

Configuration Management And Change control



Alan Estrada, Austin Odom, Manisha Devi, Christelle Mbemba.

What is Configuration Management?

Configuration Management is the process of maintaining systems, such as computer hardware and software, in a desired state. Configuration Management (CM) is also a method of ensuring that systems perform in a manner consistent with expectations over time. It helps organizations maintain consistency and integrity in the system, ensuring that all components are properly identified and maintained throughout their life cycle.

Importance of Configuration Management

Enhanced Stability : By implementing configuration management, organizations can achieve greater stability and reliability in their systems. They manage dependencies, libraries, and configurations, ensuring that the software can be built and deployed reliably. This helps in avoiding issues related to incompatible versions or missing dependencies.

Risk Mitigation : Proper configuration management helps in identifying and mitigating potential risks associated with system changes, thereby safeguarding against unexpected failures and security breaches.

Regulatory Compliance : Many industries have strict regulatory requirements related to configuration management. Adhering to these standards is crucial for legal compliance and maintaining the organization's credibility.

History of Configuration Management

1. Early Development : Configuration management practices can be traced back to the 1950s, primarily in industries such as aerospace and defense. The Focus was on managing complex hardware ensuring reliability.
2. Evolution in Software : In the 1980s, the concept of configuration management expanded to software development. It became crucial aspect of maintaining various versions and releases of software products.
3. Modern Applications : Today, configuration management extends beyond traditional IT systems. It is integral to DevOps practices, continuous integration and deployment in agile environments.

Configuration Management in the Department of Defense (DoD)

1.Stringent Requirements : The DoD places high emphasis on configuration management to ensure the reliability and security of its complex technological systems.

2.Integrated Approach : Configuration management is integrated into the entire lifecycle of defense systems, from initial development to deployment and ongoing maintenance.

3. Standardization : There are specific standards and protocols set by the DoD to govern configuration management processes across its diverse range the projects.

Configuration Management Standard of ANSI

ANSI : The American National standards Institute provides crucial guidelines for implementing configuration Management practices across industries and sectors.

Global Impact on ANSI : Many ANSI standards are used internationally, contributing to global trade and harmonization of industry practices.

Accessibility : ANSI standards are available to the public, allowing organizations to readily access and implement them.

For example: Imagine you're building a big LEGO castle together. To keep things organized and make sure everyone is on the same page, you create a plan. This plan includes rules on how to add new LEGO pieces, how to fix parts that break, and a checklist to track everything.

IEEE

IEEE standards are known for their technical rigor and are developed by experts in their respective fields. The institute of Electrical and Electronics Engineers is a major contributor to defining standards related to software and system configuration management.

- It helps in designed to be adaptable to different contexts and industry advancements.
- They encompass various areas such as telecommunications, power and energy and the development of new technologies.

ITIL

ITIL provides a framework for aligning IT services with the needs of the business and customers.

- ITIL include a set of practices for configuration Management, emphasizing the importance of maintaining accurate and up- to- date information about configuration items(cks) and their relationship within an IT environment.
- It emphasizes continual improvement and a focus on the overall business strategy.

How CM activities support products/projects

A small project may be relatively simple to document, make changes to, revert changes, and overall track the progress of throughout its lifespan through the use of basic methods familiar to us. However, as a project grows in scale, the cost of its development, the amount of risk involved, the number of stakeholders relying on its completion, etc. also grows along with it.

This means that more professional methods and tools will be required to effectively keep track of any and every decision, addition, and change that may occur throughout the product/project development process. Proper use of configuration management in these projects can assist in their completion through its implementation of:

- Traceability
- Proper identification and documentation of all systems and their elements
- Change control
- Audit trail

CM supporting products/projects - Traceability

For large projects, it's important to be able to trace back to right when a change was made to the software or other product. For example, in the event that a program no longer runs correctly/as intended after a series of updates, the program should be traceable to whenever it was running correctly to quickly uncover the error and fix it.

One of the key ideals of utilizing CM is establishing baselines (agreed upon attributes and components of a program or other product, of which satisfy certain requirements, implemented at a specific point in time) in order to recognize whether the product has achieved the goals intended by the developers, managers, and stakeholders.

