

CH-2: Maintenance Support Processes

- ◆ Maintenance Planning
- ◆ Evolution and Maintenance Testing
- ◆ Configuration Management
- ◆ Problem Management
- ◆ Maintenance supporting tools

- ✓ A process that is used, when required, by an operational process is said to be an operational support process.
- ✓ In most IS/IT organizations, developers, maintainers, and operations share the responsibility for these processes.

Maintenance Support Processes...

- ✓ Support processes typically include:
 - ✓ A documentation process
 - ✓ A software configuration and version management process and tools
 - ✓ A product quality assurance process
 - ✓ The verification and validation processes
 - ✓ The problem resolution process, which is often shared with operations
 - ✓ The reviews and audits processes

Maintenance Planning

- ✓ Software maintenance, being a large portion of IT costs, needs to be part of the strategic IT plan.
- ✓ Maintenance planning ensures the creation and updating of plans that describe the current and planned software maintenance activities
- ✓ Software maintenance planning is typically elaborated from three perspectives: organizational, tactical, and operational.

Maintenance Planning...

- ✓ At the organizational level, planning goals are strategic, general, and corporate, and include partners, the SLA, contracts, and licensing.
- ✓ At the tactical level, the planning goal is to identify and plan the activities of pre delivery and transition for new software and also the yearly plans associated with a specific customer account.
- ✓ Finally, at the operational level, the goals are specific:
 - (a) To conduct impact analysis on a given maintenance request,
 - (b) To plan the recovery/failure tests,
 - (c) To plan software versions/releases (e.g., assign each request to a future version/release of the software), and
 - (d) To plan software upgrades (especially the COTS) for the many platforms.

Maintenance Planning...

Goals of Maintenance Planning KPA

1. Ensure a proactive collection and documentation of customer/user, development, and computer operation requests (in the short term, medium term ...) to conduct detailed planning.
2. Identify, communicate, and obtain consensus on priorities of requests currently in queue and assigned.
3. Identify the required controls for each type of maintenance service and application software under maintenance.
4. Plan the recovery/failure tests for all software in maintenance.
5. Identify version management plans (also called release plans) for all software maintained.

Maintenance Planning...

6. Prepare software upgrade plans.
7. Establish a rationale for the allocation of new requests to maintenance engineers and for the capacity of maintenance.
8. Inform the stakeholders of maintenance work performed, waiting in queue, and in progress based on agreed priorities.
9. Inform stakeholders on the status of budgets and on the use of resources.

Maintenance Planning...

Expected Results from the KPA

After the successful implementation of the exemplary practices of this KPA, the following can be observed:

- ✓ A sound forecast is available on needed resources in software maintenance (strategic plan, tactical plan, and operational plan).
- ✓ SLA, agreement, license, and contract renewals are planned.
- ✓ The failure/recovery tests are planned for each software in maintenance.
- ✓ Requests are allocated to future versions of the software.
- ✓ Software upgrades are planned.
- ✓ Pre-delivery and transition services are planned.
- ✓ There is an optimal allocation of maintenance resources to work items.
- ✓ Software maintenance has a plan, and this plan is communicated and approved.
- ✓ The stakeholders are informed about the plans.

EVOLUTION AND MAINTENANCE TESTING

Maintenance Testing

- ✓ Maintenance Testing is done on the already deployed software.
- ✓ The deployed software needs to be enhanced, changed or migrated to other hardware.
- ✓ The Testing done during this enhancement, change and migration cycle is known as maintenance testing.

Maintenance Testing...

