



WANDERERS

SOFTWARE CONFIGURATION MANAGEMENT PLAN

Version 1.4

03/25/2025

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 2 / 11

TABLE OF CONTENTS

WANDERERS	1
Software Configuration Management Plan	1
1 Identification	3
1.1 Document overview	3
1.2 Abbreviations and Glossary	3
1.2.1 Abbreviations	3
1.2.2 Glossary	3
1.3 References	3
1.3.1 Project References	3
1.3.2 Standard and regulatory References	3
1.4 Conventions	3
2 Organization	4
2.1 Activities and responsibilities	4
2.1.1 Decisions process and responsibilities	5
3 Configuration identification	5
3.1 Identification rules	5
3.1.1 Identification rules of configuration items	5
3.1.2 Identification rules of SOUPs	6
3.1.3 Identification rules of documents	7
3.1.4 Identification rules of a media	7
3.2 Reference configuration identification	7
3.3 Configuration Baseline Management	7
4 Configuration control	8
4.1 Change Management	8
4.2 Evolutions control of SOUP items	8
5 Configuration support activities	9
5.1 Configuration Status Accounting	9
5.1.1 Evolutions traceability	9
5.1.2 Setting up Configuration status	9
5.1.3 Configuration status diffusion	10
5.1.4 Configuration status records storage	10
5.2 Configuration audits	10
5.3 Reviews	11
5.4 Configuration management plan maintenance	11

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 3 / 11

1 Identification

1.1 Document overview

This document contains the software configuration management plan of software Wanderers.

1.2 Abbreviations and Glossary

1.2.1 Abbreviations

- SCM: Software Configuration Manager
- QA: Quality Assurance
- SRS: Software Requirement Specification
- VDD: Version Description Document
- CM: Configuration Management
- CI: Configuration Item
- SOUP: Software of Unknown Provenance
- CMP: Configuration Management Plan

1.2.2 Glossary

- Branch: a line of development that exists independently of another line, yet still shares a common history, and can be merged in the future. Often used as in concept of feature branching, where each independent branch works on a feature, before merging to the main branch after completion.
- Version: state of a configuration item at a well-defined point in time
- Variant: versions that coexist

1.3 References

1.3.1 Project References

#	Document Identifier	Document Title
R1	SRS	Software Requirements Specification

1.3.2 Standard and regulatory References

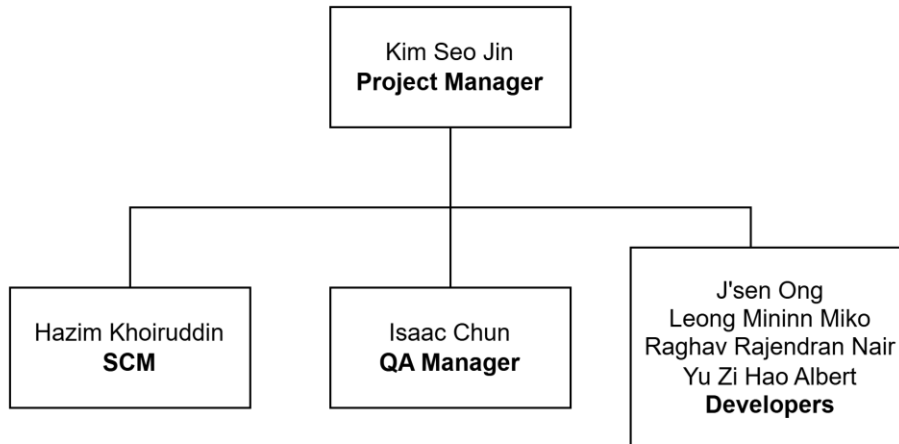
#	Document Identifier	Document Title
STD1	SCM1	IEEE 828-2012- IEEE Standard for Configuration Management in Systems and Software Engineering
STD2	SSCM1	IEEE 828-2012- IEEE Standard for Configuration Management in Systems and Software Engineering

1.4 Conventions

Typeface	Usage	Example
Bold	Emphasis, headers, titles	Software Configuration Management
Italics	Minor emphasis, filenames, figures, tables	<i>Figure1:</i>

2 Organization

The software configuration is managed by members of the project, with specific tools. Responsibilities are shared between:



Note: While **Raghav** is part of the Release and QA Team, he is considered a developer as he is building release related infrastructure.