Speed is a vital trait to have when working on a project that could be affecting the lives of several people who use the product. Without an effective method of tracing, an error can remain in a system for a prolonged period of time before it gets repaired, leading to customer dissatisfaction and other consequences.

“The customers come back and the products don't.”

CM supporting products/projects - Identification and Documentation

In order to establish and maintain the previously stated baselines(as well as other features throughout the development process), a proper method of identification and documentation is required, which can be implemented through the use of Configuration Identification(a discipline of CM) and Configuration Status Accounting.

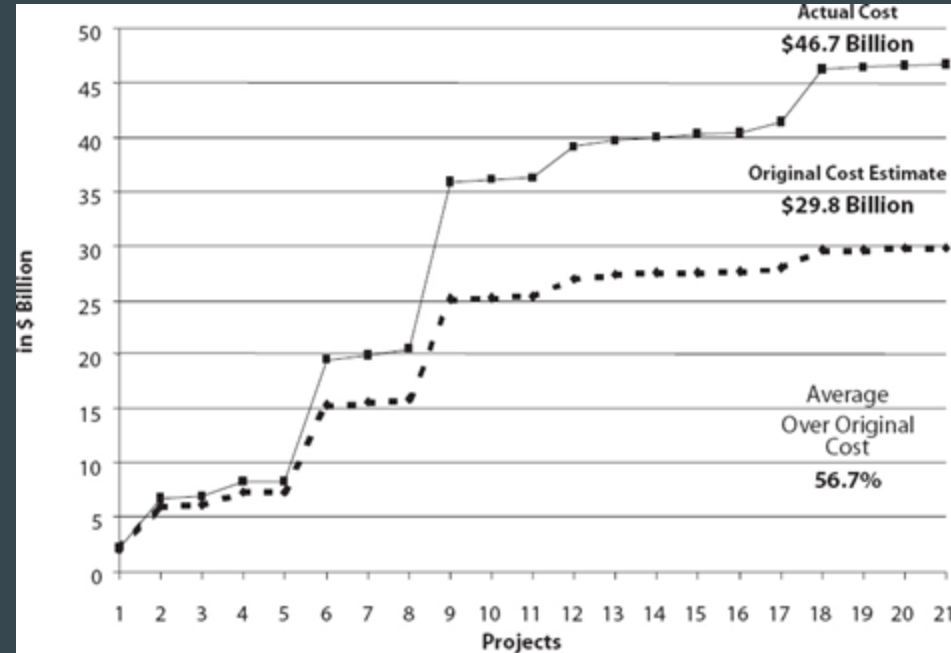
More in detail later, but the premise of CI is defining specific attributes of key Configuration Items, the significant decisions in making those items, identifiers that assist in tracking them, and more.

CSA is a method that involves the documentation of all details related to the CI process as well as details of changes made, including date and descriptions. This can be used to keep team members informed of components of a system and their status, even if they aren't directly involved in their creation, observe how changes have impacted the system as a whole, and even be able to provide documentation to stakeholders/ end-users for them to observe the history of a system or whether it will fulfill their needs.

CM supporting products/projects - Change Control

Change or deviation from the currently set path is typically not possible to be planned for, and may occur suddenly with little preparation. This sense of uncertainty can lead to added complexities and costs to a project, not just in terms of what will need to be added, but also any unforeseen delays directly cause by these changes. This is why change control and management is necessary for any sort of development process.

“Even the best set of design plans and detailed contract specifications are no guarantee that a particular construction project will not experience numerous changes.” - John E. Canepari, Sr. Vice President, PACO Group, Inc.



CM supporting products/projects - Change Control

Configuration management complements the accommodation of changes and the perpetual documenting of configurations and reconfigurations made to a system. Multiple steps are taken as to ensure that changes that need to be made will not compromise the integrity and functionality of a system. Maintaining useful baselines before and after changes are made will also assist in identifying roadblocks caused by the changes or the absence of them, along with in general tracking the status of the project in the long term

Even if a sudden change can't always be accounted for on the spot, having a well-defined measure for implementing it can help save on time and resources spent within the project.



CM supporting products/projects - Audit Trail

Audits and heavy documentation may seem contradictory of the AGILE engineering process, however, they are still completely necessary in identifying and defending against potential failures and faults that may/will occur throughout product lifespan.

