

IEEE Standard for Software Maintenance

Based on IEEE Std 1219-1993

By: Bassam Sayed

Scope

- This standard describes an iterative process for managing and executing software maintenance activities.
- The criteria established apply to both:
 - Software under development.
 - Existing software products.
- This standard prescribes requirements for:
 - process, control, and management of the
 - planning, execution, and documentation of software maintenance activities
- This standard does not assume the use of any particular development model

Definitions and Acronyms

- **Adaptive maintenance:** Modification of a software product performed after delivery to keep a computer program usable in a changed or changing environment.
- **Corrective maintenance:** Reactive modification of a software product performed after delivery to correct discovered faults.
- **Customer:** The person, or persons, for whom the product is intended, and usually (but not necessarily) who decides the requirements.

Definitions and Acronyms (Cont.)

- **Emergency Maintenance:** Unscheduled corrective maintenance performed to keep a system operational.
- **Interoperability testing:** Testing conducted to ensure that a modified system retains the capability of exchanging information with systems of different types, and of using that information.
- **Modification request (MR):** A generic term that includes the forms associated with the various trouble/problem-reporting documents (e.g., incident report, trouble report) and the configuration change control documents

Definitions and Acronyms (Cont.)

- **Perfective maintenance:** Modification of a software product after delivery to improve performance or maintainability.
- **Project:** A subsystem that is subject to maintenance activity.
- **Regression test:** Retesting to detect faults introduced by modification.
- **Repository:** (A) A collection of all software-related artifacts belonging to a system. (B) The location/format in which such a collection is stored.

Definitions and Acronyms (Cont.)

- SCM: Software configuration management
- CSA: Configuration status accounting
- FCA: Functional configuration audit
- PCA: Physical configuration audit
- PDL: Program design language
- SQA: Software quality assurance
- SCR: System/software change request
- V&V: Verification and validation
- VDD: Version description document

Software maintenance

- This standard defines changes to software process through the following phases:
 1. Problem/modification identification and classification
 2. Analysis
 3. Design
 4. Implementation
 5. Regression/system testing
 6. Acceptance testing
 7. Delivery
- We will look at each phase from the **input, process, control, and output.**

1- Problem/modification identification, classification, and prioritization

- In this phase, software modifications requests (MRs) are identified, classified, and assigned an initial priority ranking.
- Classification can be from the following maintenance types: corrective, adaptive, perfective, and emergency

1- Problem/modification identification, classification, and prioritization (Cont.)

- **Input:**
 - Input for this phase shall be an MR.
- **Process:** the following determinative activities must occur
 1. Assign an identification number
 2. Classify the type of maintenance
 3. Analyze the modification to determine whether to accept, reject, or further evaluate
 4. Make a preliminary estimate of the modification size/magnitude
 5. Prioritize the modification
 6. Assign MR to a block of modifications scheduled for implementation

1- Problem/modification identification, classification, and prioritization (Cont.)

- **Control:** MR and process determinations shall be uniquely identified and entered into a repository.
- **Output:** The output of this process shall be stored in a repository that contain the following items
 1. Statement of the problem or new requirement
 2. Problem or requirement evaluation
 3. Classification of the type of maintenance required
 4. Initial priority
 5. Verification data (for corrective modifications)
 6. Initial estimate of resources required to modify the existing system

2 - Analysis

- The analysis phase shall use the repository information and the MR validated in phase 1, along with system and project documentation, to study the feasibility and scope of the modification
- To devise a preliminary plan for design, implementation, test, and delivery.
- Classification can be from the following maintenance types: corrective, adaptive, perfective, and emergency

2 – Analysis (Cont.)

- **Input:**
 1. Validated MR
 2. Initial resource estimate and other repository information
 3. Project and system documentation, if available
- **Process:** Analysis is an iterative process having at least two components
 1. A feasibility analysis
 2. A detailed analysis
 - If the documentation is not available or is insufficient and the source code is the only reliable representation of the software system, reverse engineering is recommended

2 – Analysis (Cont.)

