SportZ

SOFTWARE CONFIGURATION MANAGEMENT PLAN

Lin Zixing
Fabian Wong
Chew Poshi
Lim Sheng Zhe
Chia Songcheng
Chee Zi Hoe
Hermes Lim

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Hermes Lim	17/03/2022	Chee Zi Hoe	18/03/2022	Initial draft
1.1	Hermes Lim	28/03/2022	Chee Zi Hoe	29/03/2022	New software configuration item identified and finalised for first release.

UP Template Version: 11/30/06

TABLE OF CONTENTS

1 Introduction

1.1 Purpose

The purpose of a Software Configuration Management Plan (SCMP) is to define a project's structure and methods for identifying, defining, and baselining configuration items (CI). Controlling modification and releases of CIs are also carried out to maintain system integrity. As changes are inevitable in the software development cycle, whether during development or maintenance phase, reporting and recording the status of CIs and any requested modification is required as well. Furthermore, SCMP ensures the completeness, consistency and correctness of CIs and controls the storage, handling, and delivery of CIs.

The SCMP determines the guidelines that will be used to maintain consistency and control over the technical production of our project - SportZ. Assuring that the correct products are delivered, the requirements are met, and the software and supporting materials are consistent.

This SCMP thereby provides a process and means to control and manage software upgrades to SportZ by establishing uniform configuration management practices for managing systems, software, hardware, and documentation changes throughout the lifecycle.

1.2 Document overview

This document contains the SCMP of SportZ.

Configuration Management (CM) is the process of identifying, organising, controlling system configuration and change to maintain system integrity.

There are three parts of configuration management which includes the following:

- Change control of configuration items:
    - The management of project's deliverables and related documentation, throughout the lifecycle of the project.
- Version control of configuration items:
    - A process of tracking and managing the versions generated during system implementation.
- Product building of configuration items
    - The transformation of software and/or documentation source into a deliverable product as one part of the configuration management process. Tools are used to manage and control an accurate, repeatable, and documentable building process for software and documentation.

1.3 Abbreviations and Glossary

1.3.1 Abbreviations

- SCMP: Software Configuration Management Plan
- CM: Configuration Management
- SCM: Software Configuration Manager
- PM: Project Manager
- QM: Quality Manager
- VDD: Version Delivery Description
- FBL: Functional Baseline
- ABL: Allocated Baseline
- PBL: Proposed Baseline
- API: Application Programming Interface
- CSA: Configuration Status Accounting
- CI: Configuration Item

1.3.2 Glossary

- Branch: The practice of creating copies of programs or objects in development to work in parallel versions, retaining the original and working on the branch or making different changes to each.
- Version: The process of numbering different releases of a particular program for both internal use and release designation.
- Variant: Instances of different configuration items that are similar but different.

1.4 References

1.4.1 Project References

S/N	Document Identifier	Document Title
1	1.0	SportZ_Project_Plan_V1.0
2	1.0	SportZ_Change_Management_Plan_V1.0
3	1.2	SportZ_Risk_Management_Plan_V1.2

2 Organisation

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

- The software configuration manager (SCM) - Hermes Lim
- The project manager (PM) - Lin Zixing
- The quality manager (QM) - Chee Zi Hoe

2.1 Activities and responsibilities

The functions required to manage the configuration of the software and its associated persons in charge are listed below:

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	PM
Verify version to be delivered and authorise deliveries	PM
Backup spaces	SCM
Do configuration audits	QM
Inspect configuration records	QM
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	QM

2.1.1 Decision’s 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	PM
Present the records of the configuration management process	SCM
Present the quality records of the configuration management process	QM
Present the records of the documentation management process	SCM

3 Configuration identification

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 as follows:

XXX_x.y.z where **XXX** is the name of the configuration item where **x.y.z** represents the version number and is further elaborated in 3.1.1.2.

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:

- Version numbers are based on semantic versioning in the form of **x.y.z**,
- x = major version. A major release consists of major new features and/or large architectural changes. Some existing features may be deprecated.
- y = minor version. A minor release includes limited architectural changes that enhance sustainability and performance.
- z = patch version. A component fix that may apply to multiple products or multiple releases. Patch fixes are periodically consolidated into maintenance release.

3.1.2 Identification rules of documents

3.1.2.1 Description of documents identifiers

The identification of documents is described below:
XXX_<document type>_<document number>_<revision index>
where:

- "Document type" is the type of document that is listed
- "Document number" is an incremental number, with a separate list for each document type
- "Revision index" designates the approved iteration of the document. The revision index is V1.0 for the first iteration, V1.1 for the second and so on.

3.1.2.2 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 categorised to minor and major mistakes as follows:

- The following are treated as minor revisions:
  - Typing mistakes, spelling mistakes, translation mistakes not affecting a value, a parameter, or a dimension.
  - Addition of information where the added information does not change the meaning of the document.
- The following are treated as major revisions:
  - Comments made during an approval cycle that require major changes of the document (the modified version must go through the approval cycle).
  - Changes to the document where the changes affect values and parameters.

    - Any design change that will require a revision of the Configuration Baseline and associated documents. Potentially impacting the cost, schedule, and performance of the project.

3.2 Reference configuration identification

Each reference configuration is defined by:

- An identifier,
- Its content listed in the corresponding Version Delivery Description document,
- The acceptance or validation reviews associated with 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

The established baselines and their respective definitions are listed below:

- Functional baseline (FBL): The functional baseline is the approved configuration documentation that describes a systems or top-level CI’s performance requirements (functional, interoperability, and interface characteristics) and verification required to demonstrate the achievement of those specified characteristics.
- Allocated baseline (ABL): The allocated baseline is the approved performance-oriented documentation for a CI to be developed that describes the functional performance, and interface characteristics that are allocated from a higher-level requirements document or a CI and the verification required to demonstrate achievement of those specified characteristics.
- Product baseline (PBL): The product baseline is the approved technical documentation that describes the configuration of a CI during the production, deployment, and operational support phases of its life cycle. It consists of completed and accepted system components and documentation that identifies these products.

