

Project Management of Estimation, Planning and Tracking

Rima Debnath¹, Dr. Amit Asthana²

¹M Tech, Computer Science and Engineering, S.I.T.E, Swami Vivekanand Subharti University, Meerut

²H.O.D, Department of Computer Science and Engineering, S.I.T.E, Swami Vivekanand Subharti University, Meerut

Abstract: *Software engineering is the discipline which is the road map for development of software within the given schedule and effort and with the desired quality. The process begins with estimating the size, effort and time required for the development of the software and ends with the product and other work products built in different phases of development. The technique uses different parameters for estimation. For the estimates to be accurate, these parameters needs to be satisfied with the organizations' past projects' experience, failing to which leads to wrong estimates and consequently results in software crisis. The tools available for automating some of the activities are great help in the whole development process. Tools isolate the process of estimation, planning & tracking. Secondly Software Engineering is a still the metrics introduced for quantifying the attributes of software are not sufficient enough to the judgment. The goal of the thesis is to develop a web based tool for integrating the estimation (using COCOMO II model), planning & tracking process. The emphasis of the calibration process is to combine experts' judgments and organization's past projects' experience.*

Keywords: software engineering, estimation, COCOMO II

1. Introduction

Software engineering is the aggregates the application of scientific and technological knowledge functional specification, design description, program implementation, and test methods that lead up to test the code [1]. It is the software development process, highest degree of analyses, hard work. Software development has become a process and needs to take care of even the simplest activity in the development process.

Wrong estimation results a disaster in the development process. Effective estimation is essential for proper project planning and control, is one of the most critical and challenging task in the development process. Under-estimating a project leads to quality degradation, employee over exploitation and setting short schedule and hence results in missed deadlines. Over-estimating is even worse than the previous condition; allocating more resources to the project and thus increasing the cost of the project without any scope.

Proper planning of the project and tracking the project development is the second essential task for assuring the success of the project. Once the estimates are available the next task is to assign the tasks to individuals. Tracking gives opportunity to the project manager to take care of any unexpected situation while development.

Estimation is in the management of the development process, it is essential that the model or the method being used should be correct and satisfied with the most recent data available and if standard parameters are being used in the method then those parameters should be the available data.

2. Literature Review

2.1 Project management

A project is an effort put towards achieving an objective. The purpose of project management is to first find out the activities needed to take the project to its end and secondly to allocate resources to these activities in a planned way [4].

- 1) Allocation of resources
- 2) Organizing the work
- 3) Estimating resources
- 4) Planning the work or objectives
- 5) Analysis & Design of objectives
- 6) Acquiring human and material resources
- 7) Communicating to stakeholder
- 8) Directing activities
- 9) Tracking and reporting progress
- 10) Defining the product of the project
- 11) Forecasting future trends in the project
- 12) Assigning tasks
- 13) Controlling project execution
- 14) Issues solving
- 15) Analyzing the result based on the facts achieved
- 16) Project closure meets
- 17) Quality management

The triangle of relationship of the project management.

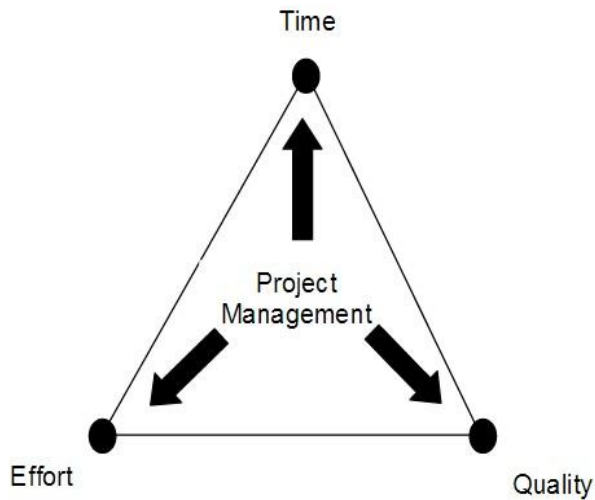


Figure 1: Relationship triangle

Quality, effort and time are inter-related. The first challenge that project management faces is to ensure that the project is delivered within time and budget and with the desired quality. The second challenge is more crucial and grueling one for optimizing the resource requirements.

2.2 Description of estimation

An effective estimation is the back bone for the development of any project. Without effective estimates proper project planning and tracking is impossible. If the estimates are too low then the project management tries to employ more personnel in order to expedite the development process; that eventually results in poor quality product and employee dissatisfaction [5].

2.3 Size estimation

A sound size estimate could be a good foundation for the software estimation. If the size estimation is being done in the later stages such as design or during coding, then design specifications and other work products can be used as information source for estimation [5].

