

Planning in Software Development Projects: An Empirical Study

Dr. Mohammed Farooque,

Assistant Professor, Poona College of Arts, Science & Commerce, Pune

drfarooq4u@gmail.com

Abstract: Software project management can be defined as the process of making visible what is invisible. Specifically, the software engineering process is invisible because its progress is not immediately visible. Software products contain more complexity than other engineered artefacts. Software developers have to conform to the requirements of human clients. The project result in terms of time, cost and quality, and the satisfaction of customers. The project's value and feasibility are measured. Project managers typically use two evaluation tools to decide whether or not to pursue a project. First is the Business Case Document justifies the need for the project, and it includes an estimate of potential financial benefits. Second one is the Feasibility Study is an evaluation of the project's goals, timeline and costs to determine if the project should be executed. It balances the requirements of the project with available resources to see if pursuing the project makes sense. Once the project receives the green light, it needs a solid plan to guide the team, as well as keep them on time and on budget. A well-written project plan gives guidance for obtaining resources, acquiring financing and procuring required materials. The project plan gives the team direction for producing quality outputs, handling risk, creating acceptance, communicating benefits to stakeholders and managing suppliers. This paper describes an empirical investigation on software project management.

Keywords: *Software Project Management, Software engineering, Software products, Business case document, Feasibility study*

INTRODUCTION

Software is defined (Pressman, 1997) as

- instructions (computer programs) that when executed provide a desired function and performance,
- data structures that enable the programs to adequately manipulate information, and
- Documents that describe the operation and use of the programs.

Software is a logical rather than a physical system element. According to Pressman (1997) a software product has three characteristics. Firstly, software is developed or engineered. It is not manufactured in the classical sense. Secondly, software is not “worn out”. It is not susceptible to the environmental condition which causes hardware to be worn out, but it may deteriorate.

According to the Institute of Electrical and Electronics Engineers (IEEE, 1993), software engineering is

- the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software
- The study of approaches to application.

A software process determines activities and organizational procedures to enhance collaboration in the development team so that a quality product is delivered to the customers (Leszek, 2001). Pressman (1997) described the software process as an approach that includes framework activities and umbrella activities as shown in Figure 1.

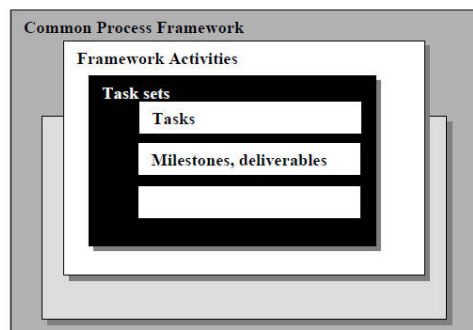


Fig.1. The Software Process (Pressman, 1997)

A Common Process Framework is established by defining a small number of framework activities that are applicable to all software projects, regardless of their size or complexity. A number of task sets each collection of software engineering work tasks, project milestones, software work products and deliverables, and quality assurance points which enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team. Finally, umbrella activities, including software quality assurance, software configuration management, and measurement overlay the process model. Umbrella activities are independent of the framework activity and occur throughout the process.

Software engineering occupies an intermediary position between the mathematical and physical disciplines of computer science and technology and the requirements of the particular application domains applying the findings of the former to solve problems of the application. The techniques for the engineering of software can be viewed, in part, as specializations of more general disciplines, such as project management, system engineering, and quality management (Moore, 2000). The relationship is shown in Figure 2.

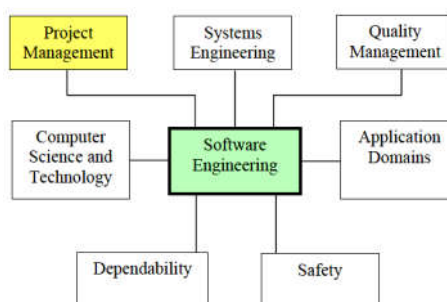


Fig.2. Relationship of software engineering to other disciplines (Moore, 2000)

Computer programmers write, test, and maintain the detailed instructions, called programs or software that computers must follow to perform their functions. They also conceive, design, and test logical structures for solving problems by computer. Many technical innovations in programming advanced computing technological innovations in programming tools – have redefined the role of a programmer and elevated much of the programming work done today.

CONCEPTS OF SOFTWARE PROJECT MANAGEMENT:

The Institute of Electrical and Electronics Engineers (IEEE) (1987) defines “software project management is the process of planning, organizing, staffing, monitoring, controlling, and leading a software project”. This is a common definition of project management.

