What is Software Project Management?

Anson Yeoh/YANG XIANG
Zhejiang Normal University
Software Project Management

Abstract

With the rapid develop of technology, the requirement of software also dramatically increasing now. However, many instances in history illustrate that software industry need to be more effective in producing or it will be worse. The main point to address this question is trying to find out a right way to manage projects and coordinate jobs between staffs. Thus, the subject of Software Project Management was put forward in 1970s, and it is applicable to raise efficiency of development and reduce unnecessary costs. And now, this subject is becoming more and more vital in software industry cause the splendid results and many people get start to study in this area.

Keywords: software engineering, project management, efficiency, planning

1 Introduction

Software project management was put forward in the United States in the mid-1970s, when the department of defense devoted to the study of the software development cannot be submitted on time, and budget and quality of software were not up to the requirements of the user. The data showed that 70% of the projects failed because of the improper management rather than technical reasons. So the software developers began to pay more attention to the management of software development. [1]

In 1995, according to statistics, the United States to cancel the $81 billion of commercial software project, which 31% of the project unfinished is cancelled, 53% of the software project schedule is usually 50% longer than the time, only 9% of software projects can timely delivery and cost control within the budget. [1]

Software project management is very special compared with other project management. First of all, the software is pure knowledge products, the development progress and the quality is very difficult to estimate and measure, the production efficiency is also difficult to predict and guarantee. Secondly, the complexity of software system also led to difficult to predict and control the risks in the development process. [1]

For instance, the Windows operating system has more than 15 million lines of code at the same time, there are thousands of programmers and hundreds of project managers in the development. Such a huge system without good management, the quality of the software is hard to imagine.

2 Related Work

This subject is significant in this area, because it can make a great progress in our daily life and make more sense of our software develop work.

Many software organizations have problems delivering quality software that is finished on time and meets the users' needs. Luckily, most software project problems have surprisingly few root causes, and these causes are well understood. Solutions to these problems have been discovered, explained, and tested in thousands of software organizations around the world. These solutions are generally straightforward and easy to implement. However, they are not always intuitive to people who do not understand project management, and that makes them difficult to introduce. The goal of this book is to teach you about these solutions and help you integrate them into your own organization. [2]

Like electricity, water, transportation, and other critical parts of our infrastructure, IT is fast becoming intrinsic to our daily existence. In a few decades, a large-scale IT failure will become more than just an expensive inconvenience: it will put our way of life at risk. In the absence of the kind of industrywide changes that will mitigate software failures, how much of our future are we willing to gamble on these enormously costly and complex systems? [3]

3 Overview

In order to successfully develop software projects, managers should well know about the scopes of projects, potential risks, necessary resources (people, hardware, software), tasks, costs, schedule and so on. And this management and analysis should begin before the technology business began, continue in the software process from concept to implementation, when the software engineering process at the end to end. And this part is really important.

Principles:

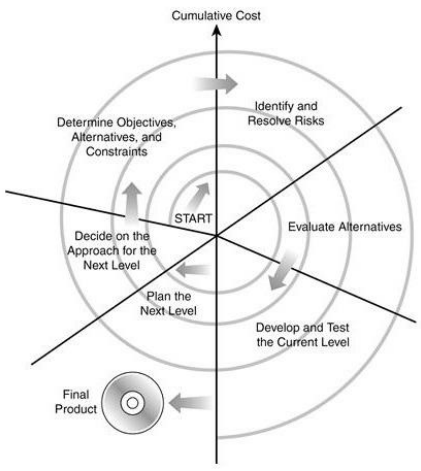
- Balance principle
- Efficiency principle
- Decomposition principle
- Real-time control principle
- Classification management principles
- Simple and effective principle
- Scale control principle

4 Methods/Techniques

Main Methods:

1. Platform Organize
2. Construction of Software Projects
3. Software Development Process
4. Software Estimation
5. The Contents of the Software Project Plan
6. Software Configuration Management
7. Software Quality Management
8. Software Metrics
9. Risk Management
10. Software Outsourcing Management
11. Human Resources Management and Team Building
12. Software Intellectual Property Management

The spiral model starts small and gradually expands as the project becomes better defined and gains stability.

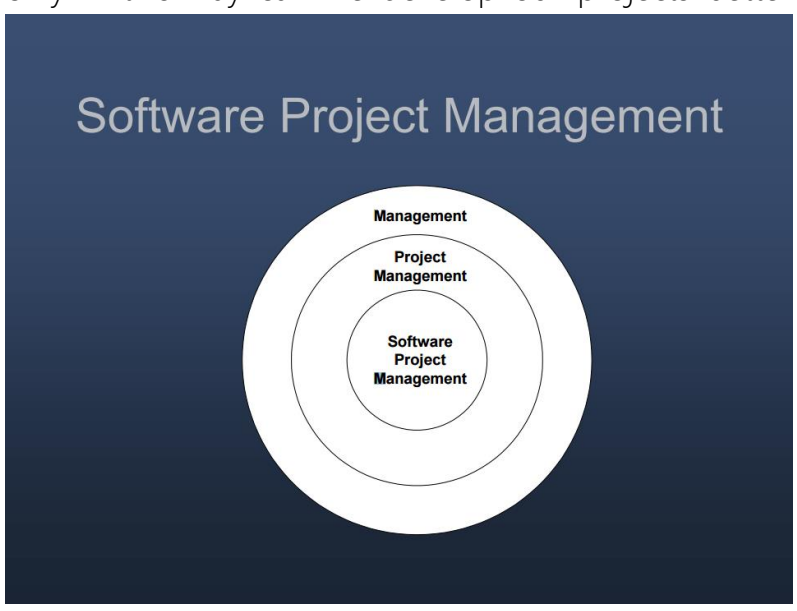


Among the most common factors for software projects' failure:

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements
- Poor reporting of the project's status
- Unmanaged risks
- Poor communication among customers, developers, and users
- Use of immature technology
- Inability to handle the project's complexity
- Sloppy development practices
- Poor project management
- Stakeholder politics
- Commercial pressures

5 Conclusion

Software project management is the discipline of software engineering and project management, and it is the application of the principle and method of project management in the field of software engineering. And we should pay attention to cultivate this awareness, only in this way can we develop our projects better



References

[1]Baidu, Software Project Management

[2]*Stellman, Andrew; Greene, Jennifer (2005).* Applied Software Project Management. *O'Reilly Media.* ISBN 978-0-596-00948-9.

[3]"Why Software Fails", in IEEE Spectrum