Measures must be taken to evaluate the status of configuration items and determine whether activities conform to the advised CM process. The methods described previously and ones that will be described all complement the use of audits to verify if components and practices are satisfactory enough for the final goal of the project.

Documentation, for example, of all changes, decisions, and testing of a work product can be examined during an audit to guide not just team members, but also stakeholders and inform them that the product is either meeting standards defined in previous planning activities, or still need to be improved upon.

Tasks in CM - Planning and Management

Definition - A comprehensive description of the roles, responsibilities, policies, and procedures that apply when managing the configuration of products and systems.

Its purpose is to provide a schedule and a structured framework for implementing effective CM practices for tasks such as:

- Identifying, defining, and baselining configuration items (CI)
- Controlling modifications and releases of CIs
- Reporting and recording the status of CIs and modifications
- Ensuring completeness, consistency, and correctness of CIs
- Controlling storage, handling, and delivery of the CIs

We will talk about each of these tasks in detail, but a CM plan must be implemented for effective project and product management.

Planning and Management Cont.

Here is an example of what a sample
CM Plan might look like

Sample Configuration Management Plan (CMP) Outline

- INTRODUCTION
 - Purpose
 - Background
 - Scope
 - Document Overview
 - References
- SYSTEM OVERVIEW
- CONFIGURATION MANAGEMENT COMPONENTS
 - Organizations and Responsibilities
 - Configuration Identification
 - Configuration Change Control
 - Configuration Status Accounting
 - Configuration Reviews
- CONFIGURATION MANAGEMENT PROCESS
 - Process Overview
 - Classification
 - Evaluation
 - Modeling and Testing
 - Implementation
- NEXT STEPS
 - Plan CM Implementation
 - Establish CM Folders
 - Identify CM Tools
- Establish Change Control Board
- APPENDIX A FORMS
- APPENDIX B CHARTER
- APPENDIX C REFERENCES
- APPENDIX D APPROVAL

Tasks in CM - Configuration Identification

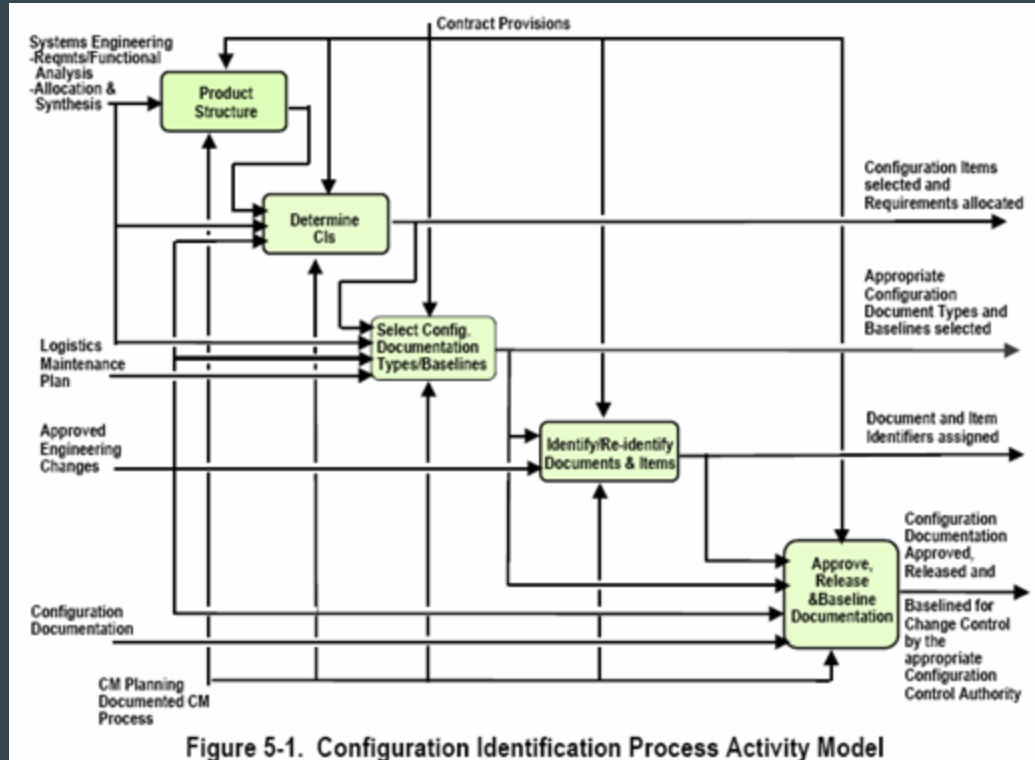
Definition - Configuration identification focuses on identifying the configuration items (CIs) that will be managed and controlled under a CM process.

