

# The Software Engineering Analysis in Computer Science and Technology

Zhou Shanguo

Jiangxi College Of Foreign Studies

**Abstract**—Computer software engineering has an irreplaceable role in the information age, and it also has important influence to promote the development of computer technology. The project development of computer software engineering has a certain complexity and contains relatively much content. As a result, the management for computer software engineering is also crucial. In the unified organization of colleges and universities computer science and technology teaching steering committee of the Ministry of Education, the author puts forward the teaching implementation plan of the course according to the cultivate goal and the key problems of software engineering course. This paper introduces the overall framework of the implementation plan, combined with different types of computer talent cultivation, it expounds the emphasis of the curriculum implementation and builds the practical teaching plan from two aspects of course experiment and comprehensive training.

**Key words:** *Software engineering; Computer science and technology; The feasibility; Demand analysis*

## I. INTRODUCTION

"Software engineering" is a basic specialized core curriculum of computer science and technology and software engineering. It is about the basic principle and basic methods of software engineering and focuses on the analysis and solutions of software development problems. It makes students understand the design and application of software in large systems from the perspective of engineering, and masters the basic theory and method of software engineering, then uses the knowledge to the analysis, design, coding and testing activities of software development. It also cultivates the good team cooperation ability and practice ability and has a certain analysis and problem solving ability and innovation ability.

"Software engineering" is a basic course of paying equal attention to theory and practice, so there is a big difficulty in teaching implementation. The existing software engineering course has the disadvantages of not unified system, not clear position, not being divorced knowledge and reality, complex and abstract curriculum content and weak teaching practice. It will be boring and hard to understand to simply teach the abstract theoretical knowledge for the undergraduates who are lack of practical experience of software development. Especially along with the development of software engineering, traditional teaching contents have been far behind the current theory and technology and cannot meet the requirements of modern software engineering talents cultivation. The teaching goal of this course is to make students understand and master the methods, technology and ideas of modern software engineering, and have the professional abilities to become a software engineer. Therefore, how to make the teaching contents be advanced, teaching be vivid and real and practical effect be good have always been a difficult problem which is needed solving in software engineering teaching.

This paper introduces the overall framework of the implementation plan, combined with different types of computer talents cultivation, it also expounds the emphasis of the curriculum implementation and builds the practical teaching plan from two aspects of course experiment and comprehensive training.

## II. THE FEASIBILITY ANALYSIS OF SOFTWARE ENGINEERING

The purpose of the feasibility study is to put forward the problems which are needed to be solved but do not need to solve temporarily. Since then, stand in the angle of the efficiency, we should design it to get the best effect with the least man power and resource, after which the

follow-up of software development can be successfully completed. In the present situation, developing software needs to get certain conditions, the analysis is as follows:

#### A. Economic feasibility

Economic feasibility study mainly contains cost-benefit analysis. For example, in colleges or universities, if the books management has not yet entered the stage of computer management, in order to complete the business of "buy-borrow- return", a large number of staff must be employed.

#### B. Technical feasibility

Technical feasibility study is the most difficult and the most important in the process of system development. If colleges or universities are willing to develop the library management system, it is necessary to consider that whether there is available system development personnel. If not, agent is needed to finish the development. And the staff of colleges or universities shall learn the later maintenance.

#### C. Operational feasibility

Operational feasibility can determine whether the operation mode of new system is feasible. For example,

our computer, network, maintenance personnel must be in place, or the system won't be able to normally operate for 24 hours a day.

#### D. Development plan feasibility

"Brainstorming" can be used to enlighten all kinds of feasible development plans, which will be selected after team discussion and the consent of the competent.

### III. THE ENGINEERING PRACTICE OF DEVELOPMENT REQUIREMENTS

In engineering practice, with reference to RD process area of CMMI model, combining with the enterprise actual development situation, when defining the product life cycle, the needs analysis shall be defined in two stages, namely users demand analysis stage and scheme design stage. The results are respectively software technical specification and the analysis of the software requirements specification, the corresponding position of which in V mode of software development is shown as the dotted boxes in Fig . 1.

