**CHAPTER 2**

**PLANNING FOR SOFTWARE CONFIGURATION MANAGEMENT (SCM)**

**Chinecherem Umezuruike Ph.D., and Kayode Opadoyin**

[**chinecherem.umezuruike@bowen.edu.ng**](mailto:chinecherem.umezuruike@bowen.edu.ng)**, kayoed.opadoyin@bowen.edu.ng**

**Introduction**

Planning for Software Configuration Management is an integral component of software development, emphasizing the management and regulation of alterations to software artifacts during the development life cycle. It involves strategically delineating the processes, tools, and resources necessary to ensure the efficient management of software configurations.

The main objective of Planning for Software Configuration Management is to establish a methodical approach to overseeing software configurations to attain consistency, traceability, and regulation over software artifacts. Through the development of a comprehensive plan, organizations can streamline development processes, enhance collaboration among team members, and elevate the overall quality of software products.

Effective Planning for Software Configuration Management provides various advantages, including:

1. Ensuring the uniformity and soundness of software configurations - Fostering collaboration and communication among team members
2. Enhancing traceability and auditability of software artifacts
3. Improving overall software quality and productivity
4. Facilitating efficient change management and release procedures

This chapter delves into the development of a Software Configuration management plan after understanding that planning for Software Configuration Management is imperative for establishing a robust groundwork for managing software configurations proficiently throughout the software development life cycle. The chapter went ahead to explain the components, purpose and development procedures of SCMP. Sample outlines and examples were included.

**Learning Outcome**

After studying the content provided in this chapter, participants will be able to:

1. Explain the concept of SCMP and establish its components and purpose to SDLC.
2. Develop a comprehensive Software Configuration Management plan
3. Understand and explain the procedures in developing an SCMP

**2.1 Developing Software Configuration Management Plan**

Developing a software configuration management plan involves mapping software development stages to specific configurations, selecting the appropriate stage-specific configurations, and executing the application based on these configurations. Software configuration management is crucial throughout the software life cycle, controlling changes and reducing their impact, especially in complex systems with frequent updates. A high assurance secure product requires a detailed Configuration Management (CM) plan to ensure integrity, track changes, and authorize modifications only to configuration items. Factors like Requirements Management, Change Management, and Release Management play significant roles in the effectiveness of configuration management processes, as highlighted in a study involving system engineers and project managers. Software Configuration Management (SCM) is essential for planning, organizing, and coordinating software changes to maintain quality and integrity throughout development, integration, and transfer, emphasizing the critical role of SCM in project success.

**2.1.1 Components of SCMP**

The important components of SCMP are:

1. Introduction: This section provides a comprehensive overview of the SCM plan, its fundamental purpose, and the extent of its scope.
2. Configuration Management Process: This section explains the methodologies and procedures governing the management of software configuration, including version control, change management, and release management.
3. Configuration Items: This part identifies the essential software elements slated for management and regulation, such as source code, libraries, documentation, and associated files.
4. Version Control: This section explains the tools and protocols pertinent to version control that are to be employed for overseeing alterations to the software and monitoring distinct versions of its components.
5. Change Management: This section expounds upon the protocols governing the handling of change requests, encompassing the review, approval, and implementation of changes, alongside the assessment of their impact.
6. Release Management: This part expounds on the procedures for packaging, disseminating, and deploying software releases, encompassing the development of release notes and the management of dependencies.
7. Configuration Audits: This segment elaborates on the methodologies for conducting configuration audits to validate the accuracy and completeness of the software configuration.
8. Tools and Infrastructure: This section specifies the tools, frameworks, and infrastructure earmarked for supporting the software configuration management processes.
9. Roles and Responsibilities: This part defines the specific roles and responsibilities assigned to team members engaged in the software configuration management process, inclusive of configuration managers, developers, and quality assurance personnel.
10. Training and Documentation: This segment delineates the requisite training criteria for team members engaged in software configuration management and elucidates the documentation to be generated within the process.
11. Approval and Review: This section provides a detailed account of the procedure for endorsing and evaluating the SCM plan, encompassing the stakeholders involved and the frequency of reviews.