2.4 Effort Estimation

1. By Using Past Projects' Data: The best way to derive the estimates for the project is to use data of the past projects. This approach also assumes that the organization has done similar projects earlier.

2. By Using Parametric Measures: If data of the past projects is not available then parametric models can be used for effort estimation. These models consider the features of the new project and use the standard values for these parameters for calculating the size of the software [5].

2.5 Schedule estimation

Estimation of schedule includes the number of people who will work on the project, what work they will do, what are the start time and end time for them. Once the effort estimates are available schedule can be laid out in calendar months.

Schedule in months = $3.0 * (\text{effort-months})^{1/3}$

2.6 Cost Estimation

Software cost estimation includes many factors to be hardware cost, labor cost, tools cost etc. How the cost would be estimated depends on the organization. Labor cost for developing the software constitutes the major portion of the total cost.

3. Architecture

The proposed tool follows the Model-View-Controller (MVC) architecture. MVC is an architectural pattern used in software designing. In a highly coupled design it is very difficult to make even the smallest change in the code. The need for the separation of business logic from the user interface is a major concern to any designer [7].

The MVC Pattern decouples the Model, View and Controller, and hence is a solution with the desired flexibility.

- 1) Different Graphical User Interface (GUI) is needed perhaps for different environment (Different Operating systems) without affecting the rest of the application.
- 2) Events generated by the user must immediately update application data or other components of the application, while the change in application data must be reflected to the user interface components immediately.
- 3) Reusability of one or more GUI components is needed independent of application data.

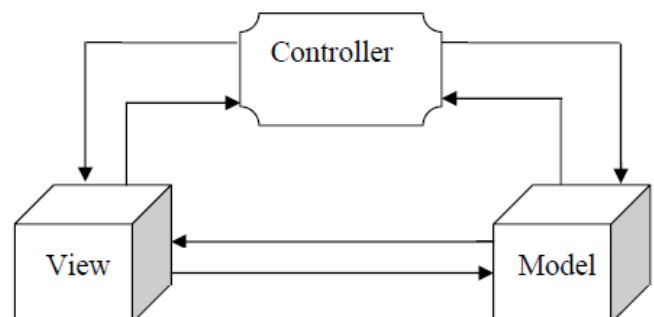


Figure 2: Model-View-Controller Architecture

- 1) **Model:** Represents the application data and functional logic in the form of a component.
- 2) **View:** The part visible to the user is known as view. A model can have more than one view according to the choice of the users. For example data can be represented in the form of tables or can be represented as charts.
- 3) **Controller:** View generates user events according to the action of the users. Controller is responsible for the processing of events and for any action taken for that action. It may or may not result in manipulation of user interface.

3.1 Java server faces

Java Server Faces (JSF) is a framework introduced which follows MVC architecture. In its model it keeps the java classes used in the software for the implementation of

business logic. JSP pages are used as view in the framework. JSP pages use the models for rendering of data and for processing the business logic. JSP pages cannot directly perform operation on the model. A servlet named 'Faces Servlet' is used as the controller in the framework [6].

3.2 Data access object pattern

Data Access Object (DAO) is a standard J2EE pattern. In large application where the separation of data from its data access method is necessary for maintaining the complexity low and the maintainability high, DAO is very useful [26].

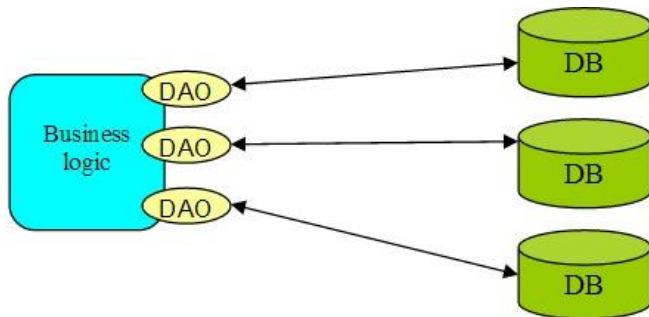


Figure 3: Data access object pattern

In the tool a XML file is used named as MyQueries.xml. The file contains all the queries for accessing the database and for any operation on it. Use of DAO pattern in the tool is shown in below:

