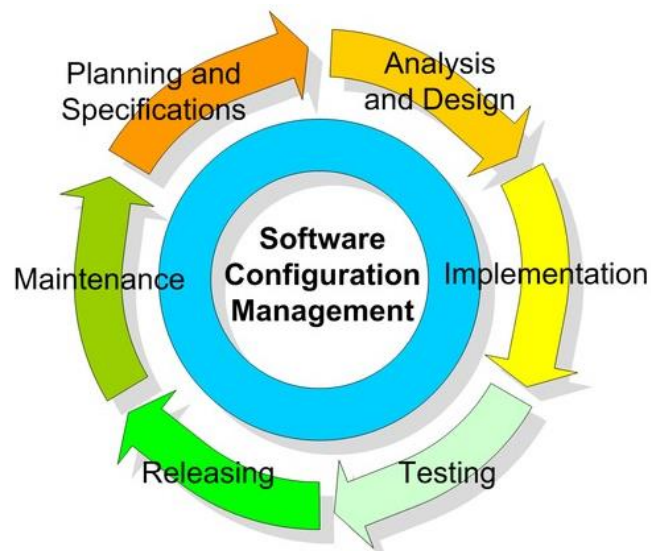


Software Configuration Management

- The process of identifying, organizing & controlling changes to the software during development and maintenance.
- SCM is support activity that makes technical & managerial activities are more effective.
- SCM operate throughout the software life cycle.
- A methodology to control & manage a software development project.
- **Purpose of SCM:**
 - Established and maintain the integrity of quality of work products.



- **MOM (Minutes of meeting):**
 - Project deliverables
 - Project artifacts
 - Work products
 - Work items
 - Project records / Outputs

- **Why we need SCM?**
 - ❖ To manage multiple people have to work on software that changing.
 - ❖ Project delivered in several releases (Builds).
 - ❖ Software must run on different machines & Operating systems.
- **Problems resulting from poor SCM:**
 - Can't roll back to the previous system.
 - One change over write another
 - Which code changes belongs to which version?
- **Using SCM:-**
 - ❖ Multiple project can be managed
 - ❖ View different things in one place
 - ❖ Rollback changes
 - ❖ KPA in SEI CMMI level – 2
- **Major activities:**
 - Configuration planning & setup
 - Configuration identification
 - Configuration baseline
 - Configuration management
 - Configuration release control
 - Configuration Audit
 - Control of customer properly
- **SCM:- How it is accomplished?**
 - Using version control tool ex. WinSCP, git,gitLab
 - What are the Configuration items identified for current project?
 - Test plan, Requirements, Design specifications, Test scenarios, test cases & code.

- What are the Non-Configuration items identified for current project?
- MOM's, Status report, summary report, mails etc;

SQA(Software Quality Assurance) & SQC(Software Quality Control)

- **SQA(Process & Audits):**

- Process oriented
- Preventing problem
- Continuous improvement
- Audits

- **SQC(Testing & reporting):**

- Product oriented
- Detecting problems
- Final check point before delivery.

Criteria	SQA	SQC
1. Definition	SQA is a set of definition for ensuring quality in software engineering process.	SQC is a set of activities for ensuring in software product.
2. Focus	Process oriented	Product oriented
3. Problems	Preventing	Detecting
4. Scope	Relates to all products that will ever be created by process.	Relates to specific product
5. Activities	<ul style="list-style-type: none">• Process definition and implementation• Audits• Training	<ul style="list-style-type: none">• Reviews• Testing
6. Example	Verification	Validation

❖ **Verification:** Checks whether we are building the right system

❖ **Validation:** Checks whether we are building the system right

- **Verification Strategies:**

- Requirement review
- Design review
- Code walkthrough
- Code review

- **Validation strategies:**

- Unit level testing
- Integration level testing
- System level testing
- Alpha testing
- User acceptance testing
- Beta testing

- **Verification Strategies:**

- ❖ **Requirement review:-**

- The study & discussion of the computer system requirements to ensure they meet stated user needs & are feasible.
 - Deliverable: Review statement of requirements (Approved SRS)

- ❖ **Design review:**

- The study & discussion of the computer system design to ensure it will support the system requirements.
 - Deliverable: Approved high level design & low level design includes DB, UI & UML diagrams.

❖ **Code Walkthrough:**

- Informal analysis of the program source code to find defects & verifying coding techniques.
- Deliverable: Software ready for initial testing by the developer.