**2.1.2 Purpose of SCMP**

The objective of a Software Configuration Management (SCM) plan is to establish a framework of processes and procedures for overseeing the configuration of software throughout its developmental life cycle. it assumes a crucial role in ensuring that software development projects are well-structured, regulated, and successful by offering a structure for effectively managing software configuration. The objectives are summarized below:

1. Consistency and Regulation: It aids in guaranteeing that software elements are managed and regulated efficiently during the developmental phase. It sets forth standards for version control, change management, and release management to uphold consistency and excellence.
2. Risk Mitigation: Through delineating procedures for managing alterations, conducting audits, and overseeing releases, the SCM plan helps in alleviating risks linked with software development, such as the introduction of errors, disparities, or unauthorized modifications.
3. Traceability: The plan permits the tracking of modifications made to software elements, enabling teams to monitor the progression of the software, pinpoint the rationales behind specific alterations, and comprehend the repercussions of those modifications.
4. Collaboration: The SCM plan fosters collaboration among team members by offering precise directives for collaborating on shared software elements, coordinating modifications, and effectively communicating about configuration management duties.
5. Efficiency: By rationalizing procedures for version control, change management, and release management, the SCM plan aids in enhancing efficiency in software development, diminishing errors, redundancies, and delays.
6. Compliance: The plan guarantees that software development adheres to organizational norms, industry regulations, and top practices for configuration management.
7. Quality Assurance: Through implementing stringent processes for overseeing software configuration, the SCM plan aids in upholding the quality of software products and eases testing, validation, and verification activities.

**2.1.3 SCMP Development Procedures**

The formulation of a Software Configuration Management Plan (SCMP) encompasses a variety of essential procedures to guarantee its efficacy and thoroughness. By adhering to these development procedures, organizations can design a comprehensive and efficient Software Configuration Management Plan to direct their configuration management endeavors throughout the software development life cycle. Several steps are involved in the development of an SCMP:

1. Scope Definition: Comprehend the scope of the software project and the specific configuration items necessitating management. This encompasses the identification of software components, documents, databases, hardware, and other configuration items mandating management.
2. Stakeholder Involvement: Engage crucial stakeholders from diverse teams participating in the software development process, including developers, project managers, quality assurance personnel, and system administrators. Their input is vital to ensuring that the SCMP mirrors the needs and prerequisites of all parties involved.
3. Gather Requirements: Amass requirements linked to version control, change management, release management, and other facets of configuration management. This involves grasping the organization's standards, industry best practices, and any project-specific requirements.
4. Define Processes: Chronicle the processes and protocols for governing software configuration. This encompasses setting up criteria for version control, change control, release management, audit procedures, as well as delineating roles and responsibilities.
5. Select Tools: Pinpoint the tools and infrastructure essential for supporting the configuration management procedures. This may entail opting for version control systems, issue tracking systems, build and release management tools, and other software tools indispensable for configuration management.
6. Draft the Plan: Compose the SCMP document, encompassing all data gathered from preceding steps. The draft should comprise an introduction, descriptions of configuration management processes, identification of configuration items, version control procedures, change management procedures, release management procedures, configuration audit processes, details of tools and infrastructure, delineation of roles and responsibilities, plans for training and documentation, and the process for review and approval.
7. Review and Refinement: Scrutinize the draft SCMP with pertinent stakeholders to obtain feedback and integrate any essential alterations or enhancements. This guarantees that the plan aligns with the organization's needs, industry standards, and project requisites.
8. Approval: Upon refining the SCMP based on feedback, pursue formal approval from relevant stakeholders, encompassing project sponsors, managers, and other pertinent authorities. This culminates in finalizing the plan and facilitating its execution.
9. Training and Communication: Subsequent to approval of the SCMP, deliver training to team members involved in configuration management procedures. Furthermore, disseminate the SCMP to all relevant personnel to ensure universal comprehension of the procedures and respective roles in configuration management.
10. Ongoing Maintenance: The SCMP necessitates regular review and updating as the project advances and new configuration management needs emerge. Additionally, it's imperative to certify that the plan continues to be in sync with any alterations in organizational standards, industry best practices, or project requirements.

