Software Maintenance

Ms. Rinku Kalotra, Harpreet Kaur

Abstract

Software maintenance represents an essential stage of the software development life cycle (SDLC), ensuring that the software system remain dependable, secure and capable of adapting to changing user requirements. With rapid advancements in technology, maintaining software has become more challenging and now involves tasks such as fixing defects, enhancing performance, implementing adaptive updates and applying preventive improvements. The study categorises maintenance into corrective, adaptive, perfective and preventive activities, highlighting how each contributes to addressing defects, meeting evolving requirements, optimizing functionality and reducing future risks.

INTRODUCTION

The term software maintenance is used to described the software engineering activities that occur following delivery of a software product to the customer. The maintenance phase of software life cycle is the time period in which a software product performs useful work. Typically, the development cycle for software span 1 or 2 years, while the maintenance phase spans 5 to 10 years.

Maintenance activities involve making enhancements to software products, adapting products to new environments, and correcting problems. Software product enhancement may involve providing new functional capacities, improving user displays and modes of interaction, upgrading external documents and internal documentation or upgrading the performance characteristics of a system. Adaptation of a software to a new environment may involve moving the software to a different machine, or for instance, modifying the software to accommodate a new telecommunication protocol or an additional disk drive. Problem correction involves modification and revalidation of software to correct errors. Some errors require immediate attention, some can be corrected on a scheduled, period basis and others are known but never corrected.

- Keywords:
    - Maintainability
    - Version control
    - Adaptation
    - Technical Expertise
    - Tracking and controlling

❖Development activities that enhance software maintainability

- Analysis Activities
  1. Develop standards and guidelines
  2. Specify quality assurance procedures
- Architectural Design Activities
  1. Emphasize clarity and modularity as design criteria
  2. Design to ease likely enhancements
- Detailed Design Activities
  1. Use standardized notations to specify algorithms, data, structure and procedure interface specifications

2. Specify side effects and exception handling for each routine

- Implementation Activities

  Use single entry, single exist construct

  Use simple, clear coding style

- Future and Scope of Software Maintenance:

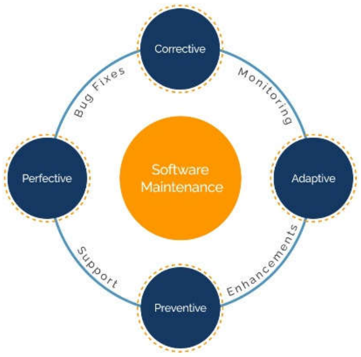
  1. Simplified- The future of software maintenance is promising, as systems will require continuous updates, security fixes and performance improvements.
  2. Academic Style: The scope of software maintenance is expanding rapidly. With emerging technologies and increasing system complexity, maintenance activities will remain essential for enhancing functionality, ensuring system stability, and meeting user expectations over time.
  3. Technical Rephrasing: Software maintenance will continue to play a major role in refining existing applications. As software lifespan extend , developer will focus more on upgrading features, resolving defects, optimizing performance and adapting products to new environments and platforms.

- ## Managerial aspects of Software Maintenance:

- Successful software maintenance, like all software engineering activities require a combination if managerial skills and technical expertise. On of the most important aspects of software maintenance involves tracking and control of maintenance activities. Maintenance activity for a software product usually occurs in response to change result filed by a user of the product.

- Change request processing can be described by the following algorithm

```
     Software change request initiated
   Request analysed
  If (request not valid) then
   Request closed
 Else
       -Request and recommendation submitted to change control board
       If (change control board concurs) then
       Modification performed with priority and constraint
Established by change control board
Regression tests performed
```



```
Changes submitted to change control board
If (change control board approves) then
Master tape updated
External documentation updated
Update distributed as directed by change control board
Else
Control board objections satisfied and changes resubmitted to control board
Else
Request closed.
```

❖ Features of Software maintenance

- Focus on existing systems: - Maintenance deals with improving or modifying software that is already deployed rather than creating new systems.
- Error removal: - It include identifying and repairing faults that appear during real world usage.
- Adapting to Environment changes: - Updates are made to ensure the software continues to function correctly when there are changes in hardware, operating systems or business rules.
- Enhancement of quality and Functionality: - Additional features, performance improvement and

usability upgrades are carried to satisfy evolving user requirements.

- Risk and failure prevention: - Internal improvements are made the likelihood of future errors and maintain long term reliability.

❖ Advantages of Software maintenance:

1. Enhances system performance-

   Ongoing updates help the software operate faster and more efficiently.

2. Improve Software reliability-

   Regular fixes and adjustments reduce the chances of system failures and errors.

3. Adapts to changing requirements-

   Maintenance ensures the software stays compatible with new technologies and evolving business needs.

4. Extends product lifespan-

   By keeping the system updated, the life of the software is prolonged without needing complete replacements.

5. Boosts User satisfaction-

   Improved features and reduced issues lead to a better user experience.

❖ Disadvantages of Software Maintenance-

1.High ongoing expenses-

Continues updates and support can be costly over the software's lifetime.

2.Required Skilled professional-

Maintenance offer needs experienced staff, which may be difficult to manage or hire.

3.Time consuming process-

Fixing issues and improving features can take a long time, possibly delaying operations.

4.Risk of new errors-

Modifying existing code might accidently introduce additional bugs or complications.

5.Complexity increases over time-

As more changes are applied, the system may become harder to manage and understand.

❖ Conclusion

Software maintenance is not just a post-development liable, secure, and aligned with evolving user needs. By continuously refining, updating, and adapting software, organizations ensure that it remains efficient, relevant, and capable of meeting future challenges. In essence, maintenance transforms software from a static product into a living, evolving solution that grows alongside its environment.

❖ Reference

- Horning, J.,
- D. Wortman
- Lehman, M.M
- Swanson, E.B
- https://tse1.mm.bing.net/th/id/OIP.PkRnWHdx7oFtwmJCoWOUjgHaJ3?rs=1&pid=ImgDetMain&o=7&rm=3
- https://www.openapp.ie/wp-content/uploads/2023/05/maintenance-processes.jpg