The Brief Introduction of Software Project Management

Zhu Yiyun*
13211210
Zhejiang Normal University
Software Project Management

Abstract

Software Project Management is an essential activity across several areas. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled. Many software organizations have problems delivering quality software, and through good management it is possible to achieve projects' deadlines, budgets and especially delivering a product that meets customer expectations. The software project management is more than just a technical engineering skill. And its activities encompasses planning, estimation, risk analysis, tracking and control. Considering the importance of managing projects, it is necessary to cover all the activities with software project management.

Keywords: software engineering, project management, planning, monitoring and control

1 Introduction

- Project is a temporary endeavour undertaken to create a unique product or service. And it put a clear emphasis on the project's being a planned activity.
- Project management is the discipline of planing, organizing, and managing resources to bring about the successful completion of specific project goals and objectives.
- Software project management is the art and science of planning and leading software projects.[Stellman and Greene 2005]. Figure 1 shows the relationship among these three technical terms.

Without project management, the projects is likely to succeed. But at the same time, it is difficult to guarantee the company's profit margins, and the company carry the greatest risk. Good management of a project in any part is important because the financial consequences of it being managed badly. It costs a great amount of money if the project might completed late or fail to achieve customers' real needs. For software projects, management becomes a particularly important part to make them right and doing more efficient.

And software project management can be seen as the process of setting objectives for a system and then monitoring the system to see what its true performance is. It is a sub-discipline of project management in which software projects are planned, implemented, monitored and controlled.

2 The Importance of Software Project Management

A software development process is concerned primarily with the production aspect of software development, as opposed to the technical aspect, such as software tools. Many software development processes
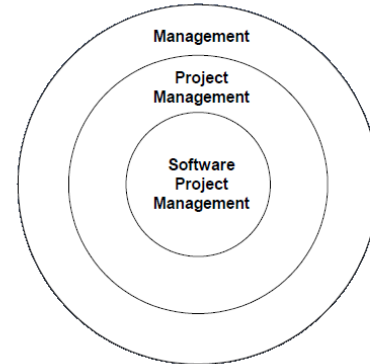
---

*e-mail:yunyun1003@outlook.com

Figure 1: *Relationship model - the project includes project management. The software project management is a sub-displine of project management.*

can be run in a similar way to general project management processes. In order to emphasis the importance of software project management, we shall introduce an extended development life-cycle for software.

Figure 2 contains this extended life-cycle which shows the major tasks which need to be performed before technical development can begin.[Darrel Ince 1993]. And as the part of software life-cycle, the software project management was being used in each stage of the project. When developers provided hundreds of methods, the software management was being use in each stage to narrow the gap between the temporary achievement of the project and customer expectations. In this way,the developer can fix the negotiation in order to avoid the mistakes.

For examples: When a project fails, it jeopardizes an organization's prospects. If the failure is large enough, it can steal the company's entire future. In one stellar meltdown, a poorly implemented resource planning system led FoxMeyer Drug Co., a 5-billion-dollars wholesale drug distribution company in Carrollton, Texas, to plummet into bankruptcy in 1996. [Charette].

3 Main Issue and Solutions

There are many steps in achieving a project, for the manager and programmers, they should defining the vision, planning the tasks, gathering the people who will do the work, estimating the effort, creating the schedule, overseeing the work, gathering the requirements, designing and programming the software, and testing the end product. And there would come up with lots of issues through the projects. The project manager has to take actions to make sure their projects are staying on track.

3.1 Issue

With small projects, the project leaders are likely to work very closely with other colleagues and might be carrying out many non-managerial tasks themselves and have a clear idea of what is going on. When the projects are larger, many separate teams may fail to have day-to-day
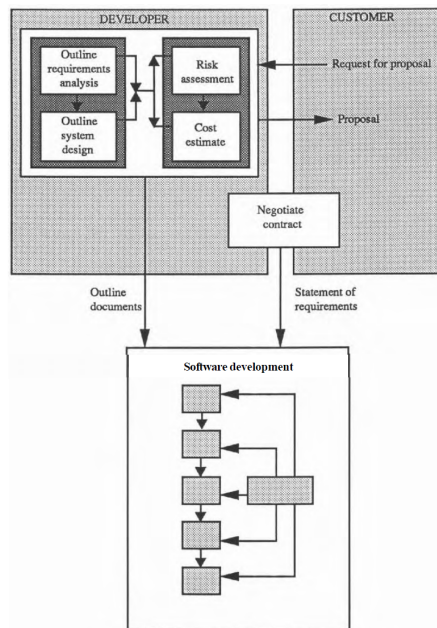
Figure 2: *An extended software life-cycle*

STELLMAN, A., AND GREENE, J. 2005. *Applied Software Project Management*. O'Reilly Media. 1

direct contact with all aspects of the work.

In computing, the term issue is a unit of work to accomplish an improvement in a system. An issue could be a bug, a requested feature, misunderstanding tasks, missing documentation, and so forth.

3.2 Solutions

- Planning The plans were used by many people in the organization. As a result of planning, it identified the scope of the project, estimate the work involved, and create a project schedule.
- Monitoring If the project deviates from the plan, then the project manager can take action to correct the problem. Project monitoring and control involves status meetings to gather status from the team. When changes need to be made, change control is used to keep the products up to date. The purpose of project monitoring and control is to keep the team and management up to date on the project's progress.

4   Conclusion

Software Project Management is an essential activity to ensure the project success. The main purpose of software project management is to make the whole software life-cycle(from analysis, design, coding, testing, maintenance the whole process) of software projects, especially large projects under manager's control. So the deadline, budget also the quality of the project meet stakeholders' needs. And by studying the software project management, people can learn the general principle from the previous cases, and avoid the mistakes.

References

CHARETTE, R. N. Why software fails. http://spectrum.ieee.org/computing/software/why-software-fails/3. 1

DARREL INCE, HELEN SHARP, M. W. 1993. *Introduction to software project management and quality assurance*. McGRAW-HILL Book Company. 1