**2.1.4 SCMP Outline Samples and Examples**

**Sample A**

1. Introduction: The objective of the Software Configuration Management (SCM) Plan will be expounded upon in this section. The boundaries of Configuration Management will also be delineated, alongside an outline of the Software Project.
2. Configuration Management Process: This section will delve into the elucidation of the various processes involved in Configuration Management. It will provide insights into the methodologies for Version Control, Change Management, and Release Management.
3. Configuration Items: The focus will be on the identification and categorization of Configuration Items (CIs). Additionally, the protocols for managing Documentation and Artifacts will be discussed.
4. Version Control: Detailed discussions will be conducted on the tools and procedures utilized for Version Control. Furthermore, the strategies for Branching and Merging, as well as the Versioning Scheme, will be examined.
5. Change Management: The Change Control Board (CCB) Processes will be elaborated upon, alongside the procedures for handling Change Requests. The Impact Analysis and Approval Process will also be scrutinized.
6. Release Management: This section will cover the procedures for Packaging and Deployment of Releases. Moreover, it will address the creation of Release Notes and the management of Dependencies and Release Versions.
7. Configuration Audits: The Configuration Audit Processes will be outlined, including various types of Audits such as Functional Configuration Audit and Physical Configuration Audit. The reporting of Audits and subsequent Corrective Actions will also be discussed.
8. Tools and Infrastructure: The tools employed for Configuration Management will be detailed, along with the Development Environments utilized. Additionally, the Automated Build and Deployment Tools will be highlighted.
9. Roles and Responsibilities: The distinct Responsibilities of Configuration Managers, Developers, and Quality Assurance Personnel will be expounded upon in this section.
10. Training and Documentation: This section will delineate the Training Requirements for Configuration Management. Furthermore, guidelines for Documenting Configuration Changes and procedures for Maintenance of the Configuration Management Plan will be provided.
11. Approval and Review: The Approval Process for the SCM Plan will be discussed, alongside the frequency of Reviews and Updates. The stakeholders involved in the Review of the Plan will also be identified.

**Sample B**

**Introduction:**

1. Purpose of the SCM Plan
2. Scope of Configuration Management
3. Overview of the Software Project

**Configuration Management Process:**

1. Description of Configuration Management Processes
2. Procedures for Version Control
3. Procedures for Change Management
4. Procedures for Release Management

**Configuration Items:**

1. Identification of Configuration Items (CIs)
2. Categorization of CIs
3. Handling of Documentation and Artifacts

**Version Control:**

1. Version Control Tools and Procedures
2. Branching and Merging Strategies
3. Versioning Scheme

**Change Management:**

1. Change Control Board (CCB) Processes
2. Change Request Handling Procedures
3. Impact Analysis and Approval Process

**Release Management:**

1. Release Packaging and Deployment Procedures
2. Release Notes Creation
3. Managing Dependencies and Release Versions

**Configuration Audits:**

1. Configuration Audit Processes
2. Types of Audits (e.g., Functional Configuration Audit, Physical Configuration Audit)
3. Audit Reporting and Corrective Actions

**Tools and Infrastructure:**

1. Configuration Management Tools Used
2. Development Environments
3. Automated Build and Deployment Tools

**Roles and Responsibilities:**

1. Responsibilities of Configuration Managers
2. Responsibilities of Developers
3. Responsibilities of Quality Assurance Personnel

**Training and Documentation:**

1. Training Requirements for Configuration Management
2. Guidelines for Documenting Configuration Changes
3. Configuration Management Plan Maintenance Procedures

