

Software Engineering Implementation Research

Chen Wen-feng

NanChang Institute of Science & Technology

Abstract—Software engineering implementation is to apply software engineering theory for software development organizations in order to improve the product development time (T), quality (Q), cost (C), service (S) of software development organization. The most important is the engineering software development process, namely the establishment of a process with working order, limited step, and in line with the rules throughout the software development organization, to improve software quality and shorten the software development cycle. Although software engineering research has been conducted for decades, the level of engineering software development organization in China now is still relatively low, which has become a major problem restricting the development of China's software industry, but also an important factor affecting the internationalization of China's software industry. In this context, this paper combines several years of experience and engineering exploration experience in project development to study the software engineering implementation problem, principles, procedures and methods.

Keywords—*software engineering, software engineering implementation, SW-CMM, SW-IDS*

I. INTRODUCTION

Software engineering is the discipline to guide the development and maintenance of computer software. Software engineering research, has been developed from solving the software crisis with technical measures (techniques, methods, and tools) to comprehensive study on two aspects of project management and development technology, so that software engineering has become a systematic and integrated engineering discipline.

A. *The most important developments in software engineering theory are:*

The research in this area has sustained for several decades. In terms of software architecture, the proposed hierarchical model makes software structure more reasonable. The development technology also changed from process-oriented to module-oriented, and the recent research focuses on component-oriented technology and object-oriented technology.

B. *Management aspects*

Since the 1980s, it was discovered that although software development techniques, methods and tools were developing continuously, the key software projects funded by the government and undertaken by the software organizations are still not delivered on time, additional funds again. Quality and performance of the delivered software products is difficult to meet the desired requirements. For many software organizations, large-scale software development projects mean potential troubles. People realize that the main problems of software development today gradually become management issues. Carnegie Mellon University Software Engineering Institute (SEI) that was established in 1984 has been working on how to solve software problems from a management point of crisis, and achieved fruitful results.

II. SOFTWARE DEVELOPMENT ARCHITECTURE

Software engineering is the engineering discipline to guide the development and maintenance of computer software. The software development architecture consists of four levels, as shown in Fig .1, in which focusing on quality (the quality herein is a broad concept, it can be

measured by customer satisfaction) is the fundamental starting point.

Tool
Method
Software process
Focus on quality

Figure 1. Software Development Architecture

Software engineering refers to a set of activities, methods, practices and transformation to develop and maintain software and related products (such as project plans, design documents, code, test cases, user manuals, etc.). Software process is the foundation of the whole system, also the adhesive of various methods. The research for the process eventually led software production from the workshop-style toward industrialization, making software develop from chaos state to be clear. ISO9000 series (2000) and SW-CMM is the standard about the process, describing the needs in process of enterprises, such as software requirements management, design process, document management processes. The basic element of the process is activity, the basic frame 2 as shown in Fig .2. The work product in the figure refers to the document or program.

The method includes management methods and techniques, to provide guidance and code on how to complete the process activities, such as the reusing technology, concurrent engineering technology in management, and object-oriented analysis technology, design technology, implementation technology and test technology in technical aspects.

The overall architecture reflects the viewpoint of quality as the center. Focusing on quality is the fundamental starting point and final goal of the process. From Fig .2, we can see the process can reflect the integration of different methods and tools. Therefore, in the software process without discipline and with chaos, even if there are good tools and methods, it is difficult to benefit from it.

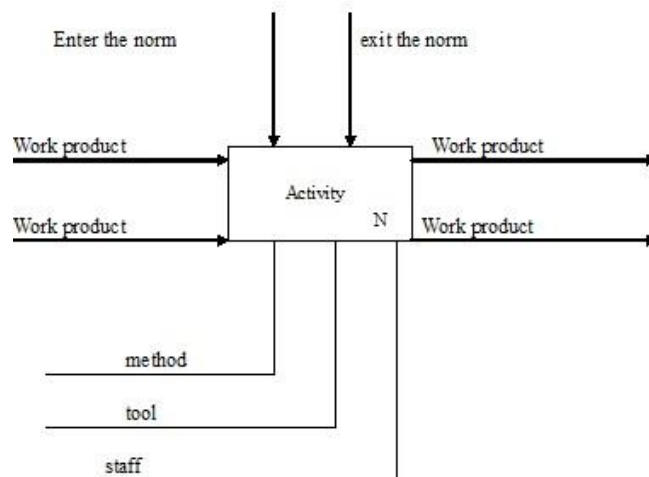


Figure 2. overall architecture of software engineering

III. SOFTWARE ENGINEERING

The software engineering in this paper can be understood as: apply software engineering techniques to the entire life cycle of software product, to achieve integration of people (organization, management), technology, process to improve product development time (T), quality (Q), cost (C), service (S) of software development organizations. Software engineering work refers to the activities that need to be developed in software development organization's software engineering process. Software engineering project means the established project of software development organizations to ensure the software engineering work.

