

# Software Maintenance: Challenges and Issues

Aakriti Gupta  
St. Xavier's College,  
JAIPUR - 302001, INDIA

Shreta Sharma  
St. Xavier's College,  
JAIPUR - 302001, INDIA  
shretasharma@hotmail.com

**Abstract**—*Maintenance plays an important role in the software development life cycle. A software project is delivered within estimated time only if all the phases of software development process are completed within estimated and primarily set up time. Various researchers have made substantial tools and techniques to achieve the quality of software maintenance phase. But at the same time, the field requires a future research work to enhance the quality of software and to reduce the challenges of maintenance phase. It has been estimated that there are more than 100 billion lines of code in production in the world. As much as 80% of it is unstructured, repaired and not well documented. Maintenance can relieve these problems. This paper describes the major activity and process of maintenance phase along with its key issues.*

**Keywords**—Software Development Life Cycle, Software Quality, Software Maintenance, Issues and Challenges in Maintenance Phase.

## I. INTRODUCTION

Software Development life cycle has several phases. The process of software development includes Requirements phase, Design, Implementation, Testing, and Maintenance. Maintenance is the last stage of the software development life cycle. The term “software maintenance” is used to understand the software engineering actions that take place during the progress of software. Software maintenance process is very compressed process and usually it comprises more than half of the development process again. Typically, the development of software takes 1 to 2 years, while maintenance phase spans 5 to 10 years.

When a company released a successful project to its client within fixed time, then the actual work of the maintenance begin. Many a time it has been seen that the cost of the maintenance exceeds the development cost of the project. Basically software maintenance phase keeps the software up to date with environment changes, correct the faults and improve the performance of software product after delivery.

A common observation of maintenance phase is that it purely includes fixing faults. However, previous research depicts that the majority, above 80%, of the maintenance effort is used for non- corrective activities [1]. Major issues of software maintenance are in arrangement with customer significances, cost estimation with some technical challenges and staffing etc.

The main aim of this paper is to highlight the major issues of maintenance phase. Beyond this Introduction on the background details, rest of the paper is organized as follows: Section II presents a brief overview of maintenance phase process whereas various maintenance challenges have been given in Section III. Finally, Conclusion and Future Work are reported in Section IV.

## II. MAINTENANCE PROCESS

Software maintenance is one of the major concerns of software development. Good maintenance process is very essential to maintain the quality of software. Several authors have proposed various process models for software maintenance. These models help to systematize maintenance into a string of associated activities, or phases, and define the array in which these phases are to be executed. Basically there are seven major phases in maintenance process, which are given as follows.

**A. Change Management**

This is the phase in which the user appeal for modification, a customer, a programmer, or a manager is assigned a maintenance category, precedence and an exclusive identifier. The phase also includes activities to establish whether to accept or reject the request and to assign it to a set of modifications scheduled for implementation Maintaining the Integrity of the Specifications.

**B. Analysis**

This phase arrange a base plan for design, execution test, and delivery. The main aim of analysis is to conclude the possibility of the requested change for arrangement and implementation of the change. Analysis is administered at two levels: feasibility analysis and detailed analysis. Feasibility analysis recognize substitute solutions and assesses their impacts and costs, whereas detailed analysis defines the necessities for the modification, devises a test policy, and develops an implementation plan.

**C. Design**

The alteration to the system is actually designed in this phase. This brings about all present system and documentation of projects, database and existing software and output of the analysis phase [8]. It aims to expand a revised logical and physical design for the change and to design the changes for all of the categories of maintenance.

**D. Implementation**

This phase includes the activities of coding and unit testing, assimilation of the customized code, integration and analysis, regression testing, and risk. The phase also includes a test-readiness review to asses' awareness for the system and regression testing.

**E. Regression/System Testing**

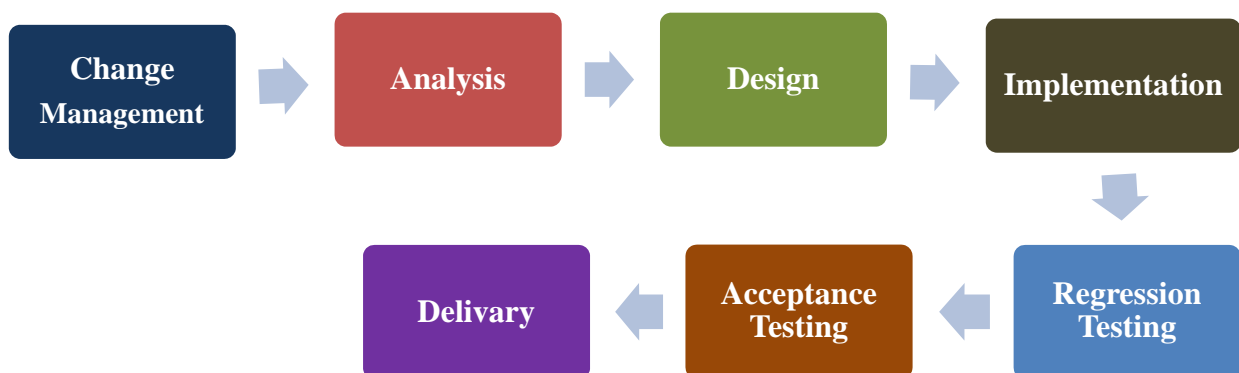
This is the phase in which the complete system is tested to make certain conformity to the new necessities plus the alterations. Additionally this phase includes regression testing to test functional and interface, which authenticate that no new faults have been added. Finally, this phase is accountable for verifying awareness for acceptance testing.

