**Need and Types of Maintenance**

<https://www.geeksforgeeks.org/software-engineering-software-maintenance/>

**Version control :**

<https://www.geeksforgeeks.org/version-control-systems/>

**Re-engineering**

<https://www.geeksforgeeks.org/software-engineering-re-engineering/>

**Reverse Engineering**

<https://www.geeksforgeeks.org/software-engineering-reverse-engineering/>

**Project Management Concepts**

<https://www.geeksforgeeks.org/what-is-project-management/>

**Feasibility Analysis**

<https://www.geeksforgeeks.org/what-is-a-feasibility-study-how-to-conduct-one-for-your-project/>

**Project and Process Planning**

<https://www.geeksforgeeks.org/what-is-planning-in-project-management/>

**Software efforts, Schedule, and Cost estimations,**

<https://www.geeksforgeeks.org/cost-and-schedule-estimating-process/>

**Project Scheduling and Tracking**

<https://www.geeksforgeeks.org/project-schedule-tracking-tools-in-software-engineering/>

**Risk Assessment and Mitigation**

[https://www geeksforgeeks.org/short-note-on-risk-assessment-and-risk-mitigation/](https://www.geeksforgeeks.org/short-note-on-risk-assessment-and-risk-mitigation/)

**Software Quality Assurance (SQA)**

<https://www.geeksforgeeks.org/software-engineering-software-quality-assurance/>

**Software Configuration Management (SCM)**

Software Configuration Management (SCM) is a systematic approach to managing changes in software products. It ensures that the integrity and traceability of the software configuration are maintained throughout the software development lifecycle. Here is an overview of the key aspects and processes involved in SCM:

**Objectives of SCM**

* **Version Control**: Track and manage multiple versions of software components.
* **Change Control**: Ensure that changes are made systematically and with proper authorization.
* **Build Management**: Automate and manage the process of building software from source code.
* **Release Management**: Plan, schedule, and control the movement of releases to various environments.
* **Configuration Identification**: Identify and define the configuration items in a system.
* **Configuration Status Accounting**: Record and report the status of configuration items and changes.
* **Configuration Auditing**: Ensure that the configuration items conform to their specified requirements.

**Key Processes in SCM**

1. **Configuration Identification**
   * **Identify Configuration Items (CIs)**: Define and document the CIs, such as source code, documentation, test scripts, and hardware components.
   * **Baseline Establishment**: Create baselines, which are formally approved versions of CIs that serve as a reference point.
2. **Version Control**
   * **Versioning**: Assign unique version identifiers to different versions of CIs.
   * **Branching and Merging**: Create branches for parallel development and merge changes back into the main codebase.
3. **Change Control**
   * **Change Requests**: Document and submit requests for changes to CIs.
   * **Impact Analysis**: Assess the impact of proposed changes on the system.
   * **Change Approval**: Review and approve or reject change requests through a change control board (CCB).
   * **Implementation**: Implement approved changes and update the configuration.
4. **Build Management**
   * **Automated Builds**: Set up automated build processes to compile and package the software.
   * **Build Scripting**: Use build scripts to ensure consistency and repeatability of builds.
   * **Build Verification**: Verify that builds are successful and meet predefined quality criteria.
5. **Release Management**
   * **Release Planning**: Plan the release schedule and content.
   * **Release Packaging**: Package the software and related artifacts for distribution.
   * **Deployment**: Deploy the software to different environments (development, testing, production).
   * **Rollback Planning**: Prepare rollback plans in case of deployment failures.
6. **Configuration Status Accounting**
   * **Status Reporting**: Record the status of CIs and changes throughout the lifecycle.
   * **Metrics and Reporting**: Generate reports on configuration status, version histories, and change activities.
7. **Configuration Auditing**
   * **Audit Planning**: Schedule and plan configuration audits.
   * **Audit Execution**: Perform audits to verify that CIs conform to their documentation and requirements.
   * **Audit Reporting**: Document audit findings and follow up on corrective actions.

**Tools and Techniques in SCM**

* **Version Control Systems (VCS)**: Tools like Git, Subversion (SVN), and Mercurial for managing source code versions.
* **Build Automation Tools**: Tools like Jenkins, Travis CI, and Bamboo for automating the build process.
* **Change Management Tools**: Tools like Jira, Bugzilla, and ServiceNow for tracking change requests and issues.
* **Release Management Tools**: Tools like Octopus Deploy and Ansible for managing releases and deployments.
* **Configuration Management Databases (CMDB)**: Databases for storing information about configuration items and their relationships.

**Best Practices for SCM**

* **Consistency**: Maintain consistent practices and tools across the development team.
* **Automation**: Automate repetitive tasks like builds, tests, and deployments to reduce errors and improve efficiency.
* **Documentation**: Keep detailed documentation of configuration items, changes, and processes.
* **Communication**: Ensure clear communication among team members about changes and configuration management activities.
* **Regular Audits**: Conduct regular configuration audits to ensure compliance with requirements and standards.