Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 9, Issue, 04, pp.48717-48719, April, 2017

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

REVIEW ARTICLE

SOFTWARE MAINTENANCE

*Ketan Sangadiya and Pinal Patel

Department of Computer Science Engineering, Government Engineering College, Gandhinagar Sector-28, Gandhinagar, India

ARTICLE INFO

Article History:
Received 24th January, 2017
Received in revised form
06th February, 2017
Accepted 22nd March, 2017
Published online 20th April, 2017

Key words:
Software Engineering, Software Maintenance, Maintenance Techniques, Maintenance Tools.

ABSTRACT

In the software development life-cycle, software maintenance is considered as one of the most critical process. A considerable amount of information needs to be managed during the process software maintenance. The requirement of particular software might not be fulfilled by a particular tool or technique. So we require different tools and techniques for different components of the software. This paper briefly explains some of the software maintenance techniques and tools.

Citation: Ketan Sangadiya and Pinal Patel, 2017. "Software maintenance", *International Journal of Current Research*, 9, (04), 48717-48719.

INTRODUCTION

Software maintenance was first introduced in 1969 by Meir Lehman. After a long time of 20 years, research work of Lehman drew to the formation of Lehman's Law. Major revelations of Lehman's research work contains that software maintenance is actually progressive expansion. Also that decisions related to software maintenance are assisted by concluding what happens to software or systems as time passes. Meir Lehman explained that as time goes, systems will go ahead to expand. As they expand, some action has to be performed like code refactoring to decrease the complexity because they will become more and more complex (Lehman *et al.*, 1997). Software maintenance is globally accepted as a part of Software Development Life Cycle now a days. It stands for all the modifications and updations done after the delivery of software product to the client. Software is multidimensional and software systems contains various artifacts such as specifications, documentation, documents of design, source code and test cases, etc. Each of the software dimensions explains only a fixed part of the software system—the actual system is their combination (Reiss, 2006). In the software engineering software maintenance is the alteration of a software outcome after successful deployment to remove errors, to increase attributes like performance and other. There are various reasons, why modifications are necessary, some of them are given below:

i. To correct errors
ii. To implement enhancements
iii. To improve the design
iv. To interface with other software
v. To retire software system
vi. To migrate legacy software (Grubb and Takang, 2003).

In the end of 1970s, a well-known and widely quoted survey done by Swanson and Lientz, unveiled the very high part of development life-cycle costs that were being spent on software maintenance. Swanson and Lientz classified activities of software maintenance into four types as listed below:

i. Preventive – improving software reliability or maintainability to prevent problems in near time
ii. Adaptive – To deal with alterations in the software environment by updating the software system
iii. Perfective – Establishing a new or altered user needs which deal with functional improvements to the software
iv. Corrective – diagnosing and fixing faults, possibly those found by users (Siegmund, 2016).

The following figure-1 will give a brief idea about the software maintenance:

This paper explains in brief some techniques and tools which is used in software maintenance process.

***Corresponding author:* Ketan Sangadiya,**
Department of Computer Science Engineering, Government Engineering College, Gandhinagar Sector-28, Gandhinagar, India.

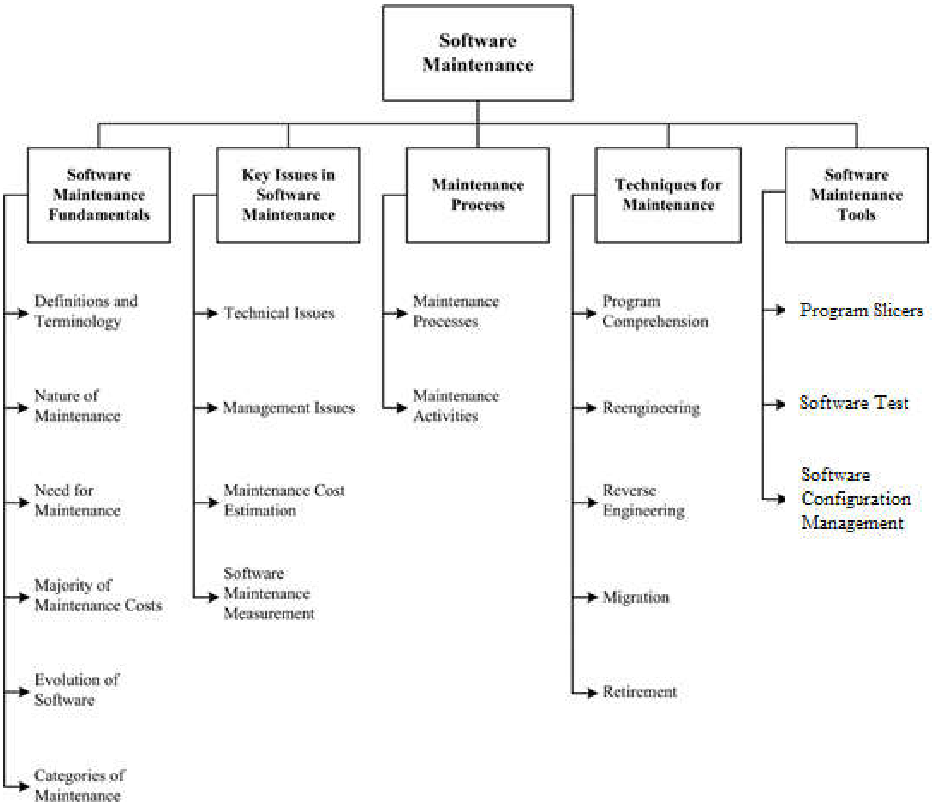
Software Maintenance
Software Maintenance Fundamentals
Definitions and Terminology
Nature of Maintenance
Need for Maintenance
Majority of Maintenance Costs
Evolution of Software
Categories of Maintenance
Key Issues in Software Maintenance
Technical Issues
Management Issues
Maintenance Cost Estimation
Software Maintenance Measurement
Maintenance Process
Maintenance Processes
Maintenance Activities
Techniques for Maintenance
Program Comprehension
Reengineering
Reverse Engineering
Migration
Retirement
Software Maintenance Tools
Program Slicers
Software Test
Software Configuration Management