3.4 Configuration Item

In configuration control, a Configuration item (CI) is any service component, infrastructure element, or other item that needs to be managed in order to ensure the successful delivery of services. Each CI has several characteristics:

- A classification, or type, which indicates what kind of item it is.
- Attributes, which vary by classification and describe the characteristics of the individual CI.
- A status value, which represents the CI’s state in the lifecycle used for CIs of this classification.
- Relationships, which indicate how the CI is related to other CIs.
- An owner, the person who is responsible for the CI.

CIs vary in complexity, size and type. They can range from an entire service, which may consist of hardware, software and documentation, to a single program module or a minor hardware component. The lowest-level CI is usually the smallest unit that will be changed independently of other components.

The following sections 3.4.1 to 3.4.2 contains CIs that are included in this SCMP. SportZ life cycle may add or remove specific items in each of these sections. If any changes are made, this document will be amended to reflect the changes.

3.4.1 Configuration Item Class: Documentation

The Online TMT life cycle includes the development and updating of project documentation.

All project documents must adhere to guidelines in section 3.1. The project documentation includes, but not limited to:

- Project Proposal
- Use Case Model
- System Requirement Specification
- Quality Management Plan
- Project Plan
- Risk Management Plan
- Design Report on Software Maintainability
- Configuration Management Plan
- Change Management Plan
- Release Plan
- CMMI Level 2 Definition
- Test Plan
- Test Cases and Requirements Test Coverage Report

The following information is to be provided for each document deliverable. The document name and publication date will be used in combination to identify unique documents.

- Publication Date
- System Title
- Author(s)
- Concurrence Author(s) Signature

3.4.2 Configuration Item Class: Software / Code

SportZ comprises a number of Application Source Code, Commercial off-the-shelf (COTS) packages and operating systems. These packages include, but may not be limited to:

- Unity3D
- Figma
- Piskelapp and Royalty Free Assets
- Royalty Free Music
- Windows 10 OS
- Github SVN

4 Configuration control

Configuration Control includes the evaluation of all change-requests and change-proposal, and their subsequent approval or disapproval. It covers the process of controlling modifications to the system's design, hardware, firmware, software, and documentation.

SportZ will not have any interfaces with 3rd parties and hence will also not be affected by Software of Unknown Provenance, aka SOUP.

Configuration control has seven main processes:

1. Identifying and documenting change request
2. Analysis and evaluation of change request and production of a change proposal
3. Approval or disapproval of change proposal
4. Unit test and implement the change.
5. Integrate the change
6. Validate the change
7. Close change request

4.1 Software Configuration Change Management

The process in controlling changes made to the baselines and implementation tracking of those changes are listed below.

Problem resolution:

- Changes requests are emitted from by the project manager according to the problem resolution process.
- When a change request is accepted by the project manager/product manager, a branch is created in github.
- The branch identification is the title of the change, prefixed by the problem
- Branch content links to the related changes

Multiple configuration:

- Changes requests of configuration files are emitted by the product manager according to the production procedure
- When a change request is accepted by the project manager/product manager, a branch is created in github
- The branch identification is the title of the change, prefixed by the relevant configurations
- Branch content links to the related changes

4.2 Documentation Configuration Change Management

- Change requests are made for changes to documentations that are already written and have a set baseline.
- The project manager decides on whether to accept or reject the change request.
- After the changes are approved, the changes are made on the document and the changes are reflected in the respective VDDs.

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.

This is useful in SportZ for processes such as problem management, audits, compliances, and investigation. In the event of suspected problems, the verification of baseline configuration and approved modifications can be quickly determined.

5.1.1 Evolution’s 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.
- Software/Code: The software configuration management tool records, for each source file or group of source files, the commit history and the changes made to the source code.
- Configuration item: The VDD of the article identifies the modification sheet included in the current version.

This helps members within the project team of SportZ to provide a single source of truth and absolute transparency between systems in the development of the game. With the modification of items, we achieve data integration from both up and downstream – from the team that plan, design and create software, to the team that build, deploy and maintain the software.

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,

The SCM oversees writing the VDD.

5.1.3 Configuration status diffusion

The SCM and the QM oversee writing 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 in chronological order.

5.2 Configuration audits

The list of audits that will be conducted during the life cycle of the project are listed below:

- Baseline audit, which is the validation of the documents and specifications on which test cases are designed.
- Functional configuration audit, which is a formal examination of a configuration item, or system, to verify that it meets the requirements specified in its functional and/or allocated documentation
- Software configuration audit, which is used to verify that all the software product satisfies the baseline needs

The above are the independent reviews of software for the purpose of assessing compliance with established performance requirements and functional, allocated, and product baselines for SportZ.

5.3 Reviews

Configuration reviews will be performed periodically to verify the correctness of the current state of configuration. The main goal of the configuration review is to verify that all project components are correctly identified, and relevant changes to any documents have been duly noted.

The SCM is to ensure that the reviews are done at least once every two weeks and any identified issues are to be placed at the highest priority to be resolved. Members in SportZ are to report to the SCM if there are such changes.

5.4 Configuration management plan maintenance

Maintenance of the configuration management plan will be done by the Quality Assurance team, which will be done weekly. This is headed by the QM which will ensure the configuration management plan is updated and component correctness is verified.

In SportZ the QM and SCM are to work hand in hand and keep each other updated on the changes in the configuration management plan as well as ensuring the team adheres to the protocols that were set.