# Software Maintenance Process Model and Contrastive Analysis

Ren Yongchang
College of Information Science and Technology
Bohai University
Jinzhou, P.R. China
rycryc@sina.com

Liu Zhongjing
Department of Computer Science
Xingtai Polytechnic College
Xingtai, P.R. China
635909530@qq.com

Xing Tao
Beijing Research Center of Urban Systems Engineering
Beijing, P.R. China
xingtao@mail.tsinghua.edu.cn

Chen Xiaoji
Department of Computer Science
Xingtai Polytechnic College
Xingtai, P.R. China
shandian820414@sohu.com

***Abstract*—Software maintenance is an important stage of software life cycle, according to the problems of software maintenance process model, research software maintenance process model, focus on software after delivery to retire between the software maintenance activities. Firstly, descript of the eight software maintenance process; then, research software maintenance process model of the structure and content, including the rapid change model, Boehm model, IEEE model, iterative enhancement four models; finally, analyze to the four models of software maintenance process, provide the basis for the selection of the model. This study's content includes the improvement of software maintenance process, guide maintenance activities, improve the quality of software maintenance, ensure the normal application software, which has important theoretical and practical significance.**

***Keywords-software maintenance; maintenance process; process model; comparative analysis***

## I. INTRODUCTION

Software maintenance refers to "a software product after delivery to be modified to correct faults, improve performance and other property, or let the product adapting to changing environment [1]." Software maintenance's workload is very large, although in different areas of application maintenance costs vary widely, but on average, large-scale software maintenance costs up about 4 times to development costs. Many software development organizations in foreign countries, use 60 percent of the manpower for the maintenance of existing software, and with the increasing number of software and service life extension, the percentage is still rising [2]. With a wide range of software applications, increasing the problems exposed, software maintenance is becoming increasingly important, software maintenance is a time-consuming, energy consuming work [3].

Software maintenance as a stage of software life cycle, the model relative to the software development model is concerned, hasn't been fully understood and applied. Development model process compares to maintenance process model, the early stages of the work requires greater, focus is also different, and less follow-up stage of development effort [4]. Researching Software maintenance process model, helping analyze the activities during software maintenance, standard software maintenance process and improve the software maintenance productivity, extend the software life cycle.

## II. SOFTWARE MAINTENANCE PROCESS

Software maintenance process show in figure 1.

- Preparation. Adequate preparation is a good start to maintenance. Including the designation of maintenance personnel, establish smooth communication channels to facilitate the maintenance, training, preparation and approval of "software maintenance plan" and so on.
- Request. Software maintenance begins with a request to change the system, usually a request by the user, on-site maintenance engineers or developers to issue report card form.
- Need analysis. Responsible officer explain to human analysts, who analysis system change requests which means issues report card, including the issue positioning, issues related to the specific product and the corresponding change of scale, give a specific solution to issues related to documentation, test programs and strategies proposed and ultimately the formation of the problem analysis.

