Key Terms: - TOOL; software; project

I. INTRODUCTION

The software cost estimation is an important concept to discuss and calculate before the actual implementation of the software project in any of the IT organization. Most estimation tools are proprietary due the huge effort to consolidate underlying history databases. This explains the lack of mainstream open source software (OSS) estimation tools. Therefore, we look only at commercial off-the-shelf tools and technologies. Most tools here overlap in their underlying functionality, so the rest of this discussion doesn't indicate bias for a specific vendor. A very important requirement to analyze in an estimation tool is to have the opportunity to run benchmarks against best-in-class projects and browse within such data. That's why all major tools have recently included the International Software Benchmarking Standards Group (ISBSG) history database (www.isbsg.org), one of the most renowned public sources of information that's continuously maintained and updated. COMPARISON TABLE1 summarizes a collection of estimation tools. Some of the cost estimation tools that I'm going to discuss are COCOMO family, Agile COCOMO-II, CostXpert, Comparative Estimating tool, Early estimate checker, FP Outline, Knowledge Plan, Reality Checker, SEER-SEM, SLIM. These are the various sort of available tools are the in the market for the estimation of cost in case of any of the software project. Any of them can be useful for any of the projects as per company budget and requirement and size of the problem for which the cost is to be analyzed. These may or may not be available online. Few of them use

LOC mechanism and other(s) use function point mechanism, some of them have high cost, few of them costs medium while the rest are having cost. In this paper, I'm going to tell all the available tool (s) in brief and then finally I would be comparing these available tools based upon certain factor(s), so that it would be beneficial for the IT organization(s) (whether small / medium/ large) to select the best possible tool for them to estimate the cost of the project that they are going to implement.

II. TOOLS FOR ESTIMATING COST

A) COCOMO SUITE

In this what basically happens is, this COCOMO suite if constructive cost models are inbuilt inside this automated tool and the various parameter(s) are there which are taken as input i.e. initially software size (LOC/FP) and other factor(s) whose value(s) are to be inputted by the those senior person(s) in the organization who want to calculate the cost for software project. This COMOCO family suite contain(s) the implementation of basic COCOMO, COSYSMO etc. these are sort of cost estimation methods whose concepts are implemented in this automated software to find out the cost as shown by figure-1.

FIGURE -1

B) AGILE –COCOMO II

This particular Agile COCOMO II a kind of web based tool for estimating the cost of software project. It actually enables one to adjust their estimation by using the concept of analogy through identifying the factors that will be changing and by how much margin. It is a sort of tool which can take analogy **parameter in the form of following:**

- Total cost in dollars
- Total effort in person months.
- Productivity in dollars or Function points.
- Productivity in dollars or Lines of Code (LOC).
- Productivity in function points/person per month.
- Productivity in lines of code/ person per month
- Project velocity in total ideal person week(s)/iteration

It actually take(s) various inputs from the person who is going to conduct estimate for project such as lines of code, function points etc as shown by the figure-2 given below:

FIGURE-2 : it shows how tool operates.[1]

C) COMPARATIVE ESTIMATING TOOL

The **ISBSG[2]** Comparative Estimating Tool is basically an easy-to-use MS Excel i.e spreadsheet based tool that is used to generate estimates of **software project effort, delivery rate, duration and speed of delivery**. These estimates are determined from the averages of project delivery rates and speed of delivery for a selection of projects taken from the ISBSG repository that are deemed to be similar to the project for which the estimate is required.

Estimating

When we talk about the estimating , I can state that after you enter the attributes of your proposed project, the tool basically selects similar projects from the ISBSG industry data, to provide a range of estimates: “Optimistic”, “Likely” and “Conservative” for project effort, delivery rate, duration and speed of delivery.

Benchmarking

In this particular concept if you enter the attributes of your completed project, the tool will provide a range of values from the **ISBSG data for software project effort, delivery rate, duration and speed of delivery**, you can then compare these to the performance.

