1. Identification

1.1 Document Overview

This report is a Software Configuration Management plan of Forumium, which is a web based forum administration that was created by Flarem Plus Software. This document is aimed at defining a formal and methodical process of locating, managing, monitoring, and auditing all software configuration objects within the software development life cycle.

Software Configuration Management is an assurance that the Forumium system is consistent, traceable, and stable as it is developed, undergoes testing, deployment, and maintenance. The SCM operations outlined in this document involve configuration identification, configuration control, version management, configuration status accounting, and configuration auditing.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

Abbreviation Definition

SCM – Software Configuration Management
SCI – Software Configuration Item
ECP – Engineering Change Proposal
CCA – Configuration Control Authority
PDR – Preliminary Design Review
CDR – Critical Design Review
TRR – Test Readiness Review

1.2.2 Glossary

Baseline - A formally reviewed and approved version of a configuration item or group of items that serves as a reference for further development and can only be modified through formal change control.

SCI - A software related artifact such as source code, configuration files, or documentation that is managed under configuration control.

Identification - The process of uniquely naming and organizing configuration items within the SCM repository.

Change Control - A controlled procedure that ensures modifications to configuration items are evaluated, approved, and documented before implementation.

Version - A uniquely identified state of a configuration item that reflects functional or structural changes.

Revision - A modification applied to a version to correct defects without altering documented functionality.

Release - The formal distribution of an approved software version.

Configuration Audit - A systematic review to verify that configuration items comply with approved baselines and requirements.

2. Organization

The members of the Flarem Plus Software development team undertake Software Configuration Management of the Forumium project. There are responsibilities that are well delegated to achieve proper coordination, accountability and control of the project life cycle. Main positions undertaken by the SCM activities are Project Manager, Software configuration Manager, Software Quality assurance Manager, Software engineers, Software designers and client stakeholders.

A centralized SCM repository is used to manage configuration items, version history, and approved baselines. The system is also executed and tested with external development and run time environments. The official configuration items are only seen as those that are stored in the SCM repository.

2.1. Decisions Process and Responsibilities

Role	Activities and Responsibilities	Decision Responsibilities
Project Manager	Coordinates project activities and approves baselines and releases.	Final authority on change approvals and conflict resolution.
Software Configuration Manager	Manages GitHub repository, version control, and configuration items.	Evaluates change impact and enforces configuration control decisions.
Software Quality Assurance Manager	Conducts audits and ensures SCM compliance and quality.	Approves quality compliance or requests rework.
Software Designer	Prepares and maintains system design documents.	Confirms design impact of proposed changes.
Programmer	Implements approved changes and supports testing.	Advises on technical feasibility of changes.
Documentation Manager	Maintains and versions project documents.	Confirms documentation readiness for approval.

3. Configuration Identification

Configuration identification forms the foundation of the SCM process for the Forumium project. All software configuration items are uniquely identified, named, and organized using an object oriented approach within the GitHub repository.

Configuration items include source code modules, configuration files, documentation, database schemas, and deployment scripts. Each item is traceable through its identification scheme, version history, and associated change records.

3.1 Identification Rules

3.1.1 Identification Rules of Configuration Items

3.1.1.1 Identification of a Configuration Item

Each configuration item is identified using a standardized naming convention to ensure uniqueness and traceability. The general identification format used is:

“XXX_Vm.n”

Where XXX represents the configuration item name and Vm.n represents the version number.

3.1.1.2 Version Number of a Configuration Item

A version number is mandatory before any delivery or release. The Forumium project adopts a structured versioning scheme:

- Major
- Minor
- Revision

Major versions indicate significant functional changes, minor versions indicate incremental improvements, and revision numbers represent bug fixes or maintenance updates.

Official versions are marked using GitHub tags after approval and testing.

3.1.2 Identification Rules of Documents

3.1.2.1 Description of Documents Identifiers

Documents are identified using the following format:

XXX_<document type><*document number*><revision index>

This ensures consistent identification and controlled document evolution.

3.1.2.2 Definition and Evolution of the Revision Index

The revision index is updated prior to document distribution. Major revisions reflect substantial changes, minor revisions reflect incremental updates, and revision numbers reflect corrections or refinements.

3.1.3 Identification Rules of a Media

3.1.3.1 Internal Identification

Media used for delivery or backup purposes are identified using the following format:

<configuration item identification>/<media>/<volume>

This allows traceability when multiple media units are involved.

3.2 Reference Configuration Identification

Reference configurations are defined at key project milestones. Each reference configuration is identified by its contents, approval status, and associated review activities. These reference configurations provide stable checkpoints during development and testing.

3.3 Configuration Baseline Management

The projects of the Forumium rely on configuration baselines as a way of establishing reference points that are stable and approved throughout the software development lifecycle. After having a baseline in place, then any modifications to its contents are also regulated using the formal configuration control process.

Three baselines are defined. The functional baseline is the one that records the approved system requirements and anticipated functionalities. The design baseline comprises of accepted system designs and architecture artifacts. Product baseline includes completed source code, configuration files and documentation that is all set to be deployed.