**Approval and Review:**

1. Approval Process for the SCM Plan
2. Review and Update Frequency
3. Stakeholders Involved in Plan Review

**Example 1: Detailed Outline of SCMP for the Development of Sales Portal**

**Introduction**

* **Purpose:** To outline the objectives and processes for managing the configuration of the sales portal software throughout its development life cycle.
* **Scope:** Define the boundaries within which the SCM Plan will apply, specifying the sales portal project and the configuration items included.
* **Objectives:** List the key objectives of the SCM Plan, such as ensuring version control, managing changes efficiently, and ensuring compliance with standards.
* **Overview of the Sales Portal Project:** Briefly describe the goals, features, and stakeholders of the sales portal project.
* **Document Structure:** Provide an overview of the main sections and subtopics covered in the SCM Plan.
* **Key Assumptions and Constraints:** Highlight any assumptions or constraints that may impact the implementation of the plan.
* **Revision History:** Track changes made to the SCM Plan over time.

**Configuration Management Process**

* **Description of Processes:** Explain the configuration management processes to be followed, including version control, change management, and release management.
* **Version Control Procedures:** Detail how version control will be maintained for all software components of the sales portal.
* **Change Management Procedures:** Provide guidelines for requesting, assessing, and implementing changes to the sales portal.
* **Release Management Procedures:** Define how release packages will be created, tested, and deployed.

**Configuration Items**

* **Identification of CIs:** List and categorize the configuration items of the sales portal, including software components, databases, and documentation.
* **Categorization of CIs:** Organize configuration items by type and importance.
* **Documentation Management:** Specify how documentation and artifacts will be managed as configuration items.

**Version Control**

* **Version Control Tools:** Identify the tools and systems used for version control.
* **Branching and Merging Strategies:** Define the branching and merging strategies to be followed.
* **Versioning Scheme:** Explain the versioning scheme used for the sales portal software.

**Change Management**

* **Change Control Board:** Describe the composition and responsibilities of the Change Control Board.
* **Change Request Procedures:** Outline the process for submitting, evaluating, and approving change requests.
* **Impact Analysis:** Detail how the impact of changes will be analyzed and communicated.

**Release Management**

* **Release Packaging Procedures:** Document how releases will be packaged, tested, and deployed.
* **Release Notes:** Explain how release notes will be created to communicate changes in each release.
* **Dependency Management:** Address how dependencies between components will be managed in releases.

**Configuration Audits**

* **Audit Processes:** Specify the configuration audit processes to ensure the integrity of the sales portal software.
* **Types of Audits:** Define different types of audits such as functional configuration audits and physical configuration audits.
* **Audit Reporting:** Outline how audit findings will be reported and corrective actions implemented.

**Tools and Infrastructure**

* **Configuration Management Tools:** List the tools and infrastructure used for configuration management.
* **Development Environments:** Describe the development environments and their configurations.
* **Automation Tools:** Identify any automated tools used for build and deployment processes.

**Roles and Responsibilities**

* **Configuration Managers:** Define the responsibilities of configuration managers in managing the sales portal configuration.
* **Developers:** Clarify the roles and responsibilities of developers regarding configuration management practices.
* **Quality Assurance Personnel:** Detail the responsibilities of QA personnel in ensuring configuration quality.

**Training and Documentation**

* **Training Requirements:** Specify the training needed for team members on configuration management processes.
* **Documentation Guidelines:** Provide guidelines for documenting configuration changes and processes.
* **Plan Maintenance:** Detail the procedures for maintaining and updating the SCM Plan as needed.

**Approval and Review**

* **Approval Process:** Define the process for seeking approval from stakeholders for the SCM Plan.
* **Review Frequency:** Specify how often the SCM Plan will be reviewed and updated.
* **Stakeholders Involved:** Identify the stakeholders involved in reviewing and approving the plan.