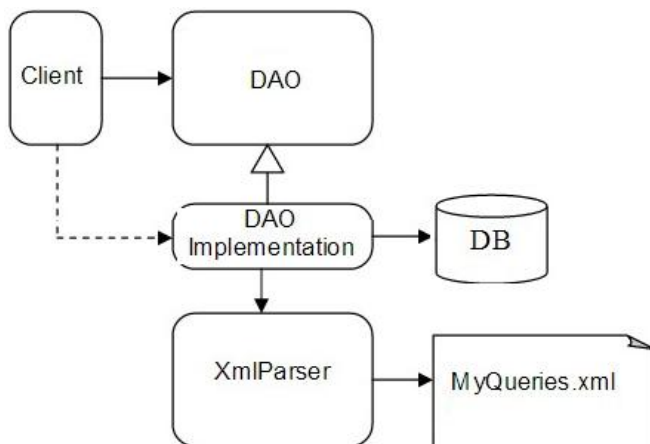


Figure 4: Implementation of Data Access Object Pattern in the tool

- 1) **DAO:** The data access interface that defines the methods to be implemented for accessing the data from the data.
- 2) **DAO Implementation:** Defines the methods declared in the DAO and is responsible for the connection to the database.
- 3) **Xml Parser:** Reads the information necessary for the database access from MyQueries.xml and sends the result back to the DAO Implementation.

The sequence of activities in DAO implementation:-

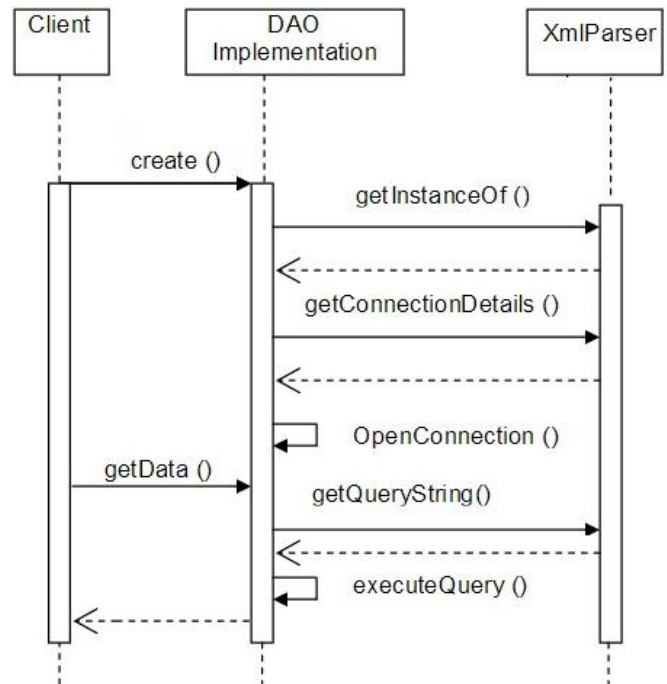


Figure 5: Sequence Diagram for Data Access Object Pattern in the tool

3.3 Package organization

The system is developed in 8 packages. A package is a collection of classes used in the application. The package organization of the tool is shown in figure 6.

The figure contains only the packages used for business logic and does not include classes used for GUI.

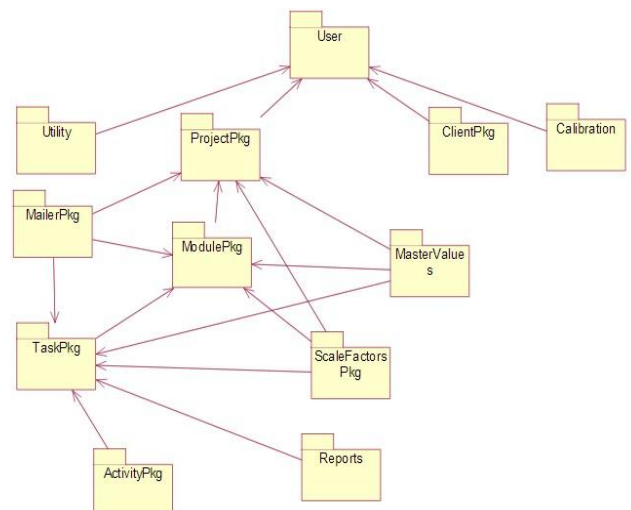


Figure 6: Organization of packages in the tool

3.4 Co-star

Costar is a software estimation tool based on COCOMO II. The tool is useful for generating estimates for size, effort, time duration and staffing level. This tool can generate reports for all the phases of development lifecycle, for the cost drivers, reports for schedule etc.

Costar 7.0 runs under Windows 95, Windows 98, Windows NT 4, Windows 2000, and Windows XP. Co-Star is a complete estimation tool. Co-Star is a perfect example of isolation of estimation process and management process in currently available tools. Co-Star does not store any past project data. Past projects' data .Co-Star does not provide any facility for any kind of project tracking [2].

3.5 Tracking

One of the most important activities in the whole management process is to keep track of the status of the project regularly to avoid any problem, and hence is a tiresome activity. Figure7 shows overall tracking process.

