



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijareeie.com](http://www.ijareeie.com)

Vol. 6, Issue 3, March 2017

## A Review on Software Maintenance

Dr. T.Ganesh Kumar

Department of Computer Science and Engineering, Galgotias University, Yamuna Expressway Greater Noida, Uttar Pradesh, India

Email Id: [tganeshphd@yahoo.com](mailto:tganeshphd@yahoo.com)

**ABSTRACT:** Over the last several years, software maintenance has developed drastically to deal with the ever-changing application development frameworks and programming languages, and to implement progressively sophisticated computer models. Maintenance performs a vital function in the life-cycle of software engineering. The software project can only be implemented within an expected time frame if all steps of the software design cycle are accomplished within an expected time frame and are mainly planned. Software maintenance is a difficult and vitally important task in the development cycle of software. Software maintenance is essentially the refinement made to ensure the quality of brand after it is distributed to the organization or customer who requested it. In modern ever-increasing technical business manufacturing is done for the finest shipment of products, and also high performance. The consultancies typically provide the client or customer with a finished model as well as further development or improvements are produced as per the request made, such evolutions or adjustments are rendered under Software Maintenance.

**KEYWORDS:** Challenges, Software Development Life Cycle, Software Maintenance, Software Quality

### I.INTRODUCTION

Building a Software goes through several procedures. Almost every stage is quite important and the absence of a single-phase can result in further measures to faulty software or mistakes. The software goes through a process called the SDLC in which each phase is specified for its use while achieving the software [1]. A company generally oversees each level's elements to obtain comprehensive software building information. The privatized firms typically play a vital role in a specific area such as questionnaires, articles or ground monitoring. Such elements are then merged according to the business-specified or user-specified software design. The lifespan of Software Development has many stages [2].

An SDLC involves coordination, collection of requirements, pilot project, and assessment of the system, product design, programming, test, convergence, execution, activities, maintenance, and disposition. Communication is the early phase where the commodity is requested with an ambiguous configuration describing how well the software application should be and the definitions are typically bargained on individual factors. Requirement collection is the second phase in which a group from multiple departments is assembled to get specific information in the issue area. Next, the requirements are identified across different questionnaires sessions and divided into different categories of individual, device, and functional requirements. The specifications are obtained by conducting different tasks as per consumer as per the product type. The pilot project is a step while the group has a vague idea of how to create the software package, whether or not all the specifications can be met. The system design ensures that software designs work best into the vague layout. It includes program constraints and issues that may be confronted during growth [3]. The next phase is software design in which all the information acquired from either the process of evaluation and specifications is placed into use by developing the basic architecture of how the software will appear and react.

The coding process is essentially implementing this development of software on multiple platforms and structures using different languages as needed by the layout. Testing is very critical for understanding software faults and functioning. There are many tests for ensuring the software works perfectly and the software is normally examined by coders themselves. After programming the commodity is incorporated with various libraries, applications and other initiatives to incorporate it with institutions of the outside world according to the architecture setup. Then it is examined for non-functional needs. In the maintenance and support process, after delivery the product is updated or improved to



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijareeie.com](http://www.ijareeie.com)

**Vol. 6, Issue 3, March 2017**

improve its productivity and maintain the system updated. In the maintenance and support process, after delivery the product is updated or improved to improve its productivity and maintain the system updated. The last step in SDLC is Software Maintenance.

## II. SOFTWARE MAINTENANCE

The word "software maintenance" is being used to know the behaviour of software engineering which occurs throughout software advancement [4]. The software maintenance method is a very compacted method and typically involves more than a quarter of the development process again. Maintenance of the software is very important to finish and update the product to improve its performance and render it less susceptible to failures. Maintenance of software also tends to bring the material as close to consumer requirements as necessary. Software maintenance will modify the software after the brand has been supplied. Changes are normally required due to ever-changing market patterns, customer requirements, network changes in equipment or systems used, and institutional changes when necessary. Maintenance is separated into various types according to commodity existence. The kinds of maintenance are "corrective maintenance, adaptive maintenance, perfect maintenance and preventative maintenance".

