



ADDIS ABABA  
**SCIENCE AND  
TECHNOLOGY**  
UNIVERSITY  
UNIVERSITY FOR INDUSTRY

COLLEGE OF ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

SOFTWARE CONFIGURATION MANAGEMENT PLAN (SCMP)  
Simple User Login & Task List System

Name	ID
1. Mikias Goitom -----	ETS1080/14
2. Natnael Fisseha -----	ETS1231/14
3. Natnael Necho -----	ETS1266/14
4. Tamirat Dejene -----	ETS1518/14
5. Temesgen Ababayehu -----	ETS1534/14

**Submitted to:** Instr. Yibrah  
**Submission Date:** Dec 30, 2025

## Table of contents

1. Purpose and Scope	1
1.1 Purpose	1
1.2 Scope	1
2. Project Overview	1
System Features:	1
3. SCM Objectives	1
4. Roles and Responsibilities	2
5. Configuration Item (CI) Identification	2
5.1 Configuration Items	2
5.2 CI Naming Convention	2
6. Version Control and Versioning Rules	3
6.1 Versioning Scheme	3
6.2 Version Updates	3
7. Branching Strategy	3
Branches:	3
Branching Rules:	3
8. Change Control Process	3
8.1 Change Request (CR) Process	3
8.2 Change Log	4
9. Baseline Management	4
9.1 Baseline Definition	4
9.2 Project Baselines	4
9.3 Baseline Identification	4
10. Release Management	4
10.1 Releases	4
10.2 Release Notes	5
11. Configuration Audit	5
11.1 Physical Configuration Audit (PCA)	5
11.2 Functional Configuration Audit (FCA)	5
12. Tools and Environment	5
13. SCMP Approval	6

# 1. Purpose and Scope

## 1.1 Purpose

The purpose of this Software Configuration Management Plan (SCMP) is to define the processes, roles, tools, and procedures used to manage and control configuration items (CIs) for the *Simple User Login & Task List System* project.

This plan ensures that all project artifacts are properly identified, versioned, controlled, and auditable throughout the project lifecycle.

## 1.2 Scope

This SCMP applies to:

- Project documentation
- Source code files
- Configuration items
- Change requests
- Baselines
- Releases

This SCM plan focuses on **SCM practices and process implementation**, not on software complexity.

# 2. Project Overview

The *Simple User Login & Task List System* is a small web-based prototype designed to demonstrate Software Configuration Management concepts.

## System Features:

- Login page (basic UI, no real authentication)
- Dashboard page displaying a list of tasks
- One core function: adding a new task
- Simple data storage using a JSON file

The project uses simple web technologies and GitHub for configuration control.

# 3. SCM Objectives

The objectives of SCM in this project are to:

1. Identify and manage configuration items.
2. Maintain version control using Git and GitHub.
3. Control and track changes through a Change Request (CR) process.
4. Establish and manage baselines.
5. Manage releases in a controlled manner.
6. Conduct configuration audits to ensure consistency and correctness.

## 4. Roles and Responsibilities

Role	Responsibility
SCM Manager	Oversees SCM process, approves changes, manages baselines and releases
Developer	Implements features and approved changes
Documentation Manager	Maintains project documents
Quality Assurance (QA)	Verifies correctness and consistency
Auditor	Conducts configuration audits

Each team member is responsible for the SCM procedures defined in this plan.

## 5. Configuration Item (CI) Identification

### 5.1 Configuration Items

Configuration Items include all project artifacts that need to be controlled.

Examples:

- SCM Plan document
- CI Register
- Change Request forms
- HTML, CSS, and JavaScript files
- JSON data file
- README file

### 5.2 CI Naming Convention

CIs shall be named using the following format:

- `<CI_Name>_<Version>.<extension>`

Example:

- `login_v1.0.html`
- `SCMP_v1.0.pdf`

## 6. Version Control and Versioning Rules

### 6.1 Versioning Scheme

- **Documents:** v1.0, v1.1, v2.0
- **Source Code:** v1.0, v1.1
- **Releases:** v1.0, v1.1

### 6.2 Version Updates

- Minor changes increment the decimal (v1.1).
- Major changes increment the whole number (v2.0).

## 7. Branching Strategy

GitHub is used as the version control system.

### Branches:

- **main** – stable branch
- **feature-login** – login feature development
- **feature-dashboard** – dashboard and task feature development

### Branching Rules:

- All development occurs in feature branches.
- Pull Requests (PRs) are used to merge changes into **main**.
- PRs must be reviewed before merging.

## 8. Change Control Process

### 8.1 Change Request (CR) Process

All changes must be submitted through a Change Request form.

Steps:

1. Submit Change Request
2. Review by SCM Manager
3. Approval or rejection
4. Implementation of approved changes
5. Update Change Log

## **8.2 Change Log**

All approved and implemented changes are recorded in the Change Log document.

# **9. Baseline Management**

## **9.1 Baseline Definition**

A baseline is a formally approved version of a set of configuration items.

## **9.2 Project Baselines**

- **Baseline 1 (BL1):**
  - Repository structure
  - Initial documentation
  - CI Register
- **Baseline 2 (BL2):**
  - Working prototype
  - Approved Change Requests implemented

## **9.3 Baseline Identification**

Baselines are tagged in GitHub using:

- BL1
- BL2

# **10. Release Management**

## **10.1 Releases**

Two releases are planned:

- **Release v1.0** – Initial working system
- **Release v1.1** – System after CR implementations

## 10.2 Release Notes

Each release includes release notes describing:

- Features included
- Changes made
- Known limitations

Releases are managed using GitHub Releases.

# 11. Configuration Audit

## 11.1 Physical Configuration Audit (PCA)

Ensures:

- All CIs are present in the repository
- Correct naming and versioning
- Documents match repository contents

## 11.2 Functional Configuration Audit (FCA)

Ensures:

- Approved CRs are implemented
- System functionality matches requirements

Audit results are documented in the Configuration Audit Report.

# 12. Tools and Environment

Tool	Purpose
Git	Version control
GitHub	Repository hosting, PRs, releases
HTML/CSS/JavaScript	Frontend development
JSON	Data storage

Google docs	Documentation
-------------	---------------

### 13. SCMP Approval

This Software Configuration Management Plan is approved and effective upon agreement by the project team.

**Prepared by:** SCM Manager (Natnael Fisseha)

**Approved by:** Project Team (Mikias Goitom, Natnael Necho, Tamirat Dejene, and Temesgen Ababayehu)