The CM identification process includes:

- Selecting CIs at appropriate levels of the project to facilitate the documentation, control, and support of the items.
- Documenting each CIs performance, functionality, internal and external interfaces.
- Issuing identifiers or labels for each CI to provide unique identities to make easily traceable.
- Define criteria for when a baseline should be established*
- Modifies identifiers to reflect major changes.
- Distinguishing the difference in product versions

Other CM activities rely on effective configuration identification. For example, if CIs and their associated documentation are not properly identified, it will be impossible to control the changes to the items. Inaccurate or incomplete configuration documentation may result in defective products, schedule delays, and/or higher maintenance costs after delivery.

Configuration Identification Cont.



Tasks in CM - Configuration Control

Definition - Procedures to control changes to configuration items (CIs). It involves processes for requesting, evaluating, approving, and implementing changes while maintaining proper documentation and ensuring changes are tracked.

When a CI is deemed ready for release it is labeled as a baseline. Configuration control involves the creation, modification, and approval of baselines.

Baselines represent the current approved version at a specific point in time in the system life cycle.

Baselines ensure that stakeholders have a reference point for tracking changes of the configuration.

Version Control is a specific type of configuration control that focuses on managing changes to source code, documents, and other files in software development. It allows to revert to previous versions, compare changes made, and effective collaboration.

While version control and configuration control are very similar, they are different things. It can be said that version control is a subset of configuration control, and happens at the software level.

Configuration Control Cont.