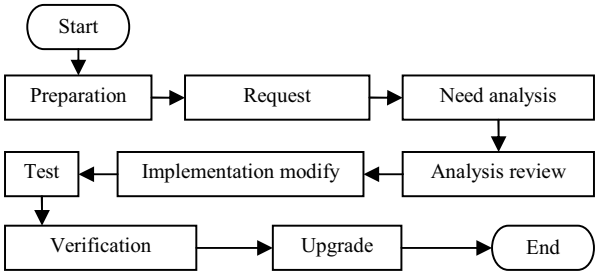


Figure 1. Software maintenance process

978-0-7695-4523-3/11 $26.00 © 2011 IEEE
DOI 10.1109/ICIII.2011.324

- Analysis review. The process is an important means to ensure maintenance of quality, can make early detection of problems and reduce the risk of maintenance problems later discovered. Analysts question the degree of difficulty according to the problem involves the size of the change and the problem-solving ability to grasp and other factors, to decide which assessment method.
- Implementation modify. According to one report to address personnel issues, problem analysis report and product manuals related to the modification of the product. It modifies the software, must first understand the program, help complete the requirements; also modify the program to understand the possible side effects, so that when changes in the program note; final report on the formation of a single software modifications.
- Test. Software is modified, the analytical report based on testers in the test program for testing. It's easy to introduce new errors and change, therefore the regression test, reduce side effects caused by change.
- Verification. After testing software changes before submission system upgrade, to go through the validation phase, the reviewers question whether the correct solution for the assessment. Verify the initiator for the managers, the purpose is to ensure the quality of software maintenance, can refer to the analysis phase of the evaluation methods for assessment.
- Upgrade. Software verification is complete in internal company, need publishing to user, upgrade system. Upgrade process from developers, engineers and users together.

## III. Software Maintenance Process Model

Software maintenance process model is an abstract representation of the evolution of software to help analyze activities during software maintenance. Which use kind of maintenance model, should be aware of the characteristics of various models and, based on preservation of the environment to decide. The following analysis of several common models [3].

### *A. Quickly Modify Model*

Quickly modify model that the maintenance process is a "fire fighting" approach, which is the temporary custom software maintenance method, software problem should be solved as soon as possible, shouldn't analyze long-term effects on the implementation of changes. Usually don't analyze the code to modify the structure of the ripple effect of the impact, even if the analysis is also very little written documentation. Quickly modify the model structure show in figure 2.

In the right environment, this model is very effective. For example, if the system is developed and maintained by one person, this person is very familiar with the system, has the ability in the absence of detailed documentation in the case management system, whether changes can be made to determine how to modify and maintain work quickly and economic.

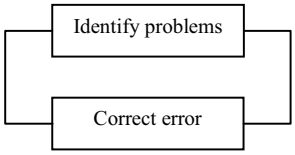


Figure 2. Quickly modify model

There are many customers in the business environment, this approach isn't reliable, but there are still many institutions using this model is due to software maintenance are time and resource constraints. For example, customers request the correction of an error, but are not willing to wait for software companies to change the cumbersome process and risk analysis.

If the software rely on quick changes with a long time, it will accumulate a lot of problems, the software will become increasingly difficult to maintain, maintenance costs will increase, it will lose use of rapid change in the initial stages of the model to get any advantage. To address this problem, using the strategy is to quickly modify the model, into another, more sophisticated models, rapid changes in external pressure as an emergency to change the way modifications are completed, according to the model requires some fine measures.

### *B. Boehm Model*

Dr. Barry W. Boehm bases on economic models and principle, proposes maintenance process model. Boehm's theory is models and principles of economics can not only improve maintenance productivity, but also helps to understand the maintenance process. Model structure show

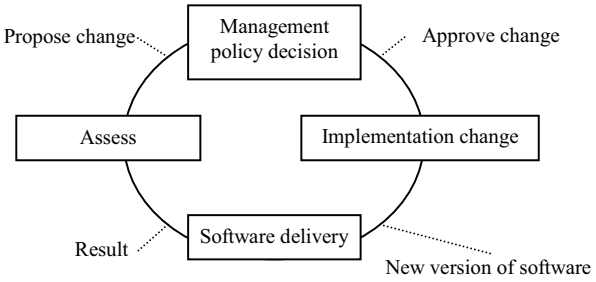


Figure 3. Boehm model

in figure 3. Model maintenance process is divided into management decision-making to achieve change, software delivery and evaluation of four stages, expressed as a closed loop to maintain the process by promoting the maintenance management decision-making process. In management decision-making stage, use of specific strategies, and proposed a set of changes for cost-effective assessment to determine a set of approved changes and the implementation of changes to a dedicated budget.

From the functional point of view of production, the model reflects the economics of investment and the relationship between earnings, reflecting a typical phases:

- Investment stage. This is a low input of resources and low-income stage, corresponding to an emergency there is a strong requirement to modify and enhance new software product;
- High-return stage. Institutions through software products have been growing returns, the initial problem is resolved. At this stage, resources and efficiency to the document, the agency's rapid growth in the accumulation of benefits;
- Effective reduction stage. At a certain point in time, the accumulation of effective growth rate gradually slows down. At the peak of the effectiveness of the product, to change to become less and less economic stage.

