



The Team

Configuration Management Plan

E-Tendance - Facial-Recognition Based Attendance Taking System

The Team

Li Shanlan
Zeng Jinpo
Akshaya Muthu
Simon El Nahas Christensen
MN Shaanmugam
Cao Ngoc Thai

Submitted to:



Version History

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Shaanmugam	11/09/2019	Simon	11/13/2019	Initial Configuration Management Plan draft

Table of Contents

Identification	4
1.1 Document Overview	4
1.2 Abbreviations	5
1.3 Project References	5
Organization	5
2.1 Activities and Responsibilities	6
2.1.1 Decisions Process and Accountabilities	6
Configuration Identification	7
3.1 Identification Rules	7
3.1.1 Identification Rules of Configuration Items	7
3.1.2 Identification Rules of SOUPs	8
3.1.3 Identification Rules of Documents	8
3.2 Reference Configuration Identification	8
3.3 Configuration Baseline Management	9
Configuration Control	9
4.1 Change Management	10
4.2 Evolutions Control of SOUP Items	10
Configuration Support Activities	11
5.1 Configuration Status Accounting	11
5.1.1 Evolutions Traceability	11
5.1.2 Setting up Configuration Status	11
5.1.3 Configuration Status Diffusion	11
5.1.4 Configuration Status Records Storage	11
5.2 Configuration Audits	12
5.3 Reviews	12

1. Identification

1.1 Document Overview

This document contains the Software Configuration Management Plan of E-Tendance, a facial recognition-based attendance-taking system. A Software Configuration Management Plan is used to identify the configuration of a system at discrete points in time, for the purposes of systematically controlling changes to the configuration and maintaining the integrity and traceability of the configuration throughout the system life-cycle.

1.2 Abbreviations

- **SCM:** Software Configuration Manager
- **PM:** Project Manager
- **QAM:** Quality Assurance Manager
- **SCI:** Software Configuration Item
- **SOUP:** Software of Unknown Pedigree
- **SQL:** Structured Query Language
- **VDD:** Version Delivery Description

1.3 Project References

#	Document Title
1	Team Information
2	System Requirements Specifications
3	Capability Maturity Model Integration (CMMI) Level 2 Definition

2. Organization

The software configuration is managed by some of the members of the project itself. The PM will take up the responsibilities of an SCM, which includes tracking and controlling changes in the software. The PM will be heavily assisted by the QAM, who also helps in revision control, as well as the establishment of certain baselines.

2.1 Activities and Responsibilities

The functions required to manage the configuration of the software and the people who are responsible for it 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	Back-End Developer
Install the software configuration repository tool and set up the database	Back-End Developer
Manage and structure the reference space	SCM
Define the configuration processes	SCM

Activities during the Project Life-Cycle	Person Responsible
Export components, for modification, test or delivery	SCM
Set under control validated components	SCM
Create version and write version delivery document	Lead Developer
Approve reference configurations	PM
Verify versions to be delivered and authorize deliveries	PM
Backup spaces	SCM
Do configuration audits	QAM
Inspect configuration records	QAM
Archive reference version	SCM

Management Activities	Person Responsible
Manage versions and archives	QAM
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	QAM

2.1.1 Decisions Process and Accountabilities

Responsibilities during reviews, audits and approvals are listed below:

At the End of an Activity of the Project	Person Responsible
Do a configuration freeze	PM
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	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	QAM
Present the records of the documentation management process	SCM

3. Configuration Identification

3.1 Identification Rules

3.1.1 Identification Rules of Configuration Items

An SCI is a component or a part of a software product.

The identification of a configuration item is based on the following schema:

- “XXX_<version_number>” (e.g. “frontEnd_1.0”)
 - ‘XXX’ is the name of the component in lower camelCase
 - ‘<version_number>’ is the version of the component, with a decimal place

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 was modified.

3.1.2 Identification Rules of SOUPs

SOUP is a software that has not been developed with a known software development process or methodology, or which has unknown safety-related properties.