### *Types of Software Maintenance*

*Corrective Maintenance:* Corrective maintenance includes changes and enhancements that are done to address issues or modify items as reported by malfunction notifications or users. Corrective maintenance concerns with correcting errors or flaws identified in the everyday operations of the device [5]. Mistakes in software design, reasoning, and coding may lead to a defect. Technical errors occur when modifications to the software are inaccurate, missing, interpreted incorrectly or the application for modification is not interpreted.

*Adaptive Maintenance:* Adaptive maintenance upgrades and modifies to maintain the software product updated to stay updated on current developments and situations. Adaptive maintenance is implementing improvements in a part of the system which was influenced by a shift in some parts of the system.

*Perfective Maintenance:* Proper maintenance is improvements and upgrades so the product can work over a prolonged period of time. It generally includes innovative features and specifications to improve the product's workability and reliability. It copes with the development of new or altered user needs.

*Preventive Maintenance:* Preventive maintenance involves upgrading and modification to avoid potential product faults or mistakes. It allows in non-functional and functional specifications that are not relevant in the current but may lead to potential errors or issues. Preventive maintenance requires doing operations to avoid errors from occurring. This requires upgrading of the software, upgrading code, and modifying code.

# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijareeie.com](http://www.ijareeie.com)

Vol. 6, Issue 3, March 2017

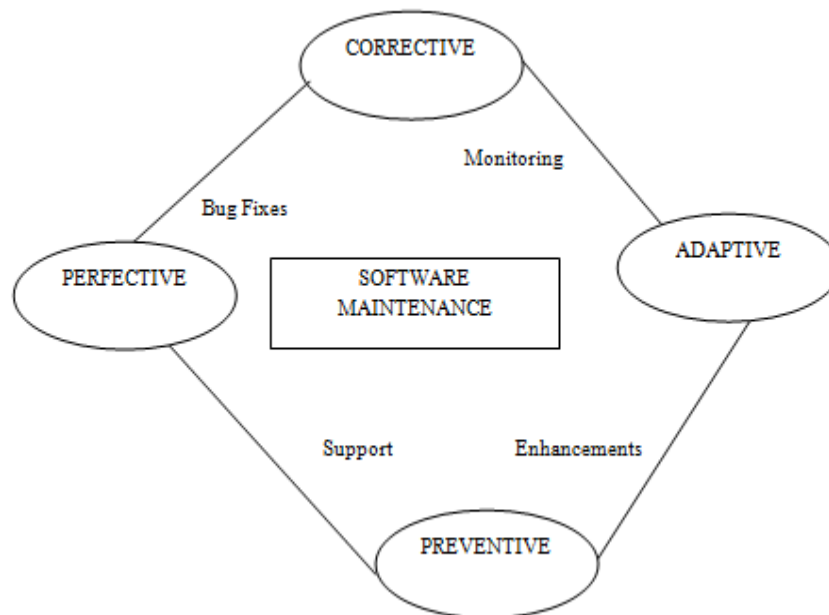


Fig. 1: Types of Software Maintenance

Software maintenance guarantees quality of operation so that the product stays on the business for as long as possible. With improvements in patterns and new technology creation, it is essential to adjust and upgrade the program as per current needs. Regular maintenance guarantees all these by taking into consideration each aspect required to keep product quality.

### III.MAINTENANCE PROCESS

Software maintenance is among the primary application development issues. A proper maintenance system is very important for maintaining software quality [6]. The various maintenance activities of software maintenance are:

*Identification and Tracing:* The essential step and also a very essential one are to identify the problem and trace its origin. The team looks very critically into each viewpoint so that the detection of the issue is achieved without any delays. The prerequisite for adjustments or upkeep is recognized by a series of steps as per customer requirements or the continuous commodity change. The primary maintenance problem is generated through survey records or texts by the user. Also recognized is the type of maintenance the team should be working on.