Boehm model focuses on the management decision-making, according to the approved changes to the implementation of changes to maintenance activities in the balance between investment and benefits from the perspective of economic interest to drive the software maintenance process. Based on this process, the organization can develop a reasonable maintenance strategy, maintenance efficiency to meet the organization to make decisions.

### *C. IEEE Model*

With the development of software industry, is increasingly recognized for the importance of software maintenance standards. Therefore, IEEE Computer Society software engineering standards branch, issued a "IEEE Software Maintenance Standards" (IEEE 1219-1993), details the activities of management and implementation of the iterative process of software maintenance, software maintenance, including input, processing, control and output and so on. The standard that should be in the planning of software development when the software maintenance plan. Model show in figure 4.

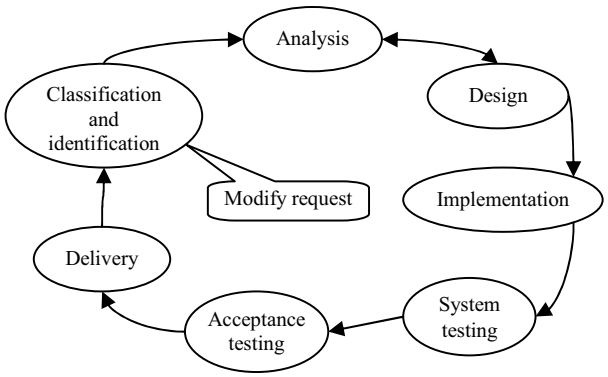


Figure 4. IEEE model

Each stage of IEEE maintenance model are described below:

- Classification and identification. Software maintenance starts in by the user, developer or manager's request to modify the software, and modifies the request to submit a single form. Change request can be any maintenance categories (corrective maintenance, adaptive maintenance, improve maintenance, preventive maintenance), maintained by the agency to determine what type, divided into the appropriate maintenance category, and determine the processing priority. Each application is assigned a unique number, the request content input to the database system for tracking. Collection, review and measure the indicator from the stage.
- Analysis. First, maintenance of the feasibility analysis, and detailed analysis. Feasibility analysis of major changes to the software to determine the impact of possible solutions and costs, etc.; detailed analysis of the major changes is that a full requirements specification, identify the need to change the elements (modules), proposed testing program or policy development implementation plan. Finally, the Configuration Control Board (CCB), to review and decide whether to proceed with work.
- Design. Summarizes all the information in order to review and change the design for the software, including project documentation, the results of the analysis phase, source code, knowledge base and other information. Baseline design analysis phase should be updated, update the test plan, revised the results of detailed analysis to verify the maintenance requirements.
- Implementation. Formulate plans for change to change. Include the following process: coding and unit testing, integration, risk analysis, test preparation review and update documentation.
- System testing. The interface between main test program to ensure that the system meets the original requirements, as well as additional changes to demand. While regression testing to ensure that does not introduce new errors.
- Acceptance testing. In complete system integration, by the user or third party. Mainly to complete the following tasks: report test results, functional configuration audit (to determine the system functions to meet the demand), the establishment of a new version of the software (or baseline), to prepare the final version of the software documentation (including system documentation and user documentation), etc.
- Delivery. The revised system give to user who install and run it. Physical Configuration Audit should be carried out, the backup documentation, installation and training work. After delivery, the system put into use.

Compared with the previous model, IEEE model details the process of software maintenance activities, as standard, can be applied to all software maintenance process. However, different software, due to different characteristics, maintenance procedures will be different, IEEE is a large and

comprehensive model specifications for different software, the process should be cut, while the process of refining some of the tasks.

### D. Iterative Enhancement Model

Initially the model is put forward as a development model, because the software developers usually can not fully understand the requirements, can not build a perfect system, so it is suitable for maintenance. Made the basis of the model: the software life cycle of software changes implemented, is an iterative process, and to iteratively enhanced software system.