**Example 2: Detail Outline of SCMP for development of Building Construction Design Software**

1. **Introduction**
   * **Purpose:** Define the purpose and objectives of the SCMP for the development of the building construction design software.
   * **Scope:** Identify the boundaries within which the SCMP will apply, focusing on the specific software project and configuration items.
   * **Objectives:** Clearly state the key objectives, such as ensuring version control, managing changes effectively, and promoting collaboration among the development team.
   * **Overview of the Software Project:** Provide a brief overview of the building construction design software project, including features and stakeholders.
   * **Document Structure:** Offer a high-level overview of the structure of the SCMP document for ease of navigation.
2. **Configuration Management Process**
   * **Description of Processes:** Detail the configuration management processes to be followed, including version control, change management, and release management.
   * **Version Control Procedures:** Explain the procedures to maintain version control for all software components.
   * **Change Management Procedures:** Provide guidelines for requesting, assessing, and implementing changes to the software.
   * **Release Management Procedures:** Define how release packages will be created, tested, and deployed.
3. **Configuration Items**
   * **Identification of CIs:** Identify and categorize the configuration items, such as software components, libraries, databases, and documentation, specific to building construction design software.
   * **Categorization of CIs:** Organize the configuration items to maintain uniformity and stability.
   * **Documentation Management:** Specify how documentation and artifacts will be managed as configuration items.
4. **Version Control**
   * **Version Control Tools:** Identify the tools and systems used for version control, such as Git, Subversion, or other versioning systems.
   * **Branching and Merging Strategies:** Define the branching and merging strategies to be followed for parallel development and release maintenance.
   * **Versioning Scheme:** Explain the versioning scheme used for the building construction design software.
5. **Change Management**
   * **Change Control Board:** Describe the composition and responsibilities of the Change Control Board.
   * **Change Request Procedures:** Outline the process for submitting, evaluating, and approving change requests, including impact analysis.
   * **Impact Analysis:** Detail the process for analyzing and communicating the impact of changes to the software.
6. **Release Management**
   * **Release Packaging Procedures:** Document how releases will be packaged, tested, and deployed, ensuring minimal disruption to ongoing projects.
   * **Release Notes:** Explain how release notes will be created to communicate changes in each release.
   * **Dependency Management:** Address how dependencies between components will be managed in releases, ensuring compatibility and interoperability.
7. **Configuration Audits**
   * **Audit Processes:** Specify the configuration audit processes to ensure the integrity and consistency of the building construction design software.
   * **Types of Audits:** Define different types of audits, including functional, physical, and documentation audits.
   * **Audit Reporting:** Outline how audit findings will be reported and corrective actions implemented.
8. **Tools and Infrastructure**
   * **Configuration Management Tools:** List the tools and infrastructure used for configuration management, software development, automated builds, and deployment tools.
   * **Development Environments:** Describe the development environments and their configurations, including local development and testing environments.
   * **Automation Tools:** Identify any automated tools used for building, testing, and deployment processes, such as continuous integration and delivery (CI/CD) platforms.
9. **Roles and Responsibilities**
   * **Configuration Managers:** Define the responsibilities of configuration managers in managing the software configuration processes.
   * **Developers:** Clarify the roles and responsibilities of developers regarding configuration management practices and adherence to version control policies.
   * **Quality Assurance Personnel:** Detail the responsibilities of QA personnel in ensuring the quality and consistency of the software components.
10. **Training and Documentation**
    * **Training Requirements:** Specify the training needed for team members on configuration management processes, version control, and usage of configuration management tools.
    * **Documentation Guidelines:** Provide guidelines for documenting configuration changes, handling version release notes, and maintaining system documentation.
    * **Plan Maintenance:** Detail the procedures for maintaining and updating the SCMP as needed to reflect changes in the software project and development requirements.
11. **Approval and Review**
    * **Approval Process:** Define the process for seeking approval from stakeholders for the SCMP, including designated approvals and reviews.
    * **Review Frequency:** Specify how often the SCMP will be reviewed and updated to ensure ongoing relevance and compliance.
    * **Stakeholders Involved:** Identify the stakeholders involved in reviewing and approving the plan, including project managers, developers, testers, and end users.