*Analysis:* The system's modifications or improvements should be evaluated with due care and managing. The research is conducted to ensure the latest changes do not affect the process, and it does not make the system prone to threats. The review guarantees the system's protection and security. If the change's effect is dramatic then more options are explored to get the best answer from the problem. When solution has been identified, the necessary changes to be made must eventuate into the obtained requirements. At this level, only the maintenance expenses are examined, and an assessment is given for the job.

*Design:* At this point, the designs needed for the new models are developed, and it is ensured that they fit into the old system. If the new design creates work issues or affects the intended work then the design will be scrapped, and the team will have to work with a new design. The group ensures that all the customer specifications and requirements are met by the new designs. The test cases are drawn to validate the new models to maintain high performance and reliability of the structure under the new model.

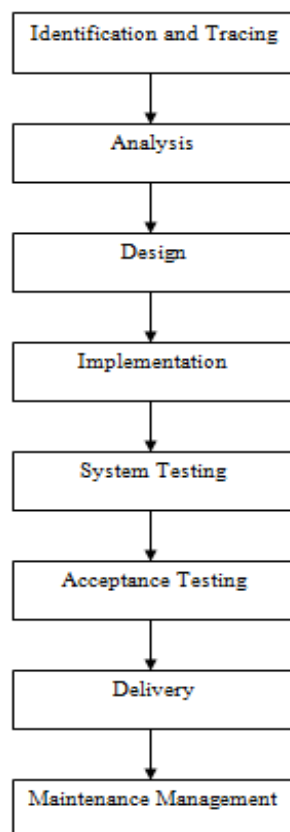


## International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijsareeie.com](http://www.ijsareeie.com)

Vol. 6, Issue 3, March 2017



**Fig. 2: Software Maintenance Process**

*Implementation:* The new models are submitted to the software developers who program to develop the new design ensuring it meets the code's structure. The testing for each element is performed in connection to assure that the compiled code functions with the new module. The coders make sure optimum code so that the system works in the new surroundings with greater precision.

*System Testing:* Integration testing of newly developed modules is conducted. The testing is also conducted at both device level and element level [7]. Between both the system and each element, compatibility checks are also taken out to insure that each new feature fits well with the framework without mistakes. The assessments are held out in the working environment to make sure that it operates at a minimum and high level in actual time with minimal shortcomings. The Software Maintenance Process is shown below in figure 2.

*Acceptance Testing:* The structure is examined for acceptance with customers after examining it in the working environment to make sure that all demands are met as stated by the customer. If at this level there are any differences then the assessment of the customer is observed and resolved by obeying incarnation. The experiment conducted is then conducted with the client in order to obtain client approval.

*Delivery:* After the customer has accepted a new adjustments, the new device is dispatched by various techniques to the field. The new notifications are usually provided through an update bundle or an install or upgrade of the entire application commodity. The user now performs the preliminary testing to ensure all defined non-functional and functional needs are met. The whole system is recorded by incorporating new adjustments / adjustments that are made on the system. The customer is provided the document and the paperwork.

*Maintenance Management:* The assigned group now answers the questions of various users facing problems with the program either in deployment or in operation. For the user-facing problems at any step, the requests are tackled and a suitable log is established so that any mistakes that occur are corrected or updated to reduce errors at the next alteration.



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijareeie.com](http://www.ijareeie.com)

Vol. 6, Issue 3, March 2017

## IV.MAINTENANCE REDUCTION

There are different ways in which software maintenance is. When earlier measures are checked closely and if they are performed with high accuracy then maintenance jobs can be reduced significantly. There are many ways of maintenance reduction they are-

*Re- Documentation:* It is also done to create alternative opinions of the system in dataflow, design or control flows to make sure that the users are able to understand the product easily.It's also designed to create alternate device views in dataflow, layout, or control processes to ensure users can easily interpret the product.The progressive redocumentation increases software comprehensibility and gradually restores the report after the developers ' jobs have been done.