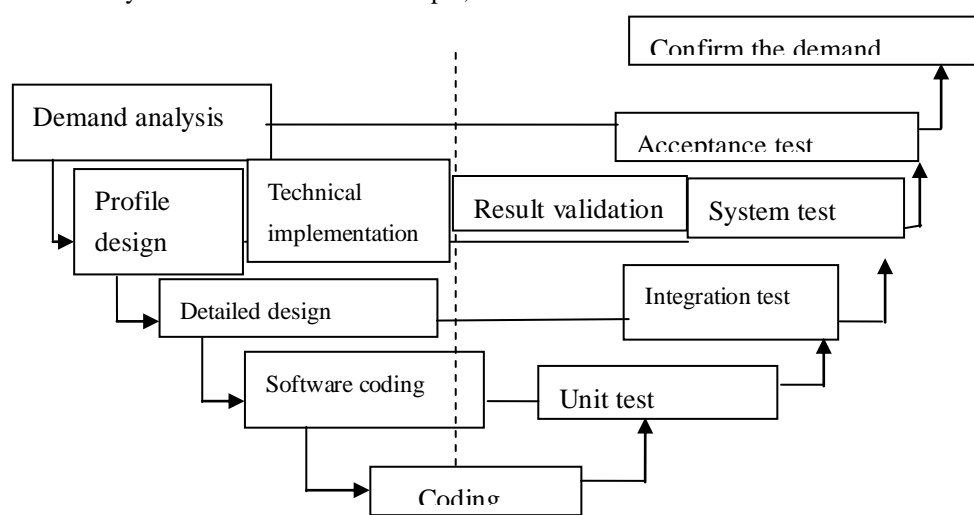


Figure 1.V mode

Software technical specification is to analyze users demands of project products according to the input user requirement documents.

Through the existing product prototype, demonstrate and explain demands to users in order to obtain the confirmation of demands; establish demand traceability

matrix to start on vertical and horizontal tracking to the decomposition of users' demands.

Technical specification includes the software function requirements, the performance requirements, the interface requirements, RAMS requirements, conditions of use, verification test requirements, compatibility requirements, service training requirements, etc.

#### IV. COMPREHENSIVE TRAINING OF SOFTWARE PROJECT

Software engineering is a comprehensive practice course without much teaching content. It mainly focuses on the practice of a complete project development. It is needed to build a typical software engineering environment of enterprises, including development process model, series document template, software coding standard, stage evaluation standard, software engineering tools and other infrastructure. Fig .2 shows the process planning and activities of the software engineering project

comprehensive training. In the process of training, it would be better to have a real customer representative to put forward real project requirements. The development process will be through multiple iterations of a software system and will adopt the standard project management to manage the project itself and to pay attention to quality problems. Teachers and support staff should have sufficient practical experience to play a guiding role in the project development.