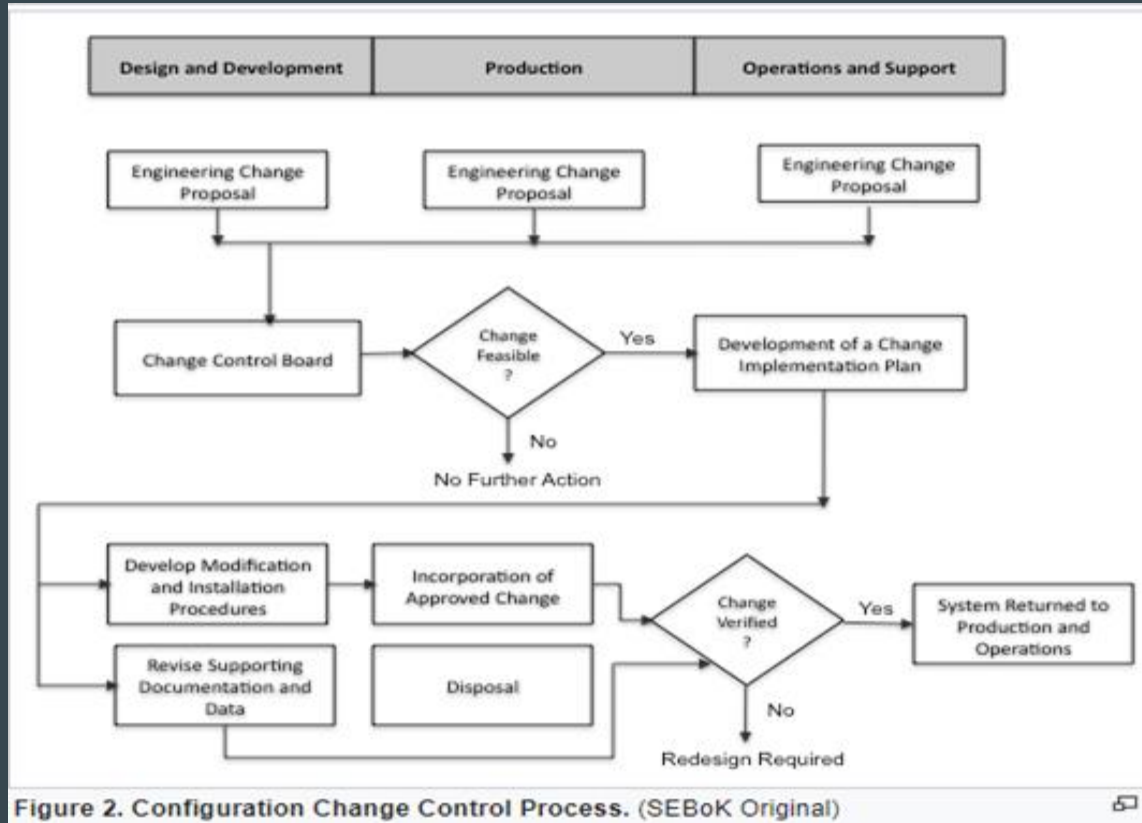


Figure 2. Configuration Change Control Process. (SEBoK Original)



Tasks in CM - Status/Reporting

Definition - Configuration status and reporting ensures that all CIs and their associated baselines are recorded and documented throughout its life cycle from testing, to production, to deployment.

Configuration status and reporting keeps track of the following:

- New items received
- Installation of items
- Transition from testing to production
- Retired or disposed items
- Unauthorized CIs and version changes of CIs

The primary purpose of configuration status reporting is to communicate the state of CIs, baselines, and changes made to stakeholders. It allows for stakeholders to be informed, track the progress, and identify any issues.

Status/Reporting Cont.

Status reports should contain the following:

- Modifications to key attributes of the CI such as status(system shutdowns), version numbers, serial numbers, installation dates, etc.
- Policy changes related to finance, procurement, contract management, and security if needed.
- Notifying stakeholders of key events such as accepted items, installation of items, lost or stolen items.
- Confirming that all data documentation is correct and complete according to CM policies and standards.
Ensuring that any changes made or version updates are authorized.
- Creating incident reports when unauthorized changes or incorrect and incomplete documentation is detected.
- Distributing reports to stakeholders.

Status reporting is a crucial step in CM providing stakeholders with detailed insight about a project and provides effective communication and continuous improvement of the state of the CIs.

Tasks in CM - Verification/Audit

Definition - Verification and auditing is responsible for ensuring that information in CM is accurate and that all CIs are identified and recorded. This process can be conducted manually or using automated tools.

The objective of verification and auditing is to detect and manage all exceptions to configuration policies, processes, and procedures, including security and license use. The verification process ensures that configuration records are accurate and complete, and that any changes are approved.

Verification/Audit Cont.

The main difference in configuration status reporting and auditing are as follows:

- Status reporting focuses on specific aspects of the system such as progress, performance, status, risks, and recommendations for change. Whereas auditing focuses on effectiveness, compliance, and adherence of processes against established standards, policies, and regulations.
- Status reporting may occur regularly (daily, weekly, monthly) or as needed. Auditing occurs periodically (annually, biennially) or in response to specific events such as before and after major releases.
- Status reports are directed to the stakeholders involved in the project. Auditing is directed to management, stakeholders, and regulatory authorities to provide assurance, ensure compliance, and to drive improvement.

Verification/Audit Cont.

Configuration verification and auditing activities include:

- Verifying that services and products are built and documented according to requirements, standards, and contractual agreements.
- Verifying that the correct and authorized versions of any CI exist and are correctly identified and described.
- Check that release documentation and configuration administration are present before making a release.
- Confirm that the current environment is as expected and documented, and any change requests are resolved.
- Verify that CI specifications are in compliance with policies and baselines.
- Validate required documentation such as maintenance contracts, licenses, and warranties.
- Check data quality for accuracy and completeness.
- Initiate incidents for discovered unauthorized changes

An unauthorized change may include:

- Unauthorized software installed.
- Unauthorized access to resources and services (not having access rights).
- Discrepancy of configuration details compared to the actual status.

Tasks in Software Configuration Management (SCM)

Definition - In Software Engineering, Software Configuration Management (SCM) is a process to systematically manage, organize, and control changes in the documents, codes, and other entities during the software development life cycle.