*Decreasing Sales:* One way to minimize maintenance costs is by cutting domestic and foreign turnovers.External turnover can also be due to several factors within the organization and can lead in tremendous revenue while external turnover is due to external causes and typically affects the individual by providing new incentives or facilities.To minimize this, stringent initiatives and work schedules are rendered in such a way that turnover is negligible in a project.

*Removal of Dead Code:* Removal of dead code makes the system more designed to operate and reduces the failure rate. Dead code is essentially a piece of software which has no part to play in any part which is not used by the program.This code is usually there when it was used in the major steps of the software designs, and the developers did not replace it from the framework after changes and upgrades.Decreasing code volume affects operating performance and reliability as the process uses limited resources and therefore operates at a higher velocity.

*Understandability:* Until implementation, it is necessary for the tech team to acquire appropriate system knowledge and awareness so as to reduce the risk of mistakes.They have to have gone through all the commodity creating stages, and must be upgraded on the actual market requirements as well as how to enforce alterations.They have to guarantee that they comprehend the layout of the structure very deeply and that the element interrelationships are achieved with par excellence for various applications.

*Software Re-Engineering:* The upgrades and changes that are performed on a device without impacting its features are called re-engineering software. It is a big effort that all organizations are carrying out to remove the user found bugs and mistakes. The program is re-engineered so all of the systems work properly.The re-engineering process includes determining which portion to reengineer. The Model of Software Re- Engineering is shown below in Fig. 3 Model of Software Re- Engineering

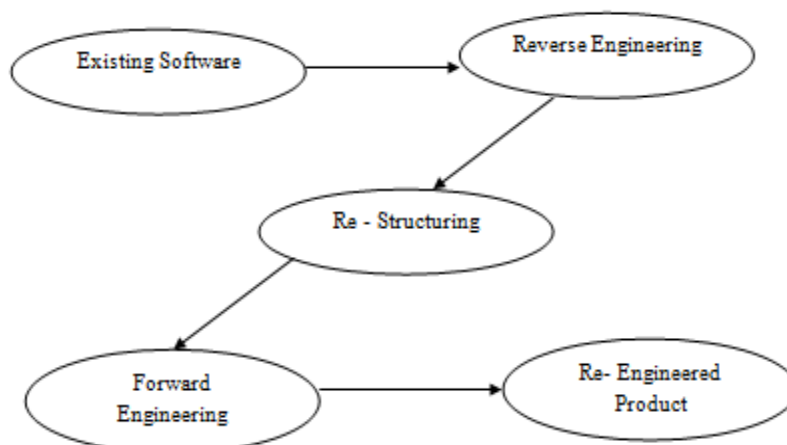


Fig. 3: Model of Software Re- Engineering



# International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijareeie.com](http://www.ijareeie.com)

Vol. 6, Issue 3, March 2017

## V. CHALLENGES OF SOFTWARE MAINTENANCE

Much of the software maintenance related problems that lead to software development method defects [8]. Many technological and administrative issues are met while retaining software. The Various challenges of software maintenance are-

*Expense:* Different research studies have suggested that software maintenance uses up 70 to 80 percent of the budget over the entire life cycle of production; such studies also document that cost of maintenance are primarily due to improvements instead of clarifications [9].

*Impact Analysis:* Among the most popular software maintenance problems is finding out about the impacts of a suggested alteration on the entire system. Impact analysis is the intervention of evaluating the likely effects of a shift with the plan to cut sudden adverse effects. The role includes determining the feasibility of a planned change and determining the risks associated with its execution, including assessments of land, resources and construction impact.

*Corrective Modifications:* One of the core issues is corrections since it is difficult to find a way to make adjustments. The code base can be hard to identify. If the conceptual design is decreased a minute addition might rely on improvements in the design which takes a long time. If a full solution of one issue has arisen then another one is even harder to break.

*Adaptive Changes:* Adaptive improvements are often not simply due to a lack of information regarding what the program is being changed to. The future technology's complex evidence is hard to take hold of. It's also difficult to observe the effect and to find architectures to the new things. Issues arising from an imbalanced feasibility study are of interest.