Usually maintenance testing is consisting of two parts:

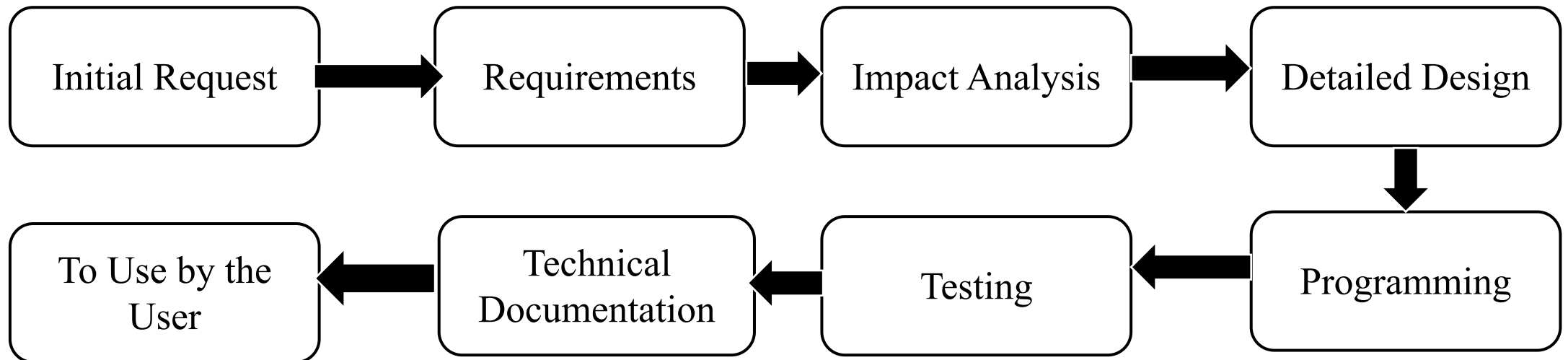
- ✓ **First** one is, testing the changes that has been made because of the correction in the system or if the system is extended or because of some additional features added to it.
- ✓ **Second** one is regression tests to prove that the rest of the system has not been affected by the maintenance work.

Configuration Management

- ✓ **Software Configuration Management** encompasses the disciplines and techniques of initiating, evaluating and controlling change to software products during and after the software engineering process.
- ✓ Configuration management consists of determining the configuration and description of products and intermediate work products at a precise time.
- ✓ Its objective is to ensure systematic control over configuration changes and to maintain integrity and traceability throughout the life cycle of a request that will modify the existing software.

Configuration Management...

- ✓ Configuration management's first objective is to establish a link between the different intermediate products throughout a change.
- ✓ The maintainer is thus able to follow a change's progress from



Configuration Management...

- ✓ This first aspect of configuration is called traceability.
- ✓ Another aspect of configuration management is the use of support software to reserve components while the software is being worked on.
- ✓ This activity ensures that employees cannot change a component without being aware of changes made by a colleague.
- ✓ Developers and maintainers also make images of software at every critical step using the same tools.
- ✓ For example
 - ✓ They would make a copy of the complete configuration at each major step in the testing phase.
 - ✓ This technique allows them to isolate an image of all software components at any given time.
 - ✓ They can always revert to this state should the work being done become corrupted.
 - ✓ It can be imagined that backup copies are, in fact, images of the software's configuration.

Configuration Management...

The fundamental differences between a backup copy and a configuration image are the following:

- ✓ A configuration image contains a great deal of information on individual components.
- ✓ A configuration image is kept on-line.
- ✓ All employees have access to configuration images and can extract reports from them.
- ✓ The configuration image is used during a change.
- ✓ A backup copy focuses only on the source code.

Problem Management

- ✓ A mechanized problem reporting process is used that ensures efficient communications for quick resolution of failures and other support requests.
- ✓ A specific user request, sometimes called a "ticket," will typically circulate between the help desk, software maintenance, and operations in order to isolate a problem.
- ✓ The IT department provides a large number of IT services to its customers, which are mainly departments from the sibling organization.
- ✓ To manage the communication with customers regarding those services, the department has implemented helpdesk management and problem management processes.

Problem Management...

- ✓ The implementation of these processes has been based on the Information Technology Infrastructure Library (ITIL).
- ✓ The ITIL process Helpdesk Management is used to guarantee the continuity of services, while the ITIL process Problem Management is used to improve the level of service in the future.
- ✓ **Helpdesk Management** deals with *incidents*, whereas **Problem Management** is concerned with solving the *problems* that cause these incidents.

Problem Management Process

The problem management process has many steps, and each is vitally important to the success of the process and the quality of service delivered.