The primary goal is to increase productivity with as few mistakes as possible. SCM is a sub-set of configuration management and it can accurately determine who made which revision.

SCM extends beyond the scope of generic CM by focusing on changes to software artifacts and addressing challenges of software development.

Tasks in Software Configuration Management

- Configuration Identification
- Baselines
- Change Control
- Configuration Status Accounting
- Configuration Audits and Reviews

The tasks of SCM are a subset of generic configuration management but are adapted to focus more on the software development life cycle.

Tasks in SCM Cont.

Configuration Identification - Determining the scope of the software system. Describing a type of Computer Software Configuration item (CSCI) with a project identifier and version information.

Activities include:

- Identification of CSCIs like source code, test cases, and requirements.
- Identification of each CSCI in a repository, using an object-oriented approach.
- Grouping objects into a collection of objects.
- Every object has its own features that identify its name that is different from all other objects
- List of resources required such as the document, the file, tools, etc.

Example: Instead of naming a file login.php its should be named login_v1.2.php where v1.2 stands for the version number

Example: Instead of naming a folder “Code” it should be name “Code_D” where D represents that it should be backed up daily.

Tasks in SCM Cont.

Baselines - A formally accepted version of a CSCI. Designated and fixed at a specific time while conducting the SCM process. Can only be changed through change control procedures.

Activities include:

- Construction of various versions of an application
- Defining and determining mechanisms for managing various versions of work products
- The functional baseline corresponds to the reviewed system requirements
- Widely used baselines include functional, developmental, and product baselines (more on baselines later)

In simple terms, a baseline just means ready for release.

Tasks in SCM Cont.

Change Control - A procedural method which ensures quality and consistency when changes are made to a CSCI. A change must be requested to a software configuration manager.

Activities include:

- Controlling changes to ensure stable software development. Changes are committed to a repository
- The request will be checked based on technical quality, possible side effects and overall impact on other CSCIs.
- Managing changes and making CSCIs available during the software life cycle.
- Change implementation
- Change testing
- Change review
- Change deployment
- Change documentation
- Change monitoring and feedback

Tasks in SCM Cont.

Configuration Status Accounting - Tracks each release during the SCM process. This stage involves tracking what each version has and the changes that lead to this version.

Activities include:

- Keeping records of all the changes made to a previous baseline to create a new baseline
- Monitor status of change requests
- A complete listing of all changes since the last baseline
- Allows tracking of progress to the next baseline
- Allows testing of previous versions

Tasks in SCM Cont.

Configuration Audits and Reviews - Verify that all the software products satisfies the baseline needs. It ensures that what is built is delivered.

Activities include:

- Configuration auditing is conducted by auditors by checking that defined processes are being followed and ensuring that the SCM goals are met.
- Verifying compliance with control standards.
- SCM audits ensure traceability is maintained during the process.
- Validation of completeness and consistency of the software.

BaseLines in CM

- A baseline a snapshot of a system 's configuration at a specific point in time.
- It serves as reference point for managing changes and ensuring consistency through the development process.
- There is three type of Baselines:
 - Functional Baseline
 - Allocated Baseline
 - Product Baseline

Functional Baseline

- It define the functional and performance characteristics of the system.
- It capture the requirement specified by the customer or stakeholders.
- This baseline ensure that the system meets the intended purpose and functionality.

Allocated Baseline

- The Allocated Baseline translates the functional requirement into allocated requirement for each subsystem.
- It specifies how the system 'functionality will be distributed across different elements
- This baseline guides the development team in designing and implementing the system architecture

Product Baseline

- The product baseline represents the completed system configuration ready for delivery to the customer.
- It includes all the developed components, documentation, and supporting materials
- This baseline serves as the basis for validation, verification and acceptance testing.

Role in the CM process

Configuration Management (CM) relies on the collaboration and coordination of various roles.

There are two key roles in the CM process:

- the Product Manager.
- the Systems Engineer.

Both the Product Manager and the Systems Engineer collaborate closely throughout the development lifecycle.

System Engineer

The Systems Engineer is responsible for designing and implementing the system architecture.

They translate user requirements into technical specifications and oversee the integration of subsystems.