2.1 Activities and responsibilities

Activities when setting up the project	Person responsible
Identify the configuration items	SCM
Install the bug repository tool and set up the database	SCM
Install the software configuration repository tool and set up the database	SCM
Manage and structure the reference space	SCM
Define the configuration processes	SCM

Activities during the project lifecycle	Person responsible
Export components for modification, test or delivery	SCM
Set under control validated components	SCM
Create version, write version delivery document	SCM
Approve reference configurations	Project Manager
Verify version to be delivered and authorise deliveries	Project Manager
Backup spaces	SCM
Do configuration audits	QA Manager
Inspect configuration records	QA Manager
Archive reference version	SCM

Management activities	Person responsible
Manage versions and archives	SCM
Manage configuration records	SCM
Produce reports and statistics	SCM
Manage reference space and its access control list	SCM
Manage spaces backup and archive media	SCM
Manage quality reports	QA Manager

2.1.1 Decisions process and responsibilities

Responsibilities during reviews, audits and approvals are listed below:

At the end of an activity of the project

Activities	Person Responsible
Do a configuration freeze	SCM
Present a configuration state of the components impacted by the activity	SCM
Present a documentation state of the components impacted by the activity	SCM

During a configuration management process audit:

Activities	Person Responsible
Do the configuration management process audit	Project Manager
Present the records of the configuration management process	SCM
Present the quality records of the configuration management process	Quality Manager
Present the records of the documentation management process	SCM

3 Configuration identification

Configuration identification (CI) involves identifying the configuration of items such as hardware, software and documentation within a system through their physical, functional and performance characteristics. This process includes:

1. Establishing baselines for each CI by documenting its **configuration and version** as it is a reference point for future changes.
2. Assigning and documenting of unique identifiers, labels, versions, descriptions, or sometimes hashes to each CI
3. Allowing traceability of changes and update to the system through keeping records of any modifications made to configuration
4. Defining relationships between different CI

3.1 Identification rules

3.1.1 Identification rules of configuration items

3.1.1.1 Identification of a configuration item

The identification of configuration item is:

- XXX_Vm.n
 - where: "Vm.n a" is the version of the configuration item.
- Code
 - Source Code
- Documentation
 - SRS
 - Quality Plan
 - Project Plan
 - Risk Management Plan
 - Test Plan

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 6 / 11

- System Design
 - Use case model
 - Use case descriptions
 - Software architecture
 - Unit Test
 - Coverage Test
- Specification
 - Project requirements specification
 - Unit Test

3.1.1.2 Version number of a configuration item

The attribution of a version number is a prerequisite to any delivery of any configuration item. This number shall be incremented before a new delivery, if the product or its documentation were modified.

The definition rules of a version number are the following:

- We will **follow semantic versioning to prevent “dependency hell”** and make releasing of new package versions more manageable as our system grows.
- Version number follows the structure of MAJOR.MINOR.PATCH. Each time there is a relevant change, the version needs to be increased.
- Source Code
 - MAJOR releases:
 - Initial release of website
 - Breaking changes (removal of major features, complete redesigns, major changes to APIs)
 - MINOR releases:
 - New features or enhancement that do not break existing functionality
 - PATCH releases:
 - Fixes for security vulnerabilities, performance optimizations, or small UI changes
- Other configuration items
 - MAJOR:
 - Initial release of configuration item
 - Major restructuring, rewrite or significant changes to methodology
 - MINOR:
 - New sections or additional details that does not change the core structure
 - PATCH
 - Grammatical issues or small changes to diagrams

3.1.2 Identification rules of SOUPs

3.1.2.1 Identification of a SOUP

Take the ID of SOUP manufacturer for clearer identification of SOUP by project members.

3.1.2.2 Version number of a SOUP

Take the version number of SOUP manufacturer, similar to SOUP identification for clearer identification by project members.

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 7 / 11

3.1.3 Identification rules of documents

3.1.3.1 Description of documents identifiers

The identification of documents is described below:

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

where:

- " document type " is:
 - Foo for FOO documents,
 - BAR for bar documents
- " document number " is a incremental number, with a separate list for each document type,
- " revision index " designates the approved iteration of the document. The revision index is V1 for the first iteration, V2 for the second and so on.

3.1.3.2 Definition and evolution of the revision index

The attribution of a revision index is a prerequisite to any delivery of a document or file. This index shall be incremented before the diffusion of a modified document. The definition rules of a revision index are the same as described in "3.1.1.2 Version number of a configuration item".

3.1.4 Identification rules of a media

3.1.4.1 Internal identification

The identification of a media is described below:

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

where:

- "media" is the media number,
- "volume" is an incremental number to distinguish the media if the delivery contains more than one media.

3.2 Reference configuration identification

Each reference configuration is defined by:

- An identifier,
- Its content listed in the corresponding Version Delivery Description document,
- The acceptation or validation reviews associated to the building of the reference configuration.

A reference configuration is established for each design review and each test review of the project.

3.3 Configuration Baseline Management

A configuration baseline is a reference point that defines a stable version of system components. As such, configuration baseline management describes how these baselines are established and controlled over the course of a project lifecycle.

