Software Project Management

Name Cheng XuQiang number 13211113
Zhejiang Normal University
Software Project Management

1 Introduction

With the continuous improvement of the performance of computer hardware and the continuous decline in prices, its applications are constantly expanding. It is hoped that more and more areas in the field of more and more difficult problems to the computer to solve. Corresponding to the size and complexity of the software and the Japanese fear. The emergence of the software crisis in 1960s was due to the complexity of the system beyond what people could control at the time. One of the most important gains from the software crisis is that it is essential to develop a software system with a certain size and complexity and to write a simple program. Simple program design may be regarded as individual works of art creation, but large, complex software systems development is a project, must according to the production and management of engineering principles and methods of software organization, must go through "plan, analysis, design, implementation, test and operation maintenance of a series of software life cycle stages".

2 Overview

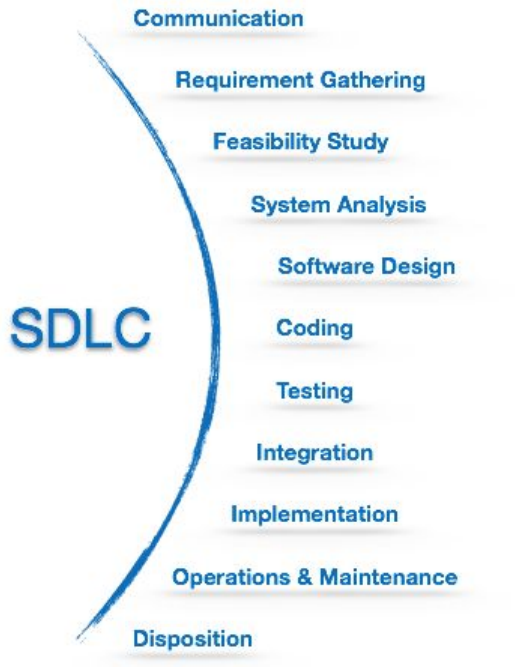
Software development of Xing goal is to construct a new software product or improve an old software products, it is a: the transformation from the users' needs for the needs of the software system of collection, can effectively develop high quality, reliable software system provides guidelines, and can reduce the risk.

In order to achieve the goal, need a can be used to guide customers, users, developers and project manager, etc. participants to all project related personnel can understand and complete their role in the project. A software development process but also with the passage of time and continuous evolution, each role in the process of evolution, are able to adjust themselves in a timely manner to adapt to the "technology, tools, personnel, organizational model" of the change. Specific say: process should be established at the time can use the practical technology on the basis of; process should be tools and development at the same time, process skills to achieve should be understood by existing staff development; process establishment should make the process suitable for the current situation.

Software development process through several years to reach a stable and mature level, to be able to withstand the strict test of the development of commercial products, and to maintain its use at the appropriate level of risk.

3 SDLC

SDLC provides a series of steps to be followed to design and develop a software product efficiently.SDLC framework includes the following steps:



3.1 Communication

This is the first step where the user initiates the request for a desired software product. He contacts the service provider and tries to negotiate the terms. He submits his request to the service providing organization in writing.

3.2 Requirement Gathering

This step onwards the software development team works to carry on the project. The team holds discussions with various stakeholders from problem domain and tries to bring out as much information as possible on their requirements. The requirements are contemplated and segregated into user requirements, system requirements and functional requirements. The requirements are collected using a number of practices as given -

- studying the existing or obsolete system and software,
- conducting interviews of users and developers,
- referring to the database or
- collecting answers from the questionnaires.

3.3 Feasibility Study

After requirement gathering, the team comes up with a rough plan of software process. At this step the team analyzes if a software can be made to fulfill all requirements of the user and if there is any possibility of software being no more useful. It is found out, if the project is financially, practically and technologically feasible for the organization to take up. There are many algorithms available, which help the developers to conclude the feasibility of a software project.

3.4 System Analysis

At this step the developers decide a roadmap of their plan and try to bring up the best software model suitable for the project. System analysis includes Understanding of software product limitations, learning system related problems or changes to be done in existing systems beforehand, identifying and addressing the impact of project on organization and personnel etc. The project team analyzes the scope of the project and plans the schedule and resources accordingly.

3.5 Software Design

Next step is to bring down whole knowledge of requirements and analysis

on the desk and design the software product. The inputs from users and information gathered in requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; logical design and physical design. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams and in some cases pseudo codes.

3.6 Coding

This step is also known as programming phase. The implementation of software design starts in terms of writing program code in the suitable programming language and developing error-free executable programs efficiently.

3.7 Testing

An estimate says that 50% of whole software development process should be tested. Errors may ruin the software from critical level to its own removal. Software testing is done while coding by the developers and thorough testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, in-house testing and testing the product at user's end. Early discovery of errors and their remedy is the key to reliable software.

3.8 Integration

Software may need to be integrated with the libraries, databases and other program(s). This stage of SDLC is involved in the integration of software with outer world entities.

3.9 Implementation

This means installing the software on user machines. At times, software needs post-installation configurations at user end. Software is tested for portability and adaptability and integration related issues are solved during implementation.

3.10 Operation and Maintenance

This phase confirms the software operation in terms of more efficiency and less errors. If required, the users are trained on, or aided with the documentation on how to operate the software and how to keep the software operational. The software is maintained timely by updating the code according to the changes taking place in user end environment or technology. This phase may face challenges from hidden bugs and real-world unidentified problems.

4 Conclusion

As time elapses, the software may decline on the performance front. It may go completely obsolete or may need intense upgradation. Hence a pressing need to eliminate a major portion of the system arises. This phase includes archiving data and required software components, closing down the system, planning disposition activity and terminating system at appropriate end-of-system time.

Acknowledgments

I would like to thank my classmate and reviewers for taking time out of their valuable schedule to help make this report more concise and clearer.

References

1.The Jaoo1.k~on I. unified software development process [M]. Beijing: Mechanical Industry Press.2002