Figure 1. Overview of Software Maintenance

Techniques

In this topic we will learn about some of the mostly accepted and used techniques in software maintenance.

Program Comprehension

Program Comprehension is an important cognitive process in software product development process, because Programmers waste lots of time to read and understand programs to implement changes. The key tool, which program comprehension uses is Coad browser to present and organize source code. Program comprehension can be also helped by clear and concise documentation. Now a days, software is used widely, and there are numbers of programming languages, IDEs, and source-code-layout guidelines that have been designed to assist the developer. Software projects can be delivered without critical errors and on time by this way (Siegmund, 2016).

Reengineering

Reengineering can be explained as the analysis and modification of software product to reconstruct it in a new form, and covers the subsequent execution of the new form. It is often not undertaken to increase maintainability but to replace aging legacy software. Reengineering has several techniques which include Refactoring that targets at reorganizing a program without altering its functionality. It tries to improve a program's maintainability and its structure. During minor changes we can use the Refactoring techniques (Grubb and Takang, 2003).

Migration

Today the development technologies are changes fast so organizations several times need to migrate their software products to a target technology from a source that could comprise a shift in programming model (IEEE Std., *14764-2006 (a.k.a.ISO/IEC 14764:2006)*). A software may have to be altered to run in various environments during its life time. While migrating software to another environment, the software maintainer requires to check the actions required to complete the migration. And there after develop and document the steps needed to accomplish the software migration in a migration plan that covers conversion of software product, software migration tools, software migration requirements and data, verification, execution, and support (Forite and Hug, 2014).

Tools

Program Slicers

Weiser first introduced program slicing which is a method for limiting the behaviour of a software program to some particular subset of interest (Gallagher and Lyle, 1991). Program Slicing minimizes the program to sub-statements, which reduces the same behaviour where the subset of the program also called the program slice comprises of the chunks or statements of the software which impact the costs calculated at a subject of absorption called as the slicing criterion (Chandra, 2015).

Software Test

Software maintenance takes place as the last activity of various software engineering processes. It is a vital activity and spends. Test case selection (*TCS*) is able to restrict the area of the licensed software that often alters its needs. Apart from that one of the critical issue is avoiding bugs of the ability of constructing the additional new codes (Lawanna, 2016).

Software Configuration Management

The process of monitoring and controlling changes in the computer software is known as software configuration management. When computer software are being built it is likely to happen changes. So we must have to manage it effectively. Software configuration management (SCM), which is often called as change management. It is a bunch of activities created to manage change by detecting the work products that are possible to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and reporting on the changes made and auditing (Pressman, 2009).

Conclusion

Software maintenance is the crucial part of the software engineering. In this paper we have discussed some of the techniques and tools which is used in software maintenance. Though this tools and techniques are used in software maintenance, there is always scope for the improvement to reduce the effort and cost to the organization.

REFERENCES

Chandra, A. 2015. "A study of program slicing techniques for software development approaches," No. September, pp. 4–5.

Forite L. and C. Hug, 2014. "FASMM: Fast and accessible software migration method," *Proc. - Int. Conf. Res. Challenges Inf. Sci.*

Gallagher K. B. B. and J. R. R. Lyle, 1991. "Using Program Slicing in Software Maintenance," *Ieee Trans. Softw. Eng.*, vol. 17, no. 8, pp. 751–761.

Grubb P. and A.A. Takang, 2003. *Software Maintenance: Concepts and Practice*, 2th ed., World Scientific Publishing.

IEEE Std., *14764-2006 (a.k.a.ISO/IEC 14764:2006) Standard for Software Engineering—Software Life Cycle Processes-Maintenance*, IEEE, 2006.

Lawanna, A. 2016. "Test Case Design Based Technique for the Improvement of Test Case Selection in Software Maintenance," pp. 345–350.

Lehman, M. M., J. Ramil, P. Wernwick, D. Perry, and W. Turski, 1997. "Metrics and laws of software evolution - The nineties view," *Fourth Int. Softw. Metrics Symp. Proc.*, pp. 20–32.

Pressman, R. S. 2009. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*.

Reiss, S. P. 2006. "Incremental maintenance of software artifacts," *IEEE Trans. Softw. Eng.*, vol. 32, no. 9, pp. 682–697.

Siegmund, J. 2016. "Program Comprehension: Past, Present, and Future," *2016 IEEE 23rd Int. Conf. Softw. Anal. Evol. Reengineering*, pp. 13–20.

Software Maintenance and Re-engineering, CSE2305 Object-Oriented Software Engineering

****