**F. Acceptance Testing**

This level of testing is apprehensive with the fully incorporated system and involves users, customers, or a third party nominated by the customer. Acceptance testing contains regression tests, interoperability tests and functional tests [4].

**G. Delivery**

This is the phase in which the customized systems is unrestricted for both operation and installation. It includes the activity of notifying the user community, performing installation and training[5].



Maintenance Phase Process

### III. ISSUES AND CHALLENGES

Most problems that are associated with software maintenance can be traced to deficiencies of the software development process. There are several technical and managerial problems encountered while maintaining software [2].

#### A. *Costs*

Various research studies proposed that software maintenance consumes 60% to 80% of cost in whole development life cycle; these surveys also report that maintenance costs are mainly due to enhancements, rather than corrections[6].

#### B. *Impact Analysis*

One of the most important challenges in software maintenance is to find out the effects of a proposed modification on the rest of the system. Impact analysis is the action of assessing the probable effects of a change with the plan of reducing sudden side effects. The task involves assessing the correctness of a projected modification and evaluating the risks related with its completion, plus the estimates of the effects on properties, energy and development.

#### C. *Corrective Changes:*

One of the major key issues is corrective changes because it is hard to find the correct place to do the changes. It can be difficult to recognize the code base. If the preliminary design is reduced a minute change might insist architecture changes that take a lot of time. If there has been a complete workaround of one problem then the next are even harder to crack. Design errors are tough to repair because it takes a lot of time and understanding of the entire code base and are linked to risks.

#### D. *Adaptive Changes*

Adaptive changes are frequently not easy due to deficiency of information about what the software is being modified to. The diverse facts of the new technology to adjust to be difficult to take hold of. Also impact analysis and discovering interfaces to the new things are difficult. Problems due to unbalanced preliminary design are a matter of concern.

#### E. *Program Comprehension*

Another key issue is program comprehension which involves that extensive amount of time should be expended by maintenance engineers to read and understand the code, the relevant documentation to have a better perspective on its logic, purpose and structure to maintain a part of software and to enhance the quality of software[6].

### IV. CONCLUSION

The paper provided an overview of maintenance phase process of software development by covering all the activity for the same. Additionally, Literature reveals several issues and challenges in maintenance phase, which result in poorly completed part of systems analysis. These issues experiential in maintenance have been shown several times to be a major reason of systems failure. This paper has attempted to present substantial issues and challenges in maintenance phase. Future work may be to bring up novel ideas/approaches to overcome these concerns along with strong validation results. It is evident from the aforementioned discussion that the root cause of each challenge is the cost estimation in this process. Incorporation of latest Artificial Intelligence (AI) techniques may limit such issue up to some extent and seems to provide some fruitful results..

### REFERENCES

- [1] Ann-Sofie Jansson. Software Maintenance and Process Improvement by CMMI. Examensarbete 20 p(1650-8319).pp.4-24.
- [2] Alkhatib, G. . The maintenance problem of application software: an empirical analysis. Journal of Software Maintenance: Research and practice 1,2 s.pp.83-104.
- [3] Herbsleb J., Carleton A., Rozum J., Siegel J., Zubrow D. Benefits of CMM-based software process improvement: Initial results. Technical report, CMU/SEI-94-TR-013, ESC-TR-94-013.
- [4] April A., Hayes J. H., Abran A., Dumke R.. Software Maintenance Maturity Model (SMmm): the software maintenance process model. Journal of software maintenance and evolution: research and practise, pp.197-233.
- [5] Coleman, D., Ash, D., Lowther, B. and Oman, P. Using Metrics to Evaluate Software System Maintainability. IEEE Computer, August, pp.44-49.
- [6] Abran A., Moore J.W. Guide to the software body of knowledge (SWEBOK). Ironman version. IEEE Computer Society Press: Los Alamitos CA,pp. 6-1-6-15.