**2.2 Conclusion**

In conclusion, the role of Software Configuration Management (SCM) in software development projects cannot be overstated. From planning and organizing to controlling and coordinating software changes throughout the Software Development Life Cycle (SDLC), SCM is essential for maintaining software integrity and ensuring project success. By emphasizing the importance of SCM planning and proactive monitoring, organizations can mitigate risks, reduce maintenance costs, improve collaboration in geographically distributed teams, and enhance the efficiency and quality of their software products.

The development of a comprehensive Software Configuration Management plan, as outlined in this chapter, provides a systematic approach to managing changes, version control, and the integrity of software products. By addressing key elements such as roles and responsibilities, tools and infrastructure, version control, change control process, build and release management, quality assurance integration, configuration status accounting, training, communication, and continuous improvement, organizations can create a solid foundation for effective SCM practices.

Moreover, the implementation of surveillance and monitoring practices in software configuration management activities is crucial for ensuring efficient and secure software development processes. By utilizing technologies like artificial neural networks, monitoring agents, and comprehensive auditing methods, organizations can proactively track changes, manage access controls, and generate reports to maintain the integrity and security of their software configurations.

Overall, by mastering the learning outcomes provided in this chapter and adhering to the principles of effective SCM planning, participants will be well-equipped to design and implement robust Software Configuration Management strategies that enhance the efficiency, quality, and success of software development projects. Continuous improvement, adaptability to change, and a proactive approach to monitoring will be key elements in ensuring that software projects meet their objectives and deliver value to stakeholders.

## **2.3 Summary**

Software Configuration Management (SCM) serves as a fundamental aspect of software development, guiding the planning, organization, controlling, and coordination of software changes throughout the Software Development Life Cycle (SDLC). Efficient SCM practices are crucial for ensuring software integrity, reducing maintenance costs, and enhancing collaboration among geographically distributed teams. Embracing SCM planning, modular design, and code refactoring is key to addressing common challenges like late product delivery. An Automated Configuration Management System (CMS) can aid in infrastructure management by automating configuration tasks.

The process of developing a Software Configuration Management plan involves a detailed approach encompassing various essential elements. This includes setting clear objectives, defining roles and responsibilities, selecting appropriate tools, implementing version control practices, managing configuration identification, overseeing change control processes, executing build and release management strategies, integrating quality assurance measures, maintaining configuration status accounts, providing training and communication, and emphasizing continuous improvement.

## In summary, mastering the learning outcomes of SCM enables participants to design and implement robust Software Configuration Management strategies that enhance efficiency, quality, and success in software development projects. Combining meticulous planning, structured monitoring, and proactive adjustments ensures that organizations can navigate challenges effectively, deliver high-quality software products, and drive continuous improvement for future endeavors.

## **Self-Assessment Questions (SAQs)**

**SAQ1:** What is the purpose of a Software Configuration Management Plan (SCMP)?

**SAQ 2:**  What are the key components included in a Software Configuration Management Plan (SCMP)?

**SAQ 3:** Why is scope definition important in a Software Configuration Management Plan (SCMP)?

**SAQ 4:** How does Software Configuration Management Plan (SCMP) enhance software development processes?

**SAQ 5:** What is the role of risk management in a Software Configuration Management Plan (SCMP)?

**SAQ 6**: What are the key steps involved in developing a Software Configuration Management Plan (SCMP)?

**SAQ 7**: Clearly elucidate how roles and responsibilities could be determined in a Software Configuration Management Plan (SCMP).