*Program Comprehension:* Another major issue is the understanding of the system, which ensures that maintenance technicians will spend a considerable amount of time reading and understanding the code, the relevant documents having a much better viewpoint on its meaning, intent, and function to retain a part of the software and improve software quality.

*Regression Testing:* After upgrading and changing the software application, regression testing is implemented to see how things work in the simulated environment. This test guarantees that the changes made are not defective, and modifications have not changed the system design. The checking can be done multiple times to ensure consistent reliability and performance products [10].

## VI. CONCLUSION

Software maintenance is known to be a major phase after the software has been made, and it can become boring to perform jobs. Software maintenance is a growing market where the resources and repair and manufacturing costs are typically more than expanding. The processes conducted are normally taken up by a group and split appropriately as per teammate's expertise and the necessity for the requirement. Processes performed to change and update a program must be done with sufficient accuracy so that all existing errors are resolved and the existing usability is not affected. The maintenance part of the product is very important to maintain with the demand so the device doesn't get outdated. The paper presented a summary of the software development maintenance process cycle by discussing all of the procedure for the same. There are many challenges/ issues during software maintenance resulting in improperly performed part of system evaluation.

## REFERENCES

- [1]Tutorialspoint, "SDLC - Overview," Tutorialspoint. 2017.
- [2]R. Scroggins, "SDLC and Development Methodologies," Glob. J. Comput. Sci. Technol. C Softw. Data Eng., 2014.
- [3]R. Arora and N. Arora, "Analysis of SDLC Models," Int. J. Curr. Eng. Technol., 2016.
- [4]"software maintenance," in SpringerReference, 2011.
- [5]E. Varga, Unraveling Software Maintenance and Evolution. 2017.
- [6]F. Javed and F. Rasul, "Software Maintenance Outsourcing ( Systematic Literature Review )," in Proceedings of 2nd International Multi-Disciplinary Conference 19-20 December 2016 Gujrat, Pakistan Software, 2016, no. December.
- [7]P. Ammann and J. Offutt, Introduction to Software Testing. 2016.





ISSN (Print) : 2320 – 3765  
ISSN (Online): 2278 – 8875

## International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijareeie.com](http://www.ijareeie.com)

**Vol. 6, Issue 3, March 2017**

- [8]A. Gupta and S. Sharma, “Software Maintenance : Challenges and Issues,” Int. J. Comput. Sci. Eng., vol. 4, no. 01, pp. 23–25, 2015.
- [9]N. Alija, “Justification of Software Maintenance Costs,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 7, no. 3, pp. 15–23, 2017.
- [10]Unknown, “Testing Mediation with Regression Analysis,” USP 654 Data Anal. II, 2015
- Gagandeep Singh Narula, Dr. Vishal Jain, Dr. S. V. A. V. Prasad, “Use of Ontology to Secure the Cloud: A Case Study”, International Journal of Innovative Research and Advanced Studies (IJIRAS), Vol. 3 No. 8, July 2016, page no. 148 to 151 having ISSN No. 2394-4404.
  - RS Venkatesh, PK Reejeesh, S Balamurugan, S Charanyaa, “Further More Investigations on Evolution of Approaches for Cloud Security”, International Journal of Innovative Research in Computer and Communication Engineering , Vol. 3, Issue 1, January 2015
  - K Deepika, N Naveen Prasad, S Balamurugan, S Charanyaa, “Survey on Security on Cloud Computing by Trusted Computer Strategy”, International Journal of Innovative Research in Computer and Communication Engineering, 2015
  - P Durga, S Jeevitha, A Poomalai, M Sowmiya, S Balamurugan, “Aspect Oriented Strategy to model the Examination Management Systems”, International Journal of Innovative Research in Science, Engineering and Technology , Vol. 4, Issue 2, February 2015
  - RS Venkatesh, PK Reejeesh, S Balamurugan, S Charanyaa, “Further More Investigations on Evolution of Approaches and Methodologies for Securing Computational Grids”, International Journal of Innovative Research in Science, Engineering and Technology , Vol. 4, Issue 1, January 2015