GitHub is used as all baselines are version controlled. The Project Manager allows baseline approval, whereas the Software Configuration Manager manages baseline creation and maintenance. This will provide consistency, traceability and controlled evolution of the system across the project.

4. Configuration Control

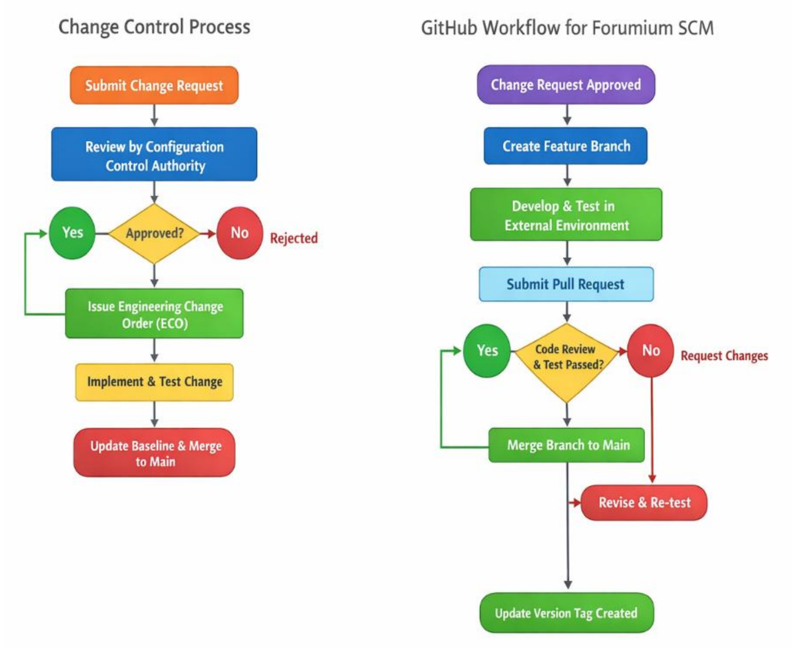
Configuration control is used to make sure that all modifications in the system of the Forumium are brought in a methodical and controlled way. Engineering Change Proposals are the documents which initiate changes, their cause, configuration items which are impacted and the effect on costs and schedule. Configuration Control Authority considers every proposal and makes a decision to approve or reject the change.

Configuration control is applied through github branching and pull request mechanisms. Modifications on the primary branch are prohibited.

4.1 Change Control Process

The Project Manager or stakeholders initiate change requests. Once approved, a dedicated GitHub branch is created to implement the change. After development and testing in external execution environments, the change is reviewed and merged into the main branch through a pull request.

In case of a rejected change, the branch is closed and the integration is not done. Approved and merged changes are the only modifications that are considered as a part of the official configuration.


Change Control Process
Submit Change Request
Review by Configuration Control Authority
Yes
Approved?
No
Rejected
Issue Engineering Change Order (ECO)
Implement & Test Change
Update Baseline & Merge to Main
GitHub Workflow for Forumium SCM
Change Request Approved
Create Feature Branch
Develop & Test in External Environment
Submit Pull Request
Yes
Code Review & Test Passed?
No
Request Changes
Merge Branch to Main
Revise & Re-test
Update Version Tag Created

5. Configuration Support Activities

5.1 Configuration Status Accounting

Configuration Status Accounting documents and reports configuration item status during the software lifecycle. The main configuration status records of the Forumium project are github commit histories, github pull requests and tagged releases.

Execution environments are treated as non authoritative. Only artifacts committed to GitHub are recognized as official configuration items.

5.1.1 Evolutions Traceability

Every change is tracked with the help of GitHub commit identifiers. Commit messages explain the nature of changes, modules that are being changed and approvals. Document changes are tracked through revision histories.

5.1.2 Setting up Configuration Status

Each configuration item is labeled with its version number, creation date, and associated Version Delivery Description prepared by the Software Configuration Manager.

5.1.3 Configuration Status Diffusion

Configuration status information is distributed by the Software Configuration Manager and Software Quality Assurance Manager to relevant stakeholders as required.

5.1.4 Configuration Status Records Storage

All configuration records are stored within the GitHub repository, including change requests, documentation, version delivery descriptions, and historical configuration states. GitHub serves as the single source of truth for configuration records.

5.2 Configuration Audits

Configuration audits are conducted to verify that the implemented system conforms to approved baselines and requirements. Baseline audits and functional configuration audits are performed to ensure traceability, consistency, and compliance.

5.3 Reviews

Formal reviews conducted during the project include Preliminary Design Review, Critical Design Review, and Test Readiness Review. These reviews assess design completeness, implementation readiness, and testing compliance.

5.4 Configuration Management Plan Maintenance

The Software Configuration Manager maintains this software configuration Management plan. Any updates of this plan use the same configuration control and version management processes detailed in this document. Accepted changes are logged into the GitHub repository so as to maintain coherence between paperwork and system execution.