The Tool

The tool filters the ISBSG data to ensure a comparison of “apples with apples”. It then allows the user to specify up to eleven project attributes to further refine the choice of projects that can be compared to the project being estimated. A range of estimates is provided: “**Optimistic**”, “**Likely**” and “**Conservative**”. The estimates generated are for the core software development team (i.e project team, project management and project administration), they do not include development team support, computer operations support or end-users or clients.

It is a kind of Microsoft excel sheet which has various functionality as shown by figure-3:

PROJECT ATTRIBUTE(s)

These are the Specifying values for some or all of the following selection of project attributes Completes setting up the data for the new estimate:

- Primary programming language
- Organization type
- Application type
- Maximum team size
- Used prototyping
- Used JAD
- Web development

FIGURE-3:

ISBSG International Software Benchmarking Standards Group
Comparative Estimating Tool V6.0
Delivering IT Confidence using the ISBSG project data

Inputs

Filters

Functional size (function points):
Match all

Development platform:
Match all

Development type:
Match all

Parameters

Functional size (function points):
100

Range minimum (percentile):
20

Range maximum (percentile):
80

Estimate

Reset

Project attributes

Primary programming language: Match none

Organisation type: Match none

Application type: Match none

Maximum team size: Match none

Used prototyping: Match none

Used JAD: Match none

Web development: Match none

Number of matches	Project Delivery Rate			Speed of Delivery		
	Optim 20 %	Likely Median	Conserv 80 %	Conserv 20 %	Likely Median	Optim 80 %

Estimates

	Project Delivery Rate	Project Work Effort	Speed of Delivery	Project Duration
Level 1-Dev Team	<i>(hours per function point)</i>	<i>(hours)</i>	<i>(function points per month)</i>	<i>(months)</i>
Optimistic	Not Defined	Not Defined	Not Defined	Not Defined
Likely	Not Defined	Not Defined	Not Defined	Not Defined
Conservative	Not Defined	Not Defined	Not Defined	Not Defined

Developed in conjunction with CHARISMATEK SOFTWARE METHODS

Estimation is done using following formula(s):

- Effort (hours) = Size * PDR
- Duration (months) = Size / SD

Where PDR is the project delivery rate and SD is speed of delivery.

So ultimately, I can say that this layout as shown by FIGURE-3 takes various parameter(s) as an input and computes the effort as shown by the formula and duration required to complete the project.

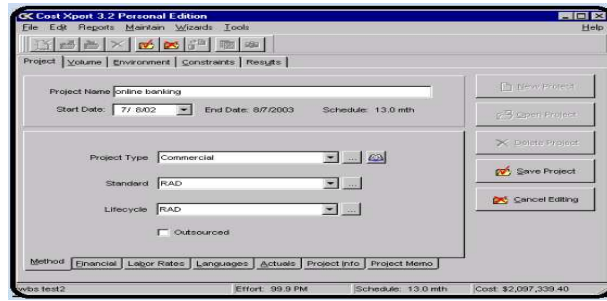
D) COST XPERT ESTIMATION TOOL

This tool termed as “CostXpert” is actually the industry's leading solution for software project **cost and schedule estimation**. This tool **differentiates itself from competitors by integrating multiple estimation models into one user-friendly solution** that provides the most accurate and comprehensive estimates possible it is also a kind of estimation tool for the software projects which has features as:

- In the Industry leader for < \$10K cost estimation tool market.
- It is easy to use for novices.
- It has sophisticated features for advanced users.
- It integrates multiple estimating models.
- It is most extensive options for project type, lifecycle and documentation standards.

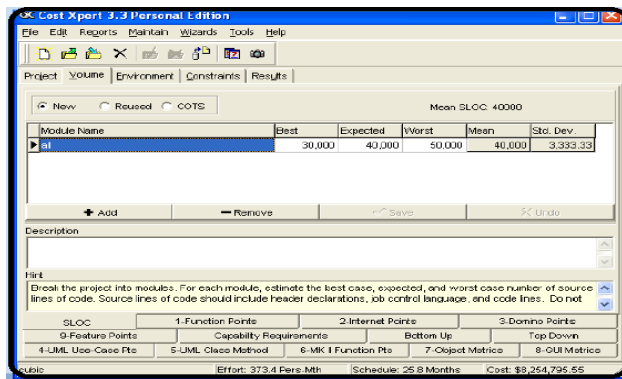
OPENING SCREEN TAB:

Opening interfaces as shown as below figure[3].