- **Feasibility Analysis:** shall be performed for MR and a feasibility report (FR) shall be prepared. This FR should contain the following
 1. Impact of the modification
 2. Alternate solutions, including prototyping
 3. Analysis of conversion requirements
 4. Safety and security implications
 5. Human factors
 6. Short-term and long-term costs
 7. Value of the benefit of making the modification

2 – Analysis (Cont.)

- **Detailed Analysis:** shall provide the following
 1. Define firm requirements for the modification
 2. Identify the elements of modification
 3. Identify safety and security issues
 4. Devise a test strategy
 5. Develop an implementation plan

2 – Analysis (Cont.)

- **Control:** Control of analysis shall include the following
 1. Retrieval of the relevant version of project and system documentation from the configuration control function of the organization
 2. Review of the proposed changes and engineering analysis to assess technical and economic feasibility, and assess correctness
 3. Identification of safety and security issues
 4. Consideration of the integration of the proposed change within the existing software
 5. Verification that all appropriate analysis and project documentation is updated and properly controlled
 6. Verification that the test function of the organization is providing a strategy for testing the change(s), and that the change schedule can support the proposed test strategy
 7. Review of the resource estimates and schedules and verification of their accuracy
 8. Technical review to select the problem reports and proposed enhancements to be implemented in the new release. The list of changes shall be documented.

2 – Analysis (Cont.)

- **Output:** The output shall include the following
 1. Ret Feasibility report for MRs
 2. Detailed analysis report
 3. Updated requirements (including traceability list)
 4. Preliminary modification list
 5. Test strategy
 6. Implementation plan

3- Design

- In the design phase, all current system and project documentation, existing software and databases, and the output of the analysis phase (including detailed analysis, statements of requirements, identification of elements affected, test strategy, and implementation plan) shall be used to design the modification to the system

3- Design (Cont.)

- **Input:** shall include the following
 1. Analysis phase output, including
 1. Detailed analysis
 2. Updated statement of requirements
 3. Preliminary modification list
 4. Test strategy
 5. Implementation plan
 2. System and project documentation
 3. Existing source code, comments, and databases

3- Design (Cont.)

- **Process:** shall include the following
 1. Identifying affected software modules
 2. Modifying software module documentation (e.g., data and control flow diagrams, schematics, etc.)
 3. Creating test cases for the new design, including safety and security issues
 4. Identifying/creating regression tests
 5. Identifying documentation (system/user) update requirements
 6. Updating modification list

3- Design (Cont.)

- **Control:** shall include the following
 1. Conduct software inspection of the design in compliance with IEEE Std 1028-1988.
 2. Verify that the new design/requirement is documented as a software change authorization (SCA), as per IEEE Std 1042-1987.
 3. Verify the inclusion of new design material, including safety and security issues.
 4. Verify that the appropriate test documentation has been updated.
 5. Complete the traceability of the requirements to the design.

3- Design (Cont.)

- **Output:** shall include the following
 1. Revised modification list
 2. Updated design baseline
 3. Updated test plans
 4. Revised detailed analysis
 5. Verified requirements
 6. Revised implementation plan
 7. A list of documented constraints and risks

4- Implementation

- In the implementation phase, the results of the design phase, the current source code, and system documentation (i.e., the entire system as updated by the analysis and design phases) shall be used to drive the implementation effort.
- **Input:** shall include the following
 1. Results of the design phase
 2. Current source code, comments, and databases
 3. Project and system documentation

4- Implementation (Cont.)

- **Process:** The implementation phase shall include the following four sub-processes, which may be repeated in an incremental, iterative approach
 1. Coding and unit testing
 - Implement the change into the code and perform unit testing and other appropriate SQA and V&V processes
 2. Integration
 - the modified software shall be integrated with the system and integration and regression tests shall be refined and performed.
 3. Risk analysis
 - risk analysis and review shall be performed periodically during the phase rather than at its end, as in the design and analysis phases
 4. Test readiness review (follow IEEE Std 1028-1988 Std)

4- Implementation (Cont.)