The identification of a SOUP is very similar to that of a configuration item:

- “XXX_<version_number>” (e.g. “SQL_14.0”)
 - ‘XXX’ is the abbreviation of the name of the software used in CAPS
 - ‘<version_number>’ is the latest version of the software released by the manufacturer

3.1.3 Identification Rules of Documents

The identification of documents is described below:

- “XXX - The Team_V<x.x>_DDMMYY”
(e.g. “ProjectPlan - The Team_V1.0_090919”)
 - ‘XXX’ is the full name of the document in UpperCamelCase
 - ‘V<x.x>’ is the version number of the document
 - ‘DDMMYY’ is the Date, Month and Year of the document since it was last modified

3.2 Reference Configuration Identification

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

Each reference configuration is defined by:

- An identifier
- Its content listed in the corresponding VDD document
- The acceptance or validation reviews associated to the building of the reference configuration

3.3 Configuration Baseline Management

1. Functional Baseline (FBL)

FBL describes the required functional and interface characteristics of the overall system, and the verification required to demonstrate the achievement of those specified characteristics. This baseline is derived from the 'CMMI Level 2 Definition Document', and usually includes a detailed functional performance specification for the overall system, as well as the tests necessary to verify and validate overall system performance. The FBL is normally controlled during the System Functional and Verification Review/Audit.

2. Allocated Baseline (ABL)

ABL describes the design of the functional and interface characteristics of the configuration items making up a system, including how system function and performance requirements are allocated across lower-level configuration items. It also includes all characteristics that are allocated from the top-level configuration items, their derived requirements, and their design constraints. The performance of each configuration item is described in its Preliminary Design Specification and as are the tests necessary to verify and validate the performance of the configuration items.

3. Product Baseline (PBL)

PBL consists of completed and accepted system components and the documentation describing all of their necessary functional and physical characteristics, including the ones for production acceptance testing. Therefore, the specifications for both the hardware and software can be found in PBL (e.g., engineering drawings, materials specifications, software modular design, etc.). The initial product baseline is usually established and put under configuration control at each configuration item's Critical Design Review.

4. Configuration Control

Configuration control is a procedural method which ensures quality and consistency when changes are made in the configuration object. The change request is submitted to the SCM for further action by uploading the changed code to the GitHub repository. The SCM will then check the change request based on the possible side effects and overall impact on other configuration objects. This will result in a stable software development environment.

4.1 Change Management

Process for controlling changes to the baselines and for tracking the implementation of those changes:

- Changes are suggested by the developers during the coding process
- The SCM analyzes the changes recommended by the developer for dependencies
- The SCM informs the PM after making a decision
- The PM approves or rejects the decision

4.2 Evolutions Control of SOUP Items

SOUPs are frozen at the beginning of the project. This is to prevent the breaking of any working piece of code using the services of the SOUP item. The update of SOUPs are only considered if it makes the code more efficient during runtime.

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 life-cycle. All software and related documentation should be tracked throughout the software life-cycle.

5.1.1 Evolutions Traceability

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

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 by revision marks if possible.
- Source File - The software configuration management tool records a comment for each source file describing the modification.
- Configuration Item - The VDD of the article identifies the modification sheet included in the current version.

5.1.2 Setting up Configuration Status

The SCM prepares the VDD document, and sets up the state of all versions and of each configuration article with:

- Label
- Version Number
- Creation Date of the VDD

5.1.3 Configuration Status Diffusion

The QAM helps the SCM to prepare the VDD document.

5.1.4 Configuration Status Records Storage

The records are stored in a configuration folder, containing:

- Requests, sorted by the respective record numbers
- All Software Documents, including the VDD
- Configuration States, sorted in a chronological manner

5.2 Configuration Audits

1. Functional Audit

The objective of the functional audit is to provide an independent evaluation of a software product, verifying that its configuration items' actual functionality and performance is consistent with the relevant requirement specification. This audit is held prior to software delivery to verify that all requirements specified in the Software Requirements Specification have been met.

2. Physical Audit

The objective of the physical audit is to provide an independent evaluation of a software product's configuration items to confirm that all components in the as-built version map to their specifications. Specifically, this audit is held to verify that the software and its documentation are internally consistent.

5.3 Reviews

Software configuration reviews verify that all software product satisfies the baseline needs, ensuring that what is built is what is delivered. The SCM checks whether the defined processes were carried out according to the plan. The SCM also ensures that traceability is maintained among the different documents and procedures are in compliance with configuration control standards.