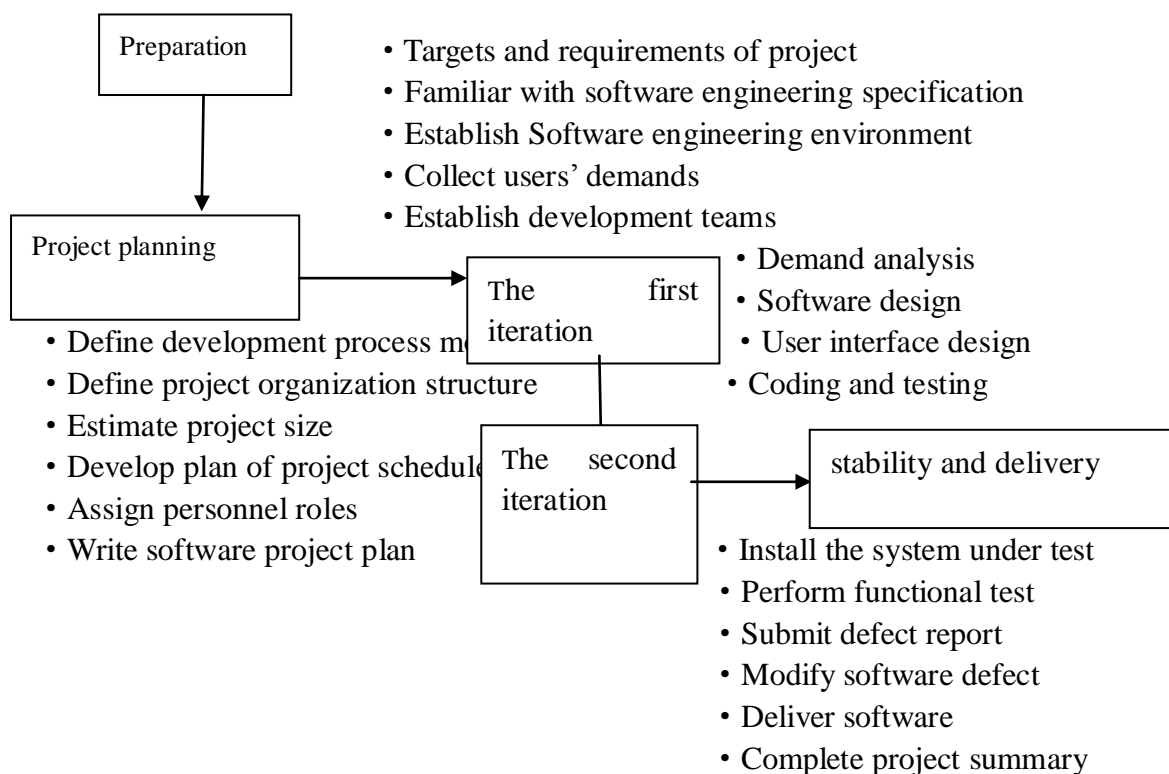


Figure 2. The comprehensive training contents of software engineering project

The assessment of the course is composed by stage evaluation summary, in which the project planning stage accounts for 10%, the first iteration accounts for 30%, the second iteration accounts for 30%, stability and delivery stage accounts for 20%, weekly inspection and report accounts for 5% and the management and collaboration of the development team accounts for 5%.

#### V. CONCLUSION

In today's era of software engineering, each project involves many manpower and material resources, management problems become increasingly prominent. The computer software project management is of vital

importance for software enterprises, and has gradually become the core competitiveness of software enterprises. Project management is an important work in software production activities, since the most software project failures are not because of technical reasons, but of the failure of the administration. The above countermeasures should be taken to improve project management level. This paper tries to be vivid and reasonable on teaching content with discipline development; pays attention to case teaching on teaching methods and arouses the enthusiasm of students; creates a software engineering environment which is suitable for software engineering course experiment projects and pays attention to the

selection of the topic and the management of the whole process; builds practical teaching plan from two aspects of course experiment and comprehensive training.

#### REFERENCES

- [1] Pressman, Roger S. Software engineering : a practitioner's approach. McGraw-Hill, 1997:45-55.
- [2] Chung, Lawrence. "On Non-Functional Requirements in Software Engineering." Lecture Notes in Computer Science 5600(2009):363-379.
- [3] Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). Experimentation in software engineering : an introduction. Kluwer International, 9(2), 171-180.
- [4] Humphrey, Humphrey Watts S. "A Discipline for Software Engineering, ISBN 0-201-54610-8." Sei 30.3(1995):1-3.
- [5] Kitchenham, B.A, Pfleeger S L, Pickard L M, et al. Preliminary guidelines for empirical research in software engineering[J]. Software Engineering IEEE Transactions on, 2002, 28(8):721-734.
- [6] Wohlin, Claes, et al. "Experimentation in Software Engineering." IEEE Trans Soft Eng se-12.7(2009):1-16.
- [7] Garlan D, Shaw M. An introduction to software architecture: advances in software engineering and knowledge engineering[J]. Advances in Software Engineering & Knowledge Engineering, 1993, (44Part2):1-39.
- [8] Jazayeri, Mehdi, and W. Tu. "Fundamentals of Software Engineering." Lecture Notes in Computer Science 9392(1991):2433-2434.
- [9] Kitchenham, Barbara /A/., et al. "Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng." IEEE Transactions on Software Engineering 28.8(2002):721-734.
- [10] Basili, Victor R., and D. M. Weiss. "A Methodology for Collecting Valid Software Engineering Data.." Software Engineering IEEE Transactions on se-10.6(1984):728-738.
- [11] Society., Ieee Computer. "IEEE Transactions on Software Engineering." Software Engineering, IEEE Transactions on 30.2(1975):0\_2.
- [12] Kemmerer, Richard A. "IEEE Transactions on Software Engineering." Software Engineering IEEE Transactions on 30.4(1983):0\_1.