- **Control:**

1. Conduct software inspections of the code in compliance with IEEE Std 1028-1988.
2. Ensure that unit and integration testing are performed and documented in a software development folder.
3. Ensure that test documentation (e.g., test plan, test cases, and test procedures) are either updated or created.
4. Identify, document, and resolve any risks exposed during software and test-readiness reviews.
5. Verify that the new software is placed under software configuration management control.
6. Verify that the training and technical documentation have been updated.
7. Verify the traceability of the design to the code.

4- Implementation (Cont.)

- **Output:**

1. Updated software
2. Updated design documentation
3. Updated test documentation
4. Updated user documentation
5. Updated training material
6. A statement of risk and impact to users
7. Test readiness review report (see IEEE Std1028-1988)

5- System Test

- System testing, as defined in IEEE Std 610.12-1990, shall be performed on the modified system. Regression testing is a part of system testing and shall be performed to validate that the modified code does not introduce faults that did not exist prior to the maintenance activity.

5- System Test (Cont.)

- **Input:** shall include the following
 1. Test-readiness review report
 2. Documentation, which includes:
 1. System test plans (IEEE 829-1983)
 2. System test cases (IEEE 829-1983)
 3. System test procedures (IEEE 829-1983)
 4. User manuals
 5. Design
 3. Updated system

5- System Test (Cont.)

- **Process:** System tests shall be conducted on a fully integrated system. Testing shall include the performance of
 1. System functional test
 2. Interface testing
 3. Regression testing
 4. Test readiness review to assess preparedness for acceptance testing

5- System Test (Cont.)

- Control:
 1. System tests shall be conducted by an independent test function
 2. the test function shall be responsible for reporting the status of the criteria that had been established in the test plan for satisfactory completion of system testing.
 3. The status shall be reported to the appropriate review committee prior to proceeding to acceptance testing
 4. The customer shall participate in the review to ascertain that the maintenance release is ready to begin acceptance testing.

5- System Test (Cont.)

- Output: shall include the following
 1. Tested and fully integrated system
 2. Test report
 3. Test readiness review report

6- Acceptance Test

- Acceptance tests shall be conducted on a fully integrated system. Acceptance tests shall be performed by either the customer, the user of the modification package, or a third party designated by the customer.
- **Input:**
 1. Test readiness review report
 2. Fully integrated system
 3. Acceptance test plans
 4. Acceptance test cases
 5. Acceptance test procedures

6- Acceptance Test (Cont.)

- **Process:**

1. Perform acceptance tests at the functional level
2. Perform interoperability testing
3. Perform regression testing

- **Control:**

1. Execute acceptance tests
2. Report test results for the functional configuration audit (FCA)
3. Conduct functional audit
4. Establish the new system baseline
5. Place the acceptance test documentation under SCM control

6- Acceptance Test (Cont.)

- **Output:**
 1. New system baseline
 2. Functional configuration audit report (see IEEE Std 1028-1988)
 3. Acceptance test report (see IEEE Std 1042-1987)

7- Delivery

- **Input:** shall be the fully tested version of the system as represented in the new baseline.
- **Process:**
 1. Conduct a physical configuration audit
 2. Notify the user community
 3. Develop an archival version of the system for backup
 4. Perform installation and training at the customer facility

7- Delivery (Cont.)

- **Control:**

1. Arrange and document a physical configuration audit
2. Provide system materials for access to users, including replication and distribution
3. Complete the version description document (IEEE Std 1042-1987)
4. Complete updates to status accounting database
5. Place under SCM control

- **Output:**

1. PCA report (IEEE Std 1028-1988)
2. Version description document (VDD)