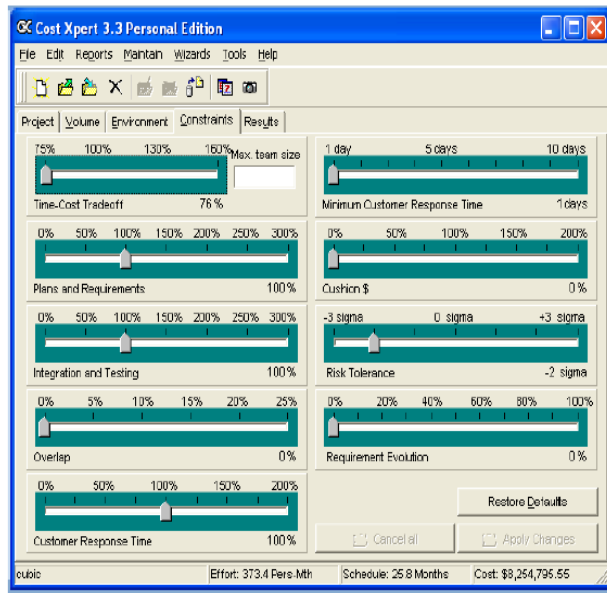
VOLUME TAB (modules):

It shows the calculation(s) for modules[3].



SETTING CONSTRAINTS:

Various time, plan and requirement and to the Integration and testing all are set in the constrain tab as shown below[3]:



RESULT SUMMARY

Final effort is shown in this tab[3].

The screenshot displays the 'Results' tab of the Cost Xpert 3.3 Personal Edition software. It features a table of methods and their associated effort and schedule, along with a detailed labor distribution cost breakdown and a summary of total effort, schedule, and cost.

Method	Effort (PM)	Schedule (mth)
<input checked="" type="checkbox"/> SLOC	373.4	25.8
<input type="checkbox"/> 1-Function Points	0.0	0.0
<input type="checkbox"/> 2-Internet Points	0.0	0.0
<input type="checkbox"/> 3-Domino Points	0.0	0.0
<input type="checkbox"/> 4-UML Use-Case Pts	0.0	0.0
<input type="checkbox"/> 5-UML Class Method	0.0	0.0

Labor Distribution Costs			
Strategist	\$251,148.89	Test/QA	\$1,139,545.47
Analyst	\$686,987.08	Copywriter	\$27,512.93
Designer	\$1,236,463.02	Art. & Media	\$15,001.87
Programmer	\$1,742,727.42	Management	\$1,400,705.20

Summary: Effort, Schedule and Cost	
Effort (Person-Months)	373.4
Schedule (Months)	25.8
Sub-Total	\$8,500,892.04
Wage Inflation (Schedule=12)	\$574,163.88
System Integration Cost	\$1,171,619.71
Final Cost	\$8,254,795.55

Summary: Effort: 373.4 Pers-Mth | Schedule: 25.8 Months | Cost: \$8,254,795.55

Finally I can say that this robust cross-industry tool allows managers to effectively forecast a complete project plan including scope, schedule, labor allocation, costs, risks and future maintenance.

CostXpert is specially designed to help an organization improve and manage **processes, mitigate risks, and exceed bottom-line expectations**

E) SEER –SEM TOOL

SEER –SEM is basically an algorithmic project management sort of software application which is devised to plan estimate cost and monitor the resources required for any type of software development and/or maintenance project.