The entire software engineering work involves the overall planning and implementation. Software engineering work of organizations is planned integrally based on software development architecture, including determining the organization's quality objective, process reengineering, determining methods and tools to carry out the process. The implementation includes processes remake, technical level improvement, and personnel quality improvement. Whatever the overall planning or implementation, process is the foundation and core, so we can say process planning and implementation is the key to success of the entire software engineering work.

Software engineering process refers to process reengineering, the introduction and improvement of technologies and various management and technical issues, so it is complex socio-technical engineering. Recently, China promotes ISO9000 certification and SW-CMM certification vigorously in the software development organizations, and there is some organizations has achieved certification with the help of certified experts, but among them, there are few organizations that can play a positive role in improving TQCS. There are still many organizations still in the workshop stage encountering many obstacles and difficulties. To change this situation, practical research under the guidance of software engineering theory especially needs to strengthen.

IV. SOFTWARE CAPABILITY MATURITY MODEL

Software Capability Maturity Model (SW-CMM, Software Capability Maturity Model) is a framework to describe the key elements of an effective software process. SW-CMM describe an evolution and improvement way from expedient immature process to a mature and disciplined process.

SW-CMM is the key practice of software development and planning, software product engineering and management related to maintenance. Following these key practices, the organization ability in achieving cost, schedule, functionality and product quality goals can be improved. The key practice here is the description of the effective implementation of key aspects of the process and the infrastructure and activities with the largest contribution to institutionalization.

SW-CMM establishes a scale, and with the scale, it is possible to determine the maturity of organization software process with a reproducible manner. Organizations can also use SW-CMM to plan software process improvement. SW-CMM mainly describes quality concerning and process, two levels in software development architecture. The whole model is divided into five maturity levels with phased structure, and each level defines the software process capability and key aspects of the process. Each key process identifies the objective and a string of related activities and when these activities are completed, the identified objectives of this process can be reached.

V. CONCLUSION

Software engineering work is a systematic project, involving all aspects of software development organizations, including technical issues, management issues and people issues. In this paper, the author studies the problems in software engineering based on the understanding of software engineering theory, and proposes the solution. The author believes that software organizations should improve their level of engineering by means of information. This paper presents software engineering integrated development system SW-IDS,

which is for this purpose. In the SW-IDS, the activities of software organization are orderly and standardized, improving the project efficiency. At the same time, many advanced management techniques are used, such as concurrent engineering, process reengineering, virtual design, virtual manufacturing, so the integration of enterprises becomes possible. Based on experience of CIMS engineering practice and software engineering experience, we propose the methods and procedures to achieve engineering.

REFERENCES

- [1] Mark C. Paulk, How to Measure Software Process Maturity, Software Engineering Institute
- [2] Mark C. Paulk, Capability Maturity Model for software, version 1.1, software Engineering Institute, February 1993
- [3] Mark C. Paulk, Key Practices of the Capability Maturity Models, version 1.1, Software Engineering Institute, February 1993
- [4] Williams L, Upchurch R. Extreme programming for software engineering education[C]. Frontiers in Education Conference, 2001. 31st Annual. IEEE, 2001: T2D-12-17 vol.1.
- [5] Cybulski J L, Reed K. A Hypertext Based Software-Engineering Environment[J]. IEEE Software, 1992, 9(2):62-68.
- [6] Berzins V, Luqi L. Software Engineering with Abstractions[J]. Boehm B Software Engineering Economics, 1990.
- [7] Pfleeger S L. Experimental Design and Analysis in Software Engineering: Types of Experimental Design[J]. Acm Sigsoft Software Engineering Notes, 1995, 20:14-16.
- [8] Calvary G, Coutaz J, Dâassi O, et al. Towards A New Generation Of Widgets For Supporting Software Plasticity: The "Comet"[J]. Engineering Human Computer Interaction & Interactive Systems, 2004:306-324.
- [9] Gegick M, Williams L. Matching attack patterns to security vulnerabilities in software-intensive system designs.[J]. Acm Sigsoft Software Engineering Notes, 2005, 30:1--7.
- [10] Tarr P, Clarke L A. Pleiades: An Object Management System for Software Engineering Environments[J]. Proceedings of the First Acm Sigsoft Symposium on the Foundations of Software Engineering, 1993, 18:56--70.
- [11] Beck L L, Perkins T E. A Survey of Software Engineering Practice: Tools, Methods, and Results[J]. IEEE Transactions on Software Engineering, 1983, 9(5):541-561.
- [12] Raccoon L B S. Fifty years of progress in software engineering[J]. Acm Sigsoft Software Engineering Notes, 1997, 22(4267):88-104.