❖ **Code Review:**

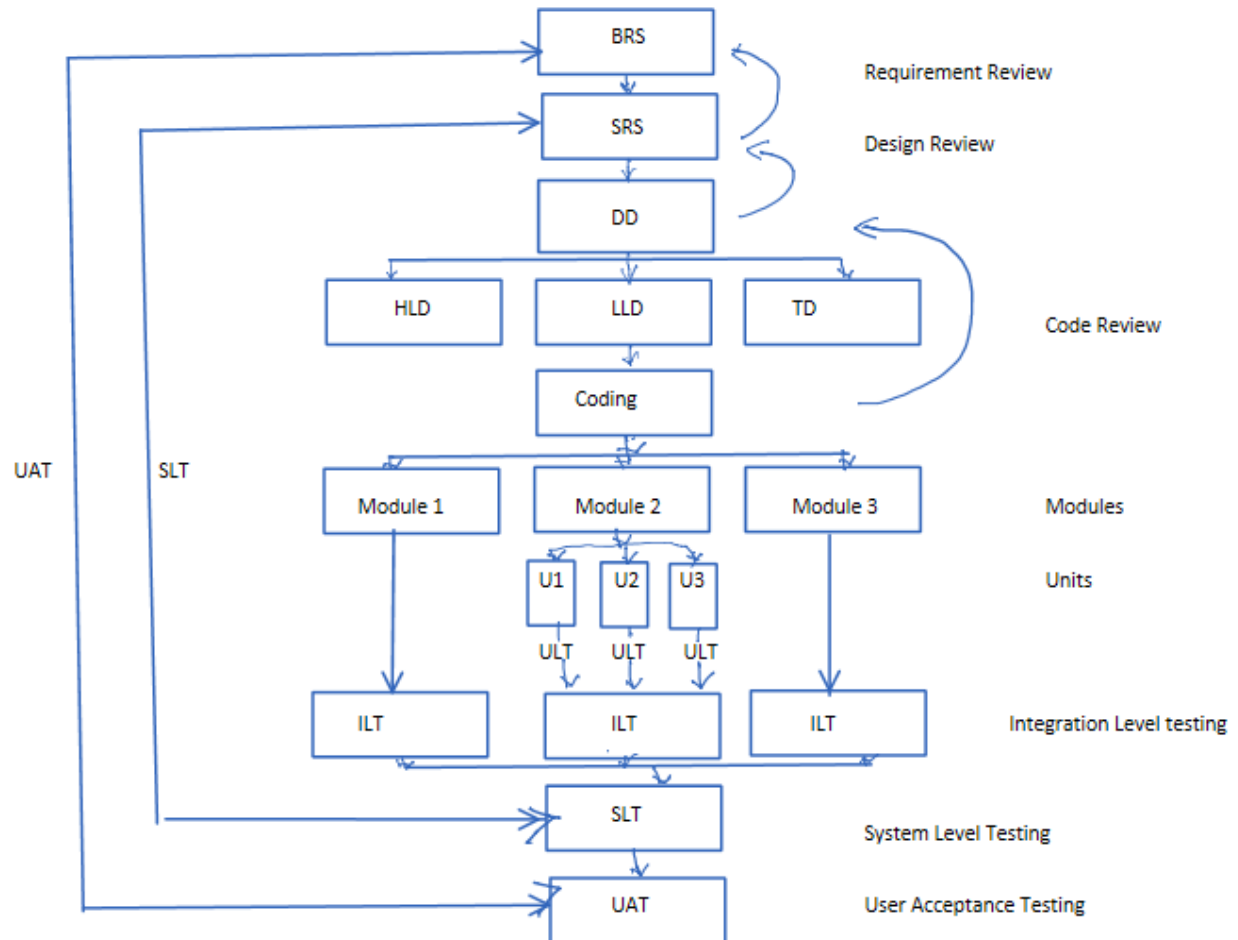
- Formal analysis of the program source code to find defect as defined by meeting system design specification.
- Deliverable: Software ready for testing by the testing team.

Validation strategies

- **Unit level testing**
 - Testing of single program, modules, or unit of code.
 - Deliverable: Software unit ready for testing with other system components.
- **Integration level**
 - Testing of related programs testing, modules, or unit of code.
 - Deliverable : System is ready for testing
- **System level testing**
 - Testing of entire computer system across all modules, this kinds of testing can be include functional & structural testing.

- Deliverable: Tested computer system based on what was specified to be developed.
- **Alpha Testing**
 - Testing of the whole computer system before rolling out to the UAT.
 - Deliverable: Stable application
- **User acceptance Testing**
 - Testing of computer system to make sure it will work in the system regardless of what the system requirements indicate.
 - Deliverable: Tested and accepted system based on the user.
- **Installation Testing**
 - Install and uninstall application using various operating system and browser.
 - Deliverable: Approved .exe
- **Beta Testing**
 - Testing of the application after the installation at the client place.
 - Deliverable: Successfully installed and running the application

Verification & Validation



Software Test Engineer Roles and Responsibilities:

- Understanding the requirement and functional specifications of the application.
- Identifying required test scenarios for project.
- Designing and preparing test case to validate application.
- Execute test cases to validate application.
- Log test result (How many test cases are pass or fails).
- Defect reporting and defect tracking.
- Retest fixed defect of previous build.
- Performed various types of testing assigned by **Test Lead** (Functionality, Usability, User Interface, Compatibility).
- **Reports to Test Lead about status of assigned tasks:**
 - Daily status report
 - Daily defect report
 - Weekly status report
 - Retesting report
 - Other assigned tasks
- Participated in regular team meeting by lead and manager.
- Creating automation scripts for regression testing.
- Provides enhancement of project based on end user perspective.
- Provides the recommendation on whether or not the application or system is ready for production.

Senior Software Test Engineer Roles and Responsibilities:

- Same as Software Test Engineer
- He is participated in review of test scenarios, test cases, and defects.
- Some time involved in preparation of test plan.
- Whenever the test lead on vacation, he will lead the team.

Testing Terminology:

- PMP: - Project Management plan
- BRS: Business requirement specifications
- SRS: Software requirement specifications
- FRS: Functional requirement specifications
- HLD: High level design
- LLD : Low level design
- LOC: Lines of code
- TP: Test Plan
- CR: Change request
- MOM's; Minutes of Meeting
- UI: User Interface
- TS: Test Scenarios
- TC: Test cases
- DSR: Daily status report
- DDR: Daily defect report
- WSR: Weekly status report
- RTM: Requirement Traceability Matrices
- STE: Software Test Engineer
- SSTE: Senior Software Test Engineer
- TL: Test Lead
- TM: Test Manager
- BA: Business Analyst

- PM: Project Manager
- SCM: Software Configuration management
- SQA: Software Quality Assurance
- SQC: Software Quality Control
- OID: Organizational Innovation & deployment
- ECP: Equivalent Class Partition
- BVA: Boundary Value Analysis