SEER for Software is actually designed to be run inside of a **Windows platform**, and from version 6 onwards, is fully COM-enabled, it basically allows the users to have SEER interact with many Windows products including the product from Microsoft i.e **Microsoft Office**. Its current primary API relies on Microsoft Automation. The program itself is coded in **C++ and C**.

1. SOFTWARE SIZING CONCEPT:

In this particular concept the Software size is a key/basic input to any estimating model and across most of the available software parametric models. In this is there is the **Supported sizing metrics include source lines of code (SLOC), function points, function-based sizing (FBS)** and a range of other measures. They are translated for internal use into effective size S_e . S_e is a form of common currency within the model and enables new, reused, and even commercial off-the-shelf code to be mixed for an integrated analysis of the software development process. The generic calculation for S_e is:

$$S_e = \text{Newsize} + \text{existingsize} * (0.4 * \text{redesign} + 0.25 * \text{reimpl} + 0.35 * \text{retest})$$

2. FUNCTION BASED SIZING

The function based sizing (FBS) metric basically extends function points accordingly in such a way so that hidden parts of software such as complex algorithms can be sized more readily. FBS is translated directly into unadjusted function points (UFP).

In SEER-SEM tool, all size metrics are translated to S_e , including those entered using FBS. This is not a simple conversion, i.e., not a language-driven adjustment as is done with the much-derided backfiring method. Rather, the model incorporates factors, including phase at estimate, operating environment, application type, and application complexity. All these considerations significantly affect the mapping between functional size and S_e . After FBS is translated into function points, it is then converted into S_e as [4]:

$$S_e = Lx * (\text{Adj Factor} * UFP)^{\frac{\text{Entropy}}{1.2}}$$

3. EFFORT AND DURATION CALCULATION

A project's effort and duration are interrelated, as is reflected in their calculation within the model. Effort drives duration, notwithstanding productivity-related feedback between duration constraints and effort. The basic effort equation is given as below[4]:

$$K = D^{0.4} * \left(\frac{S_e}{C_{te}}\right)^{1.2}$$

Where,

- S_e is actually “effective size”. C_{te} is **effective technology**: this is basically a composite metric which basically captures factors relating to the efficiency or can say productivity with which development can be carried out. An extensive set of people, process, and product parameters feed into the effective technology rating.
- D is basically a **staffing complexity**: In this a rating of the project's inherent difficulty when we talk about in terms of the rate at which staff are added to a project.

Once effort is obtained, duration is solved using the equation given below[4]:

$$t_d = D^{-0.2} * \left(\frac{S_e}{C_{te}}\right)^{0.4}$$

F) SLIM COST ESTIMATION TOOL

Software Lifecycle Management (SLIM) tool support better decision making at each stage of the project development lifecycle i.e estimating, tracking, and metrics analysis. Each tool is designed to deliver results, either as a standalone application or as part of QSM's integrated suite of proven software metrics tools.

Putnam's SLIM is one of the first algorithmic cost model. It is based on the Norden/Rayleigh function and generally known as a macro estimation model (It is for large projects). SLIM enables a software cost estimator to perform the following functions:

Calibration: Fine tuning the model to represent the local software development environment by interpreting a historical database of past projects.

Build an information model of the software system, collecting software characteristics, personal attributes, and computer attributes etc.

Software sizing SLIM uses an automated version of the lines of code (LOC) costing technique.

The Formula used is:

$$K = (LOC / (C * t^{4/3})) * 3$$

G) REALITY CHECK TOOL

Reality Check is a quick and low cost method which Project Boards can use to check that, prior to project approval, **the development effort, cost and duration expectations** of the project plan are realistic. Various steps are given below in the check[6]:

Step 1: Determine the software size of the intended application.

Step 2: Enter the relevant project parameters into the Reality Checker

- Project Size (functional units)
- Platform (MF, MR, PC)
- Language (3GL, 4GL, Application Generator)

Step3: Conduct a ‘reality check’ on the two primary expectations

- Total developer effort (Project Work Effort)
- Elapsed Time (months)

Step 4: Interpret the results and decide intended actions (if any)

H) KNOWLEDGE PLAN TOOL:

It is a software tool which is basically designed to help you plan software projects. In this particular tool With SPR Knowledge PLAN you can effectively size your projects and then one can estimate work, resources, schedule, and defects. You can even evaluate project strengths and various sort of weaknesses to determine their impact on quality and productivity of the given software project.