1. **Detect the problem:** A problem is raised either through
 - ✓ Appreciation from the service desk
 - ✓ Proactive evaluation of incident patterns and alerts from event management
 - ✓ Continual service improvement processes.
2. **Log the problem:** Problems are logged in a problem record. A problem record is a compilation of every problem in an organization. Pertinent problem data, such as the time and date of occurrence, the related incident, the symptoms, previous troubleshooting steps...
3. **Categorize the problem:** Incident [and problem] categorization involves assigning a main and secondary category to the issue. This step is beneficial in several ways.
 - ✓ It allows the service desk to sort and model incidents that occur regularly.
 - ✓ The modeling allows for automatic assignment of prioritization.
 - ✓ The third and most important benefit is the ability to gather and report on service desk data.

Problem Management Process...

4. **Prioritize the problem:** is determined by its impact on users and on the business and its urgency. Urgency is how quickly the organization requires a resolution to the problem.
5. **Investigating and diagnosing the problem:** involves analyzing the incidents that lead to the problem report as well as further testing that may not be possible at the service desk level, such as advanced log analysis.
6. **Identify a workaround for the problem:** A workaround enables the service desk to restore services to users while the problem is being resolved. A problem can take anywhere from an hour to months to resolve, therefore a workaround is vital.

Problem Management Process...

7. **Raise a known error record:** It's good practice to record a known error in both an incident knowledge base and what ITIL calls a known error database (KEDB). Documenting the workaround allows the service desk to resolve incidents quickly and avoid further problems being raised on the same issue.
8. **Resolve the problem:** Resolution resolves the underlying cause of a set of incidents and prevents those incidents from recurring. Some resolutions may require the change management board, as they may affect service levels.
9. **Close the problem:** This step should only occur after the problem has been raised, categorized, prioritized, identified, diagnosed, and resolved.
10. **Review the problem:** During the review, the problem management team evaluates the problem documentation and identifies what happened and why.
 - ✓ Lessons learned, such as process bottlenecks, what went wrong, and what helped should be discussed.

Maintenance Supporting Tools

- ✓ Software maintenance tools are programs used by software engineers
 - ✓ To increase their productivity for gathering data
 - ✓ To detecting bugs and
 - ✓ To managing their software.
- ✓ Without these tools, it would be nearly impossible for these engineers to go through thousands of lines of code to find errors or determine why a particular server went down.
- ✓ The software Maintenance supporting tools set provides software engineers can categorized
 - ✓ Supporting maintenance quality enhancement
 - ✓ E.g. Metrics Measurement tool, Programming rule Checker, Redundancy Checker
 - ✓ Support for program repair work
 - ✓ E.g. Structure Analyzer, Maintenance document generator, Impact Analyzer

Maintenance Supporting Tools...

Examples of Software maintenance tools regarding program comprehension include:

- ✓ **Program slicers:** which select only parts of a program affected by a change
- ✓ **Static analyzers:** which allow general viewing and summaries of a program content
- ✓ **Dynamic analyzers:** which allow the maintainer to trace the execution path of a program
- ✓ **Data flow analyzers:** which allow the maintainer to track all possible data flows of a program
- ✓ **Cross-referencers:** which generate indices of program components; and
- ✓ **Dependency analyzers:** which help maintainers analyze and understand the interrelationships between components of a program.

Reference

- Alain April, Alain Abran (2008), Software Maintenance Management Evaluation and Continuous Improvement
- FRANK NIESSINK and HANS VAN VLIET(2000), Software Maintenance from a Service Perspective, Faculty of Sciences, division of Mathematics and Computer Science, Vrije Universiteit, De Boelelaan Amsterdam, the Netherlands.
- Pierre Bourque, École de technologie supérieure(2014), Guide to the Software Engineering Body of Knowledge (SWEBOK) Version 3.0, A Project of the IEEE Computer Society.
- www.techwalla.com/articles/list-of-software-maintenance-tools (Retrieved March 20 2018)
- www.softwaretestingmentor.com/maintenance-testing/(Retrieved March 20 2018)
- <http://www.bmc.com/guides/itil-problem-management.html>(Updated)