Model requires a complete document, as the beginning of each round of iteration, is actually three stages of the cycle, show in figure 5.

According to the impact of the change request document, the first change of each phase (requirements, design, coding, testing and analysis of the document), full documentation of this change in the spread, and re-design systems.

According to safeguard the environment to find quick solutions. Usually use the "fastest" solution will lead to many problems, iterative model, other models of assimilation itself, it can be in a structured environment, the integrated model of rapid change, rapid change, identify problems, and Diego the next round generation to specifically address these issues.

## IV. Maintenance Model Analysis

- Quick change model applies only to "fire" of the situation, cann't guarantee that the software maintenance process reliability requirements;
- Boehm model is applicable to the management of maintenance organization, is driven from the perspective of economic interests, the process of software maintenance, focusing on the provision approving the change of strategy, change implementation process for the few requirements. The same process can not guarantee the high reliability of software maintenance requirements, but also for software maintenance, the focus on maintaining quality and reliability, rather than approving the change of strategy;
- IEEE model is a large and specifications for all software maintenance process. Through system testing and acceptance testing to ensure maintenance of quality;
- Iterative model of software changes enhance the demand is not sufficient, by constantly improving the software iteration.

Apart from the above description of several models, there are Osborne model, reuse-oriented model. Different models of a different focus, some attention to economic issues, and some attention to the product, and some attention to the process. All models have advantages and disadvantages, no single model is applicable to all situations, and often the model together is the best solution. Maintenance model to determine the future, in order to achieve effective maintenance management, should also be in-depth understanding of software process and related concepts.

## V. Conclusion

Software maintenance process begins after the delivery of software products to the end when the software products is retired. However, software maintenance issues on the software life cycle should be developed in stages to be considered throughout the software life cycle, software maintenance and software development are closely connected. Software maintenance problems in the software life cycle consider the later, maintenance costs will be greater, the difficulty will be greater, result in expensive software maintenance costs[9].Therefore, you should consider an early stage of software development software maintenance. The thesis process model for software maintenance research, focusing on software after delivery to retire between the software maintenance activities. With the acceleration of information technology, software products increasing, more and more important software maintenance, maintenance problems has become increasingly evident. In this paper, researching the software maintenance process model, has important theoretical and practical significance to improve the software maintenance process, guide maintenance activities, improve the quality of software maintenance to ensure the normal application software.

## Acknowledgment

This work is supported by Innovation Team Program of Beijing Academy of Science and Technology (IG201106N), Education department of liaoning province key laboratory fund project (2008S002) and Liaoning doctoral Foundations (20091034).

## References

[1] Z. Shi, "Software Engineering Standards Manual," China Standards Press, 2004.

[2] H.F. Zhang, "Introduction to Software Engineering (Fourth Edition)," Tsinghua University Press, 2003.

[3] Y.C. Ren, T. Xing, L. Wang, B.Y. Zhao, "Software Project," Jilin University Press, 2011.

[4] J.J. Ding, "Research on Metric-based Software Maintenance Process Management," Master's Northwest University, 2006.

[5] X.D. Zhu, X.W. Wang, "Software Maintenance Process Study Based on CMM," Computer Engineering and Applications, vol. 42, no. 29, pp. 66-69, 2005.

[6] X.W. Wang, X.D. Zhu, Y.G. Wang, "Software Maintenance Process Management Based on CMM and Tools Study," Microcomputer Information, vol. 21, no. 7-3, pp. 139-141, 2005.

[7] Q. Lin, "Research on software re-engineering," Computer Engineering and Design, vol. 27, no. 1, pp. 83-85, 2008.

[8] J. He, Q. Su, J. Gao, M.Y. Li, "Systematic Model of Software Maintenance," Application Research of Computer, vol. 22, no. 1, pp. 16-19, 2005.

[9] G.R. Wei, "Research of Maintenance Process of Core Telecommunication Software and Implementation of Maintenance Tools," Master's Beijing University of Posts and Telecommunications, 2007.