

Software Maintenance

- Overview of software maintenance activities
- Types of software maintenance
- Software change management process
- Testing and quality assurance procedures for software updates

Overview of software maintenance activities

Software maintenance is a widely accepted part of the Software Development Life Cycle. It refers to the modifications and updates done after the delivery of a software product. There are number of reasons, why modifications are required, some of which are listed below:

- New user Requirements - Over the time, user may ask for new features or functions in the software based on evolving needs in an organisation.
- Change to the Hosting Environment - If any of the hardware and/or platform (such as operating system) of the target host changes, software changes may be needed to keep adaptability.
- Updates, patches and upgrades - For externally sourced systems, the software vendor may make improvements from time to time to keep pace with technology trends or to enhance security or performance of the system.
- Legal and regulatory changes - Changes to laws and regulations may require changes to software.

Types of software maintenance

There are four types of software maintenance:

- Corrective Software Maintenance
- Adaptive Software Maintenance
- Perfective Software Maintenance
- Preventive Software Maintenance

Corrective Software Maintenance

Corrective software maintenance addresses the errors and faults within software applications that could impact various parts of your software, including the design, logic, and code. These corrections usually come from bug reports that were created by users but corrective software maintenance can also help to spot them before your users do.

Adaptive Software Maintenance

Adaptive software maintenance becomes important when the environment of your software changes. This can be brought on by changes to the operating system, hardware, software dependencies, cloud storage, or even changes within the operating system. Sometimes, adaptive software maintenance reflects organizational policies or rules as well.

Perfective Software Maintenance

Perfective software maintenance focuses on the evolution of requirements and features that exist in your system. As users interact with your applications, they may notice things that you did not or suggest new features that they would like as part of the software, which could become future projects or enhancements. Perfective software maintenance takes over some of the work, both adding features that can enhance user experience and removing features that are not effective and functional. This can include features that are not used or those that do not help users to meet their end goals.

Preventive Software Maintenance

Preventive software maintenance helps to make changes and adaptations to software so that it can work for a longer period of time. The focus of the type of maintenance is to prevent the deterioration of your software as it continues to adapt and change. These services can include optimizing code and updating documentation as needed. Preventive software maintenance helps to reduce the risk associated with operating software for a long time, helping it to become more stable, understandable, and maintainable.

For all businesses and organizations, software maintenance is an essential part of the software development lifecycle. It is absolutely necessary for the success of any software and any evolution into the future. It is important to know that maintenance needs to go much further than fixing

issues or bugs - that is only one step of the software maintenance process.

Software change management process

Software change management is the process of managing any modifications to software while minimizing the risk that may be posed by new changes. To mitigate the risk, a structured process should be followed in making any changes to software. Government institutions are should follow the following steps when introducing new software changes:

i. Raise a formal Change Request

A change request should be formally documented requesting for any changes that need to be done to software. This could be an enhancement or modification requested by users to improve the software or to respond to changing operational requirements. The change request describes the changes required and reasons for the change. A change request document or form is typically completed to clearly define the user requirement.

ii. Impact Analysis

An impact analysis is done by the technical team to understand the impact and feasibility of the change requested. This will involve:

- Identifying all the systems and the system products affected by the requested change.
- Assessing any technical impacts of the change e.g. on security or infrastructure.
- Making an estimate of the resources or costs needed to effect the change.
- Analyzing the benefits of the change against the costs.

The impact of the change is updated on the change request document with a recommendation on whether the change should be approved.

iii. Change request approval

The change request should be formally approved based on the impact analysis. The approval should be made by an authorised person in the requesting user department as well the IT department. Any resources required to implement the change should be allocated.

iv. System Release Planning

In this phase, the schedule and the contents of the software release are planned. This includes defining the release process which covers the process for testing and deployment to the production environment.

v. Change Implementation

The implementation of changes involves designing the changes based on the user requirements, writing software code to implement the change and finally testing the changes. The change could also require changes to software configurations or setups which may not require coding.

vi. Software Change Release

During software release, the following points should be carried out:

- Updating software documentation - relevant documentation should be updated including user manuals, technical specifications, and design documents.
- Training of users - users should be provided with communication and any relevant training that needs to be carried out to the user community regarding the change.
- Making any relevant hardware changes - major software changes may require hardware changes such as enhancement of processing or storage resources.
- Migration of changes - Migration of the changes from test environment to production environment. This is usually during non work hours.
- Configuration and data conversion - any software configuration or data changes that need to be done as a result of the change.
- Communication to users - informing users about the software change after it has been released

Testing and quality assurance procedures for software updates

Testing and quality assurance procedures refer to the processes that should be followed to ensure that software changes are adequately tested to avoid introducing negative impacts such as new bugs or security risks when they are deployed in the Production environment. Before software changes are introduced in the Production environment they should be thoroughly tested. For more guidance on testing refer to the RISA testing and quality assurance guidelines.