According to Hughes and Cotterell (2002), Software project management can be defined as the process of making visible what is invisible. Specifically, the software engineering process is invisible because its progress is not immediately visible. Software products contain more complexity than other engineered artifacts. Software developers have to conform to the requirements of human clients. Clients sometimes are inconsistent and organizations have lapses in their collective memory, or in communication that developers have to cater for. Lastly, software systems are likely to be subject to a high degree of changes. By this definition, managing a software project is not an easy task. This is the reason why project management in the Information Technology area has had a poor success rate.

FUNCTIONS OF SOFTWARE PROJECT MANAGEMENT:

According to Munns and Bjeimi (1996), the functions of project management are

- Defining the requirements, establishing the extent of work
- Allocating the resources required, planning the execution of the work
- Monitoring the progress and adjusting deviations from the plan

The Project Management Body of Knowledge Guide (PMBOK Guide) (Project Management Institute, 2000) describes five types of management processes:

- Initiating
- Planning
- Executing
- Controlling
- Closing

This guide specializes in process types with 37 aspects of processes that are applied to nine program management knowledge areas (Moore, 2000), namely:

- Project integration
- Scope
- Time
- Cost
- Quality
- Human resource
- Communications
- Risk and procurement

Human Resource Management:

Human resource management includes all the processes that will be undertaken to identify, secure and maintain an effective project team.

This may take place in a self-contained team management plan or it may be a section of the project plan. The team management plan may include staffing, team structure, team communication, conflict resolution, performance appraisal and training. A recent survey of leading software organizations in Europe identified that the quality of people in a software team is one of the most important factors in improving productivity and quality in software projects (Blackburn et al., 1996). To effectively manage the project team, the manager has to focus on training, motivating and evaluating the project personnel.

The function point metric was created to measure the size, productivity, quality and other attributes of software applications. The function point (FP) metric is composed of the weighted totals of five external aspects of software application, namely inputs, outputs, logical files, inquiries and interfaces (Jones, 1998).

Quality Management:

Quality management includes the activities and techniques used to ensure that all project activities and work products comply with all relevant standards, procedures and requirements. Software quality refers to attributes as reliability, testability, modifiability, portability; usability and efficiency. Reliability is the degree to which the product meets its functionality over a measured period of time. Testability is the measure of difficulty in testing a product. Modifiability is the measure of effort required to modify and test a product to allow that a product to run in an environment other than the one for which it was developed. Usability is the measure of re-usability of code. Efficiency is a measure of the satisfaction of users regardless of their background, task performed, and needs (Yang, 2001; Gong, Yen and Chou, 1998).

Table 1: Examples of quality management systems

Quality System	Organization
ISO 9000	International Organization for Standardization
Capability Maturity Model (CMM)	Software Engineering Institute (SEI) – USA
Software Process Improvement Capability Determination (SPICE)	SEI and SC7
Total Quality Management (TQM)	

ISO 9000 has been effectively adapted for use in the software industry. From the ISO 9000 series, ISO 9001 is the most pertinent standard for software development and maintenance.

TQM is a paradigm and a philosophy first adopted in Japan. TQM originated in the manufacturing sector but is increasingly applied to other sectors such as the service industries, health, government and education (Dean and Bowen, 1994). There is a growing trend of adopting the TQM philosophy to software development. Applying TQM to the software development process can control software quality and productivity (Gong, Yen and Chou, 1998). The key elements of TQM include customer focus, attention to process, continuous improvement, measurement and analysis of data, human factors, such as management and leadership.

The capacity maturity model (CMM) was created by the Software Engineering Institute in the late 1980s to help software organizations improving their software processes along an evolutionary path (Paulk, 1994). The CMM describes the principles and practices underlying software process maturity and assesses software process capability at five levels: initial, repeatable, defined, managed, and optimizing. Except for the first level, each level has a set of key process areas on which an organization should focus to improve its software process.

Time Management:

Time management includes the processes and techniques used to ensure the timely completion of the project. It involves the development and management of the project work activities and the project schedule.

PLANNING IN SOFTWARE DEVELOPMENT PROJECTS:

According to Hughes and Cotterell (2002) planning includes seven steps as presented in Figure 3.