The following baselines will be established for proper configuration management:

- **Functional Baseline (FBL)**
 - Defines the system's functional requirements and expected behavior at a high level, which includes functionality and performance to ensure alignment with user needs and expectations
- **Allocated Baseline (ABL)**
 - Describes the design and structure of the system, mapping functional requirements to technical components. It is also a schema/blueprint for system development to ensure that functional requirements are addressed through the system's careful design.

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 8 / 11

- Product Baseline (PBL)
 - Consists of final and fully implemented system, including verified and validated components such that the system meets the specified functional and nonfunctional requirements. This is also the baseline where there is transition from just local or internal deployment to operational deployment.

4 Configuration control

Configuration control ensures that all changes to system components, baselines and configurations are properly documented.

4.1 Change Management

Problem resolution:

- Changes requests are emitted from by the project manager according to the problem resolution process, which outlines the identification and resolution of issues affecting the current system/project
- When a change request is accepted by the project manager/product manager, a branch is created in the source code management system (GitHub) to implement the necessary changes. A ticket is also created on JIRA for tracking of branches, pull requests and deployments.
- The branch is assigned a unique identifier based on (i.e. change request number, ticket number or descriptive tag) for tracking and traceability purposes.
- Branch content is updated with the necessary code or configuration modifications as defined in the change request, and the update is tested and verified by the QA team before final approval of the change request and outcomes.

Multiple configuration:

- Changes requests of configuration files are emitted by the product manager according to the production procedure, which ensures that correct versioning and consistency of configuration files is used in the correct environment.
- When a change request is accepted by the project manager/product manager, a branch is created in the source code management system (GitHub) to implement the necessary changes. A ticket is also created on JIRA for tracking of branches, pull requests and deployments.
- The branch is assigned a unique identifier based on (i.e. change request number, ticket number or descriptive tag) for tracking and traceability purposes.
- Branch content is updated with the necessary code or configuration modifications as defined in the change request, and the update is tested and verified by the QA team before final approval of the change request and outcomes. They are then merged back into the main branch or released into production through automatic deployments or our internal deployment process.

4.2 Evolutions control of SOUP items

1. Initial Assessment of SOUP Items
 - a. At the beginning of the project, each SOUP items are identified and assessed to ascertain its necessity, dependability as well as support given by third party providers.
 - b. Establish a baseline version for each SOUP items
 - c. Stable and well-supported versions that fit our use cases and product are selected. As much as possible, we choose SOUPs that do not add redundant code or features to the product, keeping our product lean.
2. Freeze SOUP Versions

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 9 / 11

- a. For critical SOUP components that have a direct impact on system stability, freezing the version may be considered for the duration of the project, depending on the evolution of the project
3. Periodic Reviews of SOUP items
 - a. At regular intervals, SOUP items are reviewed to assess if updates or patches are needed. The review helps to identify:
 - i. New releases or updates to SOUP items
 - ii. Security or bug fixes
 - iii. Compatibility issues with other components
 - iv. Any critical vulnerabilities that are discovered after SOUP has been locked
4. Update Process
 - a. When an update is necessary, it must be managed carefully to avoid any breaking changes
 - b. Tests are done to ensure that system works as expected after an update
 - i. Unit tests to ensure updated components work as expected
 - ii. Integration tests to ensure that the new version works well with the rest of the system
 - iii. Regression tests to ensure that no existing functionality breaks
 - iv. All tests are to be done in a feature branch
 - c. The configuration management team must approve changes to SOUP items before merging the changes into the staging branch
5. Documentation of Changes
 - a. Any updates to SOUP items should be well-documented, including:
 - i. Exact version and patch level of updated SOUP
 - ii. Issues or considerations identified from the update
 - iii. Process for reverting changes in case of unforeseen issues

5 Configuration support activities

5.1 Configuration Status Accounting

Configuration Status Accounting (CSA) is the process to record, store, maintain and report the status of configuration items during the software lifecycle. All software and related documentation should be tracked throughout the software life.

5.1.1 Evolutions traceability

The traceability of modifications of items given their types:

- Document: The modification sheet number identifies the origin of the modification. The modified paragraphs in the document are identified, if possible, by revision marks.
- Source file: The software configuration management tool records, for each source file or group of source files, a comment where is described the modification.
- Configuration item: The Version Delivery Description of the article identifies the modification sheet included in the current version.

The modification sheet describes the modifications done to the components with enough precision to identify the modified parts.

5.1.2 Setting up Configuration status

The SCM sets up the state of all versions and of each configuration article with:

- The label
- The version number
- The creation date of the VDD,

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 10 / 11

5.1.3 Configuration status diffusion

The SCM and the QA Manager writes the VDD.