In CM, Systems Engineers play a vital role in managing configuration baselines, ensuring consistency, and assessing the impact of changes on the overall system.

Product Manager

The Product Manager serves as the primary liaison between the development team and stakeholders.

Responsibilities include defining product requirements, prioritizing features, and setting release schedules.

In the context of CM, the Product Manager ensures that configuration changes align with business objectives and customer needs.

Types of tools commonly used in CM

Configuration Management (CM) relies on various tools to facilitate the management of software and system configurations.

There are four types of tools commonly used in CM:

- Library Management Tools
- Tracking and Change Management Tools
- Version Management Tools
- Release Management Tools.

Library Management

Library Management Tools are software applications designed to manage and organize configuration items (CIs) and related documentation within a centralized repository.

These tools provide version control capabilities, allowing teams to track changes, revisions, and updates to CIs over time.

Library Management Tools typically support various types of files, including source code, design documents, requirements specifications, test scripts, and configuration files.

Examples :

Apache Subversion (SVN), GitLab, and Bitbucket.

Tracking and change Management Tools

Tracking and Change Management Tools facilitate the tracking of changes to configuration items throughout the development lifecycle.

These tools enable teams to create, track, and prioritize change requests, issues, and defects.

They also support workflow automation, collaboration, and communication among team members.

Examples:

Jira, Trello, and Microsoft Azure DevOps.

Versions Management Tools

Version Management Tools are essential for managing different versions of configuration items.

They enable teams to track changes, compare versions, and roll back to previous states if needed.

Version management tools help maintain the integrity and consistency of configurations across the development lifecycle.

Examples:

Git, Mercurial, and Perforce, svn

Release Management Tools

Release Management Tools facilitate the planning, scheduling, and deployment of software releases.

These tools help manage release cycles, coordinate activities across development, testing, and operations teams, and ensure smooth deployment.

Release management tools often integrate with other CM tools to streamline the release process.

Examples :

Jenkins, Bamboo, and GitLab CI/CD.

When selecting CM tools, consider factors such as:

- Scalability and flexibility to accommodate the organization's needs.
- Integration capabilities with existing tools and systems.
- User interface and ease of use for team members.
- Support for collaboration, automation, and reporting.

Sources

https://csrc.nist.gov/glossary/term/configuration_management_plan#:~:text=Definitions%3A,configuration%20of%20products%20and%20systems.

https://acqnotes.com/acqnote/careerfields/configuration-management-plan#google_vignette

<https://www.product-lifecycle-management.com/mil-hdbk-61a-5-1.htm>

<https://www.linkedin.com/advice/3/what-difference-between-version-control#:~:text=Version%20control%20helps%20you%20manage,activities%20that%20configuration%20management%20involves>

https://sebokwiki.org/wiki/Configuration_Management#Change_Control

https://docs.microfocus.com/SM/9.51/Hybrid/Content/BestPracticesGuide_PD/ConfigurationManagementBestPractice/Configuration_Status_Accounting_Reporting.htm

<https://www.guru99.com/software-configuration-management-tutorial.html>

<https://www.ibm.com/topics/configuration-management#:~:text=Tailored%20configuration%20management%20%28CM%29%20practices%20enable%20system%20administrators,control%20and%20prevent%20configuration%20drift%20and%20unnecessary%20downtime.>

Sources

https://www.plm.automation.siemens.com/media/global/en/CIMdata%20White%20Paper%20-%20Multi-Disciplinary%20BOM%20and%20Configuration%20Management_tcm27-90362.pdf

https://content1.dau.edu/DAUMIG_se-brainbook_189/content/Management%20Processes/Configuration-Management.html

<https://www.cmcrossroads.com/article/how-implement-cm-and-traceability-practical-way>

<https://plmadvisors.com/plm-and-configuration-management-best-practices-part-traceability-and-revisions/>

<https://www.pmi.org/learning/library/2019/04/07/15/37/configuration-management-help-controlling-changes-7842#:~:text=The%20key%20benefits%20of%20configuration%20management%20are%20as,assure%20that%20project%20performance%20criteria%20are%20being%20met.>

<https://www.theknowledgeacademy.com/blog/change-control-vs-configuration-management/>

<https://cmstat.com/cmsights-news-posts/configuration-management-audits>