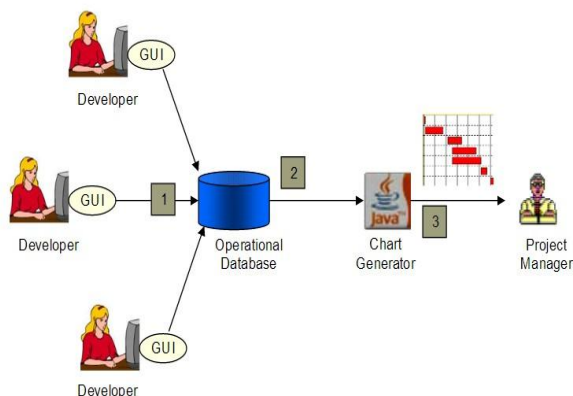


Figure 7: Overall tracking processes

- 1) Developers insert number of hours worked by them on each daily using a GUI into the operational database.
- 2) Data from operational database is retrieved for the project selected by the user.

4. A Road Ahead

- 1) The tool currently is not capable in estimating the size, effort and schedule for the prototypes and for the real time applications. This can be done with the implementation of application composition model of the COCOMO II and object point analysis.
- 2) Dependency among the tasks is given manually by entering the start dates of the modules. PERT and Critical path implementation can help in deciding the schedule for the projects and could be major amendment in the tool.
- 3) The tool stores the information about each activity in the task. This information with some data mining algorithm can be useful for inferring the problematic activity in organization and can indicate the administrator for modifications in the activity to overcome the problems occurred.
- 4) Stored information about the projects' estimates and developers' input can be used to identify the productivity of the developers.
- 5) The tool gives the estimates for the projects. A feature for estimating the best option under the given constraints (time, effort and productivity) can be added. If a user has some fix schedule then the productivity of the developers needed for the project to complete within schedule is a desired and common output for any organization.

5. COCOMO II 1999.0

This is a tool developed in University of Southern California comprises of estimation and calibration. It provides user with the facility to estimate size using three methods; function point analysis, source lines of code or adaptation source lines of code. It also provides the feature for estimating for the maintenance phase. The best feature of the tool is its flexibility that a user can even change the parameters' values used in the equation directly [3].

The tool does not have any management or tracking facilities but in calibration it can import data from any source file or the data stored in the tool of the earlier project (in this case still actual size and effort needs manual entry). The calibration method used in the tool is again multiple regression method which has its own drawbacks [11].

6. Conclusions

Software estimation helps project management to plan the project. Tools available for the project estimation are great helps in the process. But estimating the project and then planning it without caring about the status of project at any instant of time is a problem worth to be considered. The process known as tracking is an important process that needs to be integrated with the estimation and planning process.

There are much severe consequences of using a static model for estimation. The core of software crisis starts with the wrong estimation. Thus the calibration of the model being used for the estimation, with the past projects' data experienced by the organization, is an activity of utmost importance.

Software Engineering has grown enough to invent successful software development models, estimation techniques, designs, architectures and testing methods. Hence the CASE tools available should incorporate features to combine the experts' judgments with the available metrics.

The thesis has represented the specification and implementation details of a web based tool for integrating estimation, planning and tracking process. The tool was developed with the intention to be used by project management team which responsibility of completing the project within time, budget and with the specified quality.

References

- [1] "McGraw-Hill Dictionary of Scientific and Technical Terms", 6th edition, published by The McGraw-Hill Companies, Inc.
- [2] "CoStar tool", www.softstarsystems.com
- [3] "COCOMO II Model definition manual", version 1.4, University of Southern California.
- [4] Kathleen Peters, Berkun, Scott, "Art of Project Management. Cambridge", MA: O'Reilly Media. ISBN 0-596-00786-8. Software Project Estimation (White paper), Software Productivity Centre Inc. (SPC) in Vancouver, British Columbia, Canada, 2005.

- [5] David Longstreet, "Function Points Analysis Training Course", www.softwaremetrics.com.
- [6] Bill Dudley, Jonathan Lehr, Bill Willis and LeRoy Mattingly, "Mastering Java Server Faces", Wiley publications, 2004.
- [7] Data Access Object Pattern, www.corej2eepatterns.com/Patterns2ndEd/DataAccessObject.html
- [8] Sunita Chulani, Barry Boehm, Bert Steece, "Bayesian Analysis of Empirical Software Engineering Cost Models", July/August 1999 (Vol. 25, No. 4)
- [9] Sunita Devnani-Chulani, A dissertation, University of Southern California, "Bayesian Analysis of Software Cost and Quality Models", May 1999.
- [10] J. O. Berger, "Statistical Decision Theory and Bayesian Statistics". Second Edition. Springer Verlag, New York. ISBN 0-387-96098-8 and also ISBN 3-540-96098-8, 1999.
- [11] Sunita Devnani-Chulani, A dissertation, University of Southern California, "Bayesian Analysis of Software Cost and Quality Models", May 1999.