5.1.4 Configuration status records storage

The records are stored in a configuration folder, which contains:

- The requests sorted by record number,
- The software documents,
- The VDD's,
- The configuration states sorted chronologically.

5.2 Configuration audits

Configuration audits are crucial for assessing the accuracy and completeness of CIs and ensuring compliance with the defined configuration management procedures. The audits defined below are carried out to verify that CIs have met established standards, aligned with baselines and satisfied functional and quality requirements.

The Wanderers software will conduct 3 different types of configuration audits:

1. Baseline Audits (BA)

- a. The BA verifies the correctness and completeness of the configuration baseline. This audit takes place after the baseline is established and before the Wanderers software is released or deployed. When performing the audit, the team examines the CIs defined as part of the baseline and ensures that they have been correctly identified and documented. The baseline documentation, which includes design specifications, source code, and other relevant artifacts, is carefully reviewed to confirm that all configuration items are properly included and referenced. If items are found to be missing or incorrectly identified, they are flagged for correction, ensuring that the baseline is a reliable reference for future development.

2. Functional Configuration Audit (FCA)

- a. The FCA focuses on confirming that the CIs perform according to the functional requirements defined at the start of the project. This audit is concerned with verifying that the system behaves as expected, ensuring that all CIs meet the operational goals established in the planning phase. During this audit, the QA team reviews the design specifications and functional requirements, as well as the testing, such as results from unit tests, integration tests and regression tests. The FCA assesses whether the software or system functions correctly and consistently. If any features or functions fail to meet the established requirements, corrective actions are recommended to resolve these issues.

3. Software Configuration Audits (SCA)

- a. The SCA targets the specific configuration of software components within the system, ensuring that they are developed, tested, and maintained in compliance with the CM plan. This audit verifies that all software artifacts, including code, libraries, and configuration files, are appropriately versioned, tracked, and controlled. It checks that changes to software components are documented, authorized and traceable through a formal change control process. The release team will review version control logs, change management documentation and release records to ensure that every modification to the software has been correctly managed. Additionally, the SCA checks the consistency between the software's development and the established configuration management practices. Any untracked changes or deviations from the approved processes are highlighted

Software Configuration Management Plan of Wanderers		
Doc #1	Version: 1.4	Page 11 / 11

for correction to ensure that the software configuration remains in compliance with the CM plan.

5.3 Reviews

In order to guarantee effectiveness of our configuration management strategy, we have to create and implement a Configuration Management Plan Review. This review is an independent evaluation of our existing configuration management plan, to ensure that it covers the following elements:

1. Configuration Identification
2. Configuration Audit
3. Configuration Planning
4. Configuration Status Accounting
5. Verification

Additionally, technical reviews also serve as checkpoints throughout the project where the team assesses whether the project is on track in terms of deliverables. These reviews are closely tied to configuration management, in terms of establishing baselines, creating branches and ensuring everything aligns with the project goals. During technical reviews, the team evaluates whether the CIs are ready to be baselined. If so, they are added to the baseline.

The SCM ensures that all items are correctly versioned and documented before being approved. If any issue arises, the SCM ensures that these changes are tracked and incorporated into the process for future work. Throughout the project, the SCM ensures that all CIs are compliant with regulatory requirements and specified quality management standards (STD1 and STD2 respectively).

Finally, the team in charge of reviewing the CM plan must ensure that it satisfies the requirements for quality management as well as regular compliance standards. The CM Plan must be feasible, such that it is able to be successfully implemented as part of a process in the project, rather than a document that looks good on paper but is unable to be brought to realization. Through this conclusive review process, this ensures that our CM Plan aligns with the project's objectives and is feasible, allowing us to deliver a product that not only is well managed, but satisfies stakeholder and project standards.

5.4 Configuration management plan maintenance

Throughout the project's lifecycle, the CM Plan will outline the tasks, roles, and responsibilities necessary for ongoing configuration management planning. This ensures that all configuration management activities are continuously aligned with the project's evolving needs.

The maintenance of the CM Plan will follow a clear schedule. The SCM will oversee the plan's monitoring and ensure that it stays relevant. Each week, the plan will be reviewed and updated as needed. Any changes to the plan will require approval from both the project manager and the SCM before they are implemented. Once approved, the project team will be notified via email to ensure everyone is on the same page.

At the start of each software project phase, the CM plan will be reviewed, adjusted as necessary, and re-approved before being distributed to the project team. This ensures that the plan remains relevant throughout the project's progression. Detailed procedures related to CM plan maintenance, including those documented in appendices or other references, will be followed. All maintenance tasks will comply with the Software Configuration Management Plans (SSCM1) IEEE 828 standards to ensure consistency and traceability across the entire project lifecycle.