LIVE DEMONSTRATION: it works as shown in below figure(s)[5]

I) FP OUTLINE TOOL

It is used to estimate the **size** in function points of your software applications or projects in minutes

J) EARLY ESTIMATE CHECKER ESTIMATION TOOL

In The Early Estimate Checker is a sort of software tool that uses Development and Enhancement Industrial Data and regression analysis to quickly generate estimates of the work effort and elapsed time (duration), required to carry out and complete a software development project, plus project delivery rate and speed of delivery.

It can be used to:

It **generates initial rough** estimates in the early stages of software development projects. It **validates existing project estimates** (e.g. bottom-up estimates generated from a project’s work breakdown).

It **assesses the reasonableness and likely risk associated with a quoted estimate** (where on the range from most conservative to most optimistic is the estimate positioned?)

Using the **Early Estimate Checker tool, estimates for project work effort and elapsed time can be generated by simply entering the project size** and it is expressed in **IFPUG function points**. By default, estimates consider all the validated data points contained in the current release of the ISBSG Industry Data. However, a combination of development platform, primary programming language type and development type can be selected to generate more specific estimates.

III. COMPARISON OF ESTIMATION TOOLS

Tools	Product	Lines Of Code or Function point based	Online	Proprietary	ISBSG data	Complexity (H/M/L)	Size(S) / Estimation (E)	Price Range
CO COMO SUITE	USC	Lines of code	Y	N	N	L	E	Free
AGILE CO COMO II	USC	Lines of code	Y	N	N	L	E	Free
COMPARATIVE ESTIMATING TOOL	ISBSG	Function point	N	Y	Y	L	E	Low
COST XPERT	Cost Xpert Group	Lines of code	N	Y	Y	H	E+S	Mid
SEER	Galorath	Lines of code	N	Y	Y	H	S+E	Mid
SLIM	QEM	Lines of code	N	Y	Y	H	S+E	High
REALITY CHECK	ISBSG	Function point	Y	Y	Y	L	E	Mid
KNOWLEDGE PLAN	SPR	Function point	N	Y	Y	M	S+E	Mid
FUNCTION POINT OUTLINE	Total Metrics	Function point	N	Y	Y	L	E	Mid
EARLY OUTLINE CHECKER	ISBSG	Function point	N	Y	Y	L	E	Mid

In this table 10 tools for estimating the cost of software project that I have discussed so far are compared based upon certain parameters shown in the table above.

IV. CONCLUSION

Ultimately, I can say that this paper makes one to know the list of available tools for estimating the cost of any type of software project in market. As one can see from the comparison table, SLIM is highly complex tool with high cost to use. So you can use any of the available lists of tools according to the requirements and certain input parameters. This paper makes thing(s) clear for the people in industry who want to estimate the cost of the project prior to implementation to reduce the chances of any kind of uncertainties that can deteriorate the project performance or that leads to delay. Hence finally I would say that these tools are compared on the basis of which are using LOC mechanism or FP mechanism, so one can deploy any of them to calculate the cost.

REFERENCES

- [1] [http://sunset.usc.edu/cse/pub/research/AgileCO COMO/AgileCOCOMOII/Main.html](http://sunset.usc.edu/cse/pub/research/AgileCO%20COMO/AgileCOCOMOII/Main.html)
- [2] <http://www.isbsg.org/>
- [3] <http://costxpert2003cocomoforumdemooverview.pdf>
- [4] <http://en.wikipedia.org/wiki/SEER-SEM>
- [5] <http://www.sprglobal.com/Demo/RegisterFormROW.asp>
- [6] <http://www.isbsg.org/isbsgnew.nsf/WebPages/adfdc34636703781ca2575460017d762>