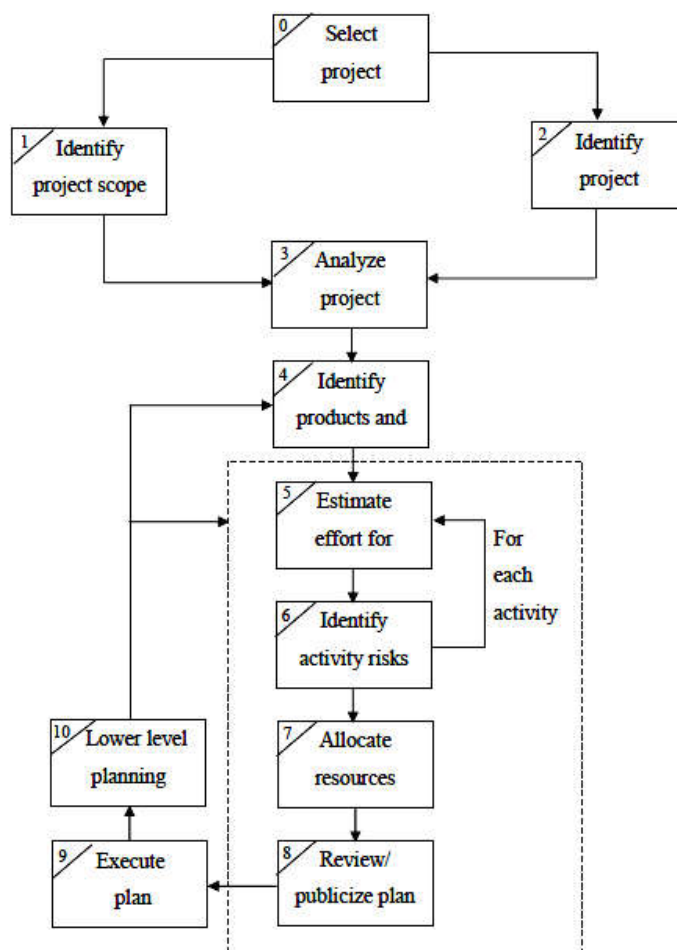


Fig.3. Step wise planning activities (Hughes and Cotterell, 2002)

Step-1: To identify the project scope and objectives. The activities in this step ensure that all the parties to the project agree on the objectives and are committed to its success. A common problem is overlooking people who are affected by the project.

Step-2: To identify the project infrastructure. There is usually an existing infrastructure into which the project can fit. Project leaders must identify the precise nature of this infrastructure.

Step-3: To analyze the project characteristics. This is to ensure that the appropriate methods are used for the project.

Step-4: To identify project products and activities. A more detailed planning of individual activities is done.

Step-5: To estimate the effort for each activity. Effort is the amount of work that needs to be done. The elapsed time, the time between the start and the end of a task is estimated based on this effort. The individual activity estimates of effort should be summed to get an overall actual estimate which can be reconciled with the previous planned estimate. These estimates could reveal that some activities are going to take a longer time.

Longer activities are more difficult to control. It would better to break this down into a series of smaller sub-tasks.

Step-6: To identify activity risks. The project manager should identify and quantify the project risks through considering each activity. A risk reduction approach is produced. The risk analysis in this step should consider the last step and change the estimation of effort for each activity.

Step-7: To allocate resources. Based on the estimated efforts in step five, the staff available for the project is identified and is allocated to specific tasks.

Step-8: The project manager will review the quality aspects of the project plan and then document this plan and obtain the agreements.

Finally, in **steps 9 and 10**, the project manager executes the plan and initiates lower level planning.

CONCLUSION

The current status of software project management is analyzed through three main areas of project management, including team management, quality management and time management. The step wise approach of planning provides the framework for conducting in depth interviews with project managers about their software projects. Project management has been discussed by executives and academics as one of the possibilities for organizations for integrating complex efforts and bureaucracy reduction. Managing projects effectively is introduced as a solution and as well as a major challenge for the business world. That is why both projects and project management have a highly important role in society and have been used as scientific research objects. Project management has become a core business process for a large number of companies both at strategic and operational level. Once software development involves a certain level of complexity and challenges, the techniques use, practices and project management tools have become common in software engineering. Nowadays it is very common for companies to deal with software development or service as a temporary project which needs to be planned, organized, conducted, monitored and controlled.

REFERENCES

1. Hughes, Bob and Cotterell, Mike (2002). *Software Project Management, Third edition. The McGraw – Hill Publishing Company*, pp.3.
2. A. Shenbar and D. Dvir, *Reinventing project management: the diamond approach to successful growth and innovation*. Harvard Business Press, 2007.
3. <https://www.villanovan.com/resources/project-management/5-phases-project-management-lifecycle/#.W7EEoGgzbDc>
4. <https://doc.rero.ch/record/6056/files/NguyenQM.pdf>
5. Aladwani, A.M. (2002), "IT project uncertainty, planning and success. An empirical investigation from Kuwait", *Information Technology & People*, Vol. 15, No.3, pp.210 –226.
6. IEEE (1987) (Revised 1993) *Standard for Software Project Management Plans, Chapter 7,13, IEEE Standard*. IEEE.
7. Gilb, T. (1988), *Principles of Software Engineering Management*, Addison-Wesley, Wokingham.