ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

COLLEGE OF ELECTRICAL AND MECHANICAL ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING

Course Name:Software Configuration Management

Section: B

Deliverable 1

Group Members

1. Dawit Sema………………...ETS 0455/14

2. Dawit Berhanu………….……..ETS 0457/14

3. Delal Mohammad …………...ETS 0482/14

4. Eden Melaku…………….…………………..….............. ETS 0496/14

5. Eyerusalem Kidane……………………………….……… ETS 0574/14

6. Gelila Nebiyu……...................……………………………..ETS 0690/14

7. Hana Abiyu…………………………………………………..ETS0728/14

Submitted to:Mr.Yibrah

Submission Date:30/12/2025

Table of Content

Software Configuration Management Plan (SCMP)

1. Purpose and Scope

1.1 Purpose

The purpose of this Software Configuration Management Plan (SCMP) is to define the processes, roles, and tools used to manage configuration items, changes, versions, baselines, and releases for the TO-DO LIST web application.

This plan ensures:

- Controlled identification and management of configuration items (CIs)
- Formal change control using Change Requests (CRs)
- Proper versioning, branching, and baseline management
- Consistency between documentation and the GitHub repository

Git and GitHub are used as the primary SCM tools throughout the project lifecycle.

1.2 Scope

This SCMP applies to all software and documentation artifacts of the TO-DO LIST system from repository initialization to final release (v1.1).

In scope:

- React source code (App.tsx, Login.tsx, Dashboard.tsx)
- Authentication and dashboard functionality
- Project documentation (SCMP, CI Register, CRs, Baselines)
- SCM activities: version control, branching, change control, releases, and audits

The focus of this project is demonstrating SCM practices rather than full-scale production functionality.

2. Roles and Responsibilities in SCMP

2.1 SCM Roles Overview

The following roles support the SCM process:

- Project Manager

- Configuration Manager
- Implementation Lead (Code Owner)
- Developer / Team Member
- Reviewer / Approver
- Tester / Verifier
- End User (Indirect role)

2.2 Roles and Responsibilities

Project Manager

- Oversees project execution
- Approves baselines and releases
- Ensures compliance with SCM policies
  Authority: Final approval of releases

Configuration Manager (SCM Lead)

- Maintains the SCMP and CI Register
- Controls versioning, baselines, and releases
- Reviews and approves Change Requests
- Conducts configuration audits (PCA/FCA)
  Authority: Approves/rejects CRs

Implementation Lead (Person B / Code Owner)

- Implements approved CRs
- Manages feature, release, and hotfix branches
- Submits and merges Pull Requests
  Authority: Code integration

Developer / Team Member

- Develops features on assigned branches
- Follows SCM standards
- Participates in code reviews
  Authority: Limited to development branches

Reviewer / Approver

- Reviews Pull Requests
- Ensures compliance with approved CRs
  Authority: PR approval

Tester / Verifier

- Verifies CR implementation
- Supports Functional Configuration Audits
  Authority: Validation only

End User (Indirect Role)

- Provides feedback and defect reports
  Authority: None

3. Configuration Item (CI) Identification & Naming Conventions

3.1 Configuration Item Identification

CI Name	Category	Description
Requirements.docx	Document	Functional requirements
UI-Mockups.pdf	Design	UI layout
App.tsx	Source Code	Main application logic
Login.tsx	Source Code	Login component
Dashboard.tsx	Source Code	Task dashboard

style.css / Tailwind	Source Code	Styling
localStorage	Data	Task storage
README.md	Document	Project overview
SCMP.docx	Document	SCM Plan

3.2 Naming Conventions

- **Documents:** `DocumentName_Version.ext`
  Example: `SCMP_v1.0.docx`
- **Source Code:** lowercase descriptive names
- **Branches:**
  - `feature/<feature-name>`
  - `release/<version>`
  - `hotfix/<issue>`
- **Tags/Releases:** `vMAJOR.MINOR` (e.g., v1.0)

4. Versioning Rules

Semantic Versioning (SemVer) is used.

Format: `vMAJOR.MINOR`

- **MAJOR:** New baseline or significant changes
- **MINOR:** Enhancements or CR implementations

Examples:

- v1.0 – Initial release
- v1.1 – Release after CRs

Documents are versioned incrementally upon approved changes.

5. Branching Model

A simplified Git Flow–based model is adopted.

5.1 Branch Types

- **main:** Stable production-ready code
- **feature/***: New feature development
- **release/***: Release preparation
- **hotfix/***: Urgent post-release fixes

5.2 Branch Workflow

1. Feature branch created from `main`
2. Development and commits performed
3. Pull Request submitted
4. Review and approval
5. Merge into `main`
6. Release branch created and tagged

6. Change Control Process

All changes follow a formal Change Request (CR) process.

6.1 Change Control Steps

1. CR submitted (CR-001 to CR-003)
2. Impact analysis performed
3. CR approved or rejected
4. Approved CR implemented in feature branch
5. Pull Request reviewed and merged
6. CR status updated to *Closed*

6.2 Change Documentation

- Change Request Forms
- Change Log
- GitHub Pull Requests referencing CR IDs

7. Baseline Management

A baseline represents an approved system snapshot.

7.1 Defined Baselines

- **Baseline 1 (BL1):**
  Repository setup, initial documents, CI Register
- **Baseline 2 (BL2):**
  Working prototype with CR implementations

Each baseline is tagged in GitHub.

7.2 Baseline Control

- Baselines are read-only
- Changes require approved CRs
- Each baseline has a Baseline Record

8. Tools Used

8.1 Version Control and Repository Management

- **Git**
  Used for tracking source code and documentation changes, maintaining version history, and supporting branching and merging.
- **GitHub**
  Used as the central repository for the project:
  - Source code hosting
  - Branch management
  - Pull Requests
  - Tagging baselines
  - Release management
- Repository link:
  https://github.com/Gelila-Nebiyu/todo-scm-project

8.2 Development Tools

- **Visual Studio Code (VS Code)**
  Used for writing and editing React (TypeScript) source code and documentation files.
- **Node.js / npm**
  Used for managing project dependencies and running the development environment.

8.3 Documentation Tools

- **Microsoft Word / Google Docs**
  Used for preparing SCM documents such as:

    - SCMP
    - CI Register
    - Change Request Forms
    - Baseline Records
    - Audit Reports
- **Microsoft Excel / Google Sheets**
  Used for maintaining the CI Register and Change Log.

8.4 Supporting Technologies

- **React with TypeScript**
  Used to implement the TO-DO LIST application interface.
- **Tailwind CSS**
  Used for styling the user interface.