Universidad de Guadalajara
Centro Universitario de los Valles

SOFTWARE CONFIGURATION MANAGEMENT

Software Configuration Management Plans

Teacher: Omar Ali Zatarain Duran

Author: Elizabeth Baños Arias

September 25, 2025

TABLE OF CONTENTS

1. INTRODUCTION 1

1.1 PURPOSE 1
1.2 SCOPE 1
1.3 KEY DEFINITIONS 1

2. SCM MANAGEMENT 1

2.1 ORGANIZATION 1
2.2 RESPONSIBILITIES 2
2.3 APPLICABLE POLICIES 2

3. SCM ACTIVITIES 2

3.1 CONFIGURATION IDENTIFICATION 2
3.2 CHANGE CONTROL 3
3.3 STATUS ACCOUNTING 3
3.4 AUDITS AND REVIEWS 3
3.5 INTERFACE CONTROL 3
3.6 SUPPLIER CONTROL ..... ¡ERROR! MARCADOR NO DEFINIDO.

4. SCM SCHEDULES 4

5. SCM RESOURCES 4

5.1 SOFTWARE TOOLS 4
5.3 TEAM AND STAFF 5
5.4 TRAINING 5

6. SCM PLAN MAINTENANCE 6

6.1 SUPERVISION RESPONSIBILITY 6
6.2 FREQUENCY OF UPDATES 6
6.3 EVALUATION AND APPROVAL OF CHANGES 6
6.4 IMPLEMENTATION AND COMMUNICATION OF CHANGES 7

1. INTRODUCTION

1.1 Purpose

This plan defines the software configuration management (SCM) activities that will be applied to the development of the academic and administrative system for the graduate program in Software Engineering at CUValles.

1.2 Scope

The plan covers all configuration elements related to:

• Users (authentication, registration, password recovery).

• Students, professors and academic procedures.

• Technical documentation and user manuals.

• Database scripts.

1.3 Key Definitions

Acronym	Full Form
SCM	Software Configuration Management
CI	Configuration Item
CCB	Configuration Control Board / Change Control Board

2. SCM MANAGEMENT

2.1 Organization

The project is organized into three architectural layers:

- **Presentation (Frontend):** shows the information and has the forms for the users.
- **Business logic (Backend):** manages the academic and administrative rules.
- **Data (Database):** saves the information about students, teachers, and procedures.

SCM activities are used in all layers to keep version control, traceability, and consistency.

2.2 Responsibilities

Role	Responsibilities
Configuration Manager	Makes sure the plan is followed, controls the repository, defines baselines, and organizes audits.
CCB (Configuration Control Board)	Reviews and approves changes. It has the project manager, lead analyst, lead developer, and academic representative.
Developers	Do the approved changes, keep the code correct in the repository, and write the documentation of the changes.
Testers/QA	Test the changes, check that they follow the requirements, help with configuration audits, and report problems to the Configuration Manager and the CCB.
Key Users	Check that the system works well for academic and administrative tasks.

2.3 Applicable Policies

• You cannot add changes to the main repository without approval from the CCB.

• All changes must be written in a Request for Change (RFC) with a reason for the change.

• QA must test and check all changes before they go to production.

• Every software release must include updated technical documents and the user manual.

3. SCM ACTIVITIES

3.1 Configuration Identification

Items under control:

- Source code (frontend in React, backend in Django/Python).
- SQL scripts (structure and procedures).
- User manual and technical manual.
- Test cases.
- Analysis and design documents.

3.2 Change Control

- Change request registered with RFC.
- CCB checks impact on time, cost, and quality.
- Code added to repository only with approved pull requests.

3.3 Status Accounting

- Versions saved in Git.
- Changes written in *changelog.md*.
- Monthly reports about status of items and changes.

3.4 Audits and Reviews

- Internal audits at the end of each sprint.
- One audit required before a stable release.

3.5 Interface Control

- Control of external dependencies (Bootstrap, Django, React, MySQL).
- Record of approved versions of external libraries.

4. SCM SCHEDULES

Phase / Activity	SCM Milestone	Date	Dependencies
Project start	Appointment of the Configuration Manager and formation of the CCB	Week 1	Project start
Requirements analysis	Baseline of requirements	Week 2	Approved requirements document
System design	Design baseline	Week 3	Approved requirements
Iterative development	Change control + sprint closures	Week 4 – 7 (2 sprints)	Design baseline
System testing	Baseline testing and results	Week 8 – 9	Implemented functionalities
Release of the first stable version (v1.0.0)	Product baseline and documentation	Week 10	Test baseline passed
Maintenance and evolution	Incremental releases (v1.1.0, v1.2.0, etc.)	Week 11 – 12	Stable version released

5. SCM RESOURCES

5.1 Software Tools

- **Version control:** GitHub/GitLab → manage source code, branches, and releases.

- **Change management and issue tracking:** Jira or Trello → register RFC, follow tasks, and create status reports.
- **Documentation:** Confluence / Google Drive → control and version of user manuals, technical documentation, and CCB records.
- **Database:** MySQL → structure and data scripts under configuration control.
- **Automated testing:** PyTest / Selenium → validate changes before merging into the main branch.

Each tool will be under configuration control with:

- Defined users and permissions.
- Regular backups.
- Change history.

5.3 Team and Staff

- **Configuration Manager:** 1 person, responsible for applying the plan and doing audits.
- **CCB:** multidisciplinary committee (director, lead analyst, main developer, academic representative).
- **Developers:** 2–3 people, responsible for implementing approved changes.
- **QA/Testers:** 1–2 people, responsible for running tests and validating releases.

5.4 Training

- **Git and version control:** basic training for all developers and QA.
- **Change management (RFC, CCB):** training in SCM policies and procedures.

- **Management tools (Jira/Trello):** introduction sessions for using change control boards.

6. SCM PLAN MAINTENANCE

6.1 Supervision Responsibility

- The Configuration Manager is responsible for supervising the SCM plan, coordinating reviews, and suggesting adjustments.
- The CCB (Configuration Control Board) has the authority to approve changes to the plan.

6.2 Frequency of Updates

- The plan will be reviewed at the start of each main phase of the software life cycle:
    - Requirements analysis.
    - Design.
    - Development.
    - Testing.
    - Release and maintenance.
- The plan can also be updated in special cases when:
    - Inconsistencies are found with the real project practice.
    - New SCM tools or policies are introduced.
    - The CCB requests changes because of significant changes in project scope.

6.3 Evaluation and Approval of Changes

- Every proposed change to the SCM plan will be registered in a Change Request (RFC) specific for management documents.
- The Configuration Manager evaluates the impact of the change and presents it to the CCB.

- The CCB approves, rejects, or delays the modification.
- Only approved changes are officially added to the plan.

6.4 Implementation and Communication of Changes

- Changes to the plan will be included in a new version of the document, with version number (example: v1.1, v1.2).
- The revision history will be kept in a section for version control of the SCMP.
- Once approved, the Configuration Manager will share the new version with the project team through official channels (document repository or institutional email).
- Each update of the plan must include a changelog that explains the modifications.