**SQA 8**: Explain how effectiveness of a Software Configuration Management Plan (SCMP) can be evaluated and improved.

**SAQ 9:** Outline the components of Software Configuration Management Plan (SCMP) for an eLearning platform.

**Self Assessment Answers (SAA)**

**SAA 1:** The purpose of a Software Configuration Management Plan (SCMP) is to establish a structured approach to managing software configurations, ensuring consistency, traceability, and control over software artifacts throughout the development life cycle.

**SAA 2:** The key components of a Software Configuration Management Plan (SCMP) typically include scope definition, objectives, process definition, tools and infrastructure, roles and responsibilities, training and documentation, risk management, and approval and review process.

**SAA 3:** Scope definition is important in an SCMP as it helps in clearly defining the boundaries of configuration management activities, including the software project, configuration items, and the phases of the software development life cycle that will be covered.

**SAA 4:** An SCMP enhances software development processes by ensuring consistency and integrity of software configurations, facilitating collaboration and communication among team members, enhancing traceability and auditability of software artifacts, and enabling efficient change management and release processes.

**SAA 5:** Risk management in an SCMP involves identifying potential risks and challenges related to configuration management and developing strategies to mitigate these risks effectively, ensuring that unexpected issues are addressed proactively.

**SAA 6**: The key steps in developing an SCMP typically include defining the scope of configuration management activities, outlining objectives, identifying processes, selecting tools and infrastructure, assigning roles and responsibilities, providing training and documentation, conducting risk management, and establishing an approval and review process.

**SAA 7**: Roles and responsibilities can be determined by identifying the team members involved in configuration management, specifying their duties and responsibilities, establishing communication channels, and ensuring clear accountability throughout the development process.

**SAA 8**: The effectiveness of an SCMP can be evaluated through regular reviews, assessments of compliance with the plan, feedback from team members, monitoring of configuration management activities, and identification of areas for improvement to enhance the plan's efficiency and effectiveness.

**SAA 9:** The key components of the Software Configuration Management Plan (SCMP) for an eLearning platform may include:

1. **Scope Definition:** The SCMP will define the scope of configuration management activities for the eLearning platform, covering aspects such as the learning content, user interfaces, databases, and any associated software applications.
2. **Version Control Procedures:** It will detail the procedures and tools to be used for maintaining version control of eLearning content and software components, ensuring that updates and changes are tracked and managed effectively.
3. **Collaboration Tools and Processes:** The plan will outline the collaboration tools and processes that will enable teams to work together on content creation, updates, and maintenance, ensuring smooth concurrent editing and seamless integration of changes.
4. **System Configuration Items:** It will identify specific configuration items such as learning modules, assessment tools, multimedia content, and user management systems, and define how these items will be controlled and managed.
5. **Testing and Release Management:** Procedures for testing eLearning content, managing release cycles, and ensuring that released versions align with controlled configurations will be included as part of the plan.
6. **Change Management and Approval Procedures:** The plan will define the process for managing change requests, assessing their impact, obtaining approvals, and implementing changes while maintaining the integrity of the eLearning platform.

References/ Further reading

# References

AcqNotes. (2024). *Configuration Management Plan*. Retrieved from AcqNotes: https://acqnotes.com/acqnote/careerfields/configuration-management-plan#google\_vignette

Alexis Leon. (2015 ). *Software Configuration Management Handbook.* Boston : Artech House .

Arie Karniel, ‎Yoram Reich ·. (2014). *Managing the Dynamics of New Product Development Processes.* London: Springer.

CMstat. (2024). *What is a Configuration Management Plan?* Retrieved from https://cmstat.com/cmsights-news-posts/what-is-a-configuration-management-plan

Cyrille Michaud. (2011). *Software-Configuration-Management-Plan-Template*. Retrieved from https://blog.cm-dm.com/public/Templates/Software-Configuration-Management-Plan-Template.doc

Matthew Martin. (2023). *Software Configuration Management in Software Engineering*. Retrieved from guru99: https://www.guru99.com/software-configuration-management-tutorial.html