

Software Configuration Management Plan

Library Management System

Meghana N Naik – 2023BCSE07AED584
CSE General

1. Introduction

Software Configuration Management (SCM) is a disciplined process used to systematically manage changes to software products throughout the development life cycle. The purpose of this Software Configuration Management Plan is to define the activities, roles, and procedures required to manage configuration items for the Library Management System (LMS).

The Library Management System is a software application designed to automate library operations such as managing books, users, issue and return processes, fine calculation, and report generation. Since the system evolves over time due to bug fixes, enhancements, and requirement changes, an effective SCM plan is essential to maintain consistency, control changes, and ensure product integrity.

This document describes the configuration identification, control, status accounting, and audit procedures to be followed during the development and maintenance of the Library Management System.

2. Configuration Identification

Configuration identification involves selecting and uniquely identifying all configuration items (CIs) that need to be managed and controlled.

2.1 Configuration Items

The following items are identified as configuration items for the Library Management System:

- Requirement documents (SRS)
- Software Design Document (SDD)
- Source code files
- Database scripts and schemas
- Test plans and test cases
- User manuals and installation guides

Each configuration item is assigned a unique name, version number, and storage location.

2.2 Versioning Strategy

A version control strategy is adopted to track changes to software and documents:

- Major versions indicate significant functional changes (e.g., v2.0)
- Minor versions indicate enhancements or improvements (e.g., v1.2)
- Patch versions indicate bug fixes (e.g., v1.2.1)

Version history is maintained for all configuration items to support traceability and rollback when required.

3. Configuration Control

Configuration control ensures that changes to configuration items are properly reviewed, approved, and implemented in a controlled manner.

3.1 Change Request Process

- All change requests are formally documented using a Change Request Form.
- Change requests may arise due to defect fixes, new requirements, or performance improvements.
- Each request includes description, reason, impact analysis, and priority.

3.2 Change Control Authority

A Change Control Board (CCB) is responsible for evaluating and approving changes. The board typically includes:

- Project Manager
- Lead Developer
- Quality Assurance representative
- Client or librarian representative

Only approved changes are implemented in the system.

3.3 Change Implementation

- Approved changes are assigned to developers.
- Changes are implemented in a controlled environment.
- Modified configuration items are updated with new version numbers.
- Changes are tested before being merged into the main system.

4. Configuration Status Accounting

Configuration status accounting provides accurate information about the status of configuration items and change requests.

4.1 Status Tracking

The following information is recorded and maintained:

- Current version of each configuration item
- Change request status (submitted, approved, rejected, implemented)
- Date of last modification
- Person responsible for the change

4.2 Reporting

Periodic reports are generated to provide visibility into:

- Completed and pending changes
- Version history of the Library Management System
- Configuration baselines and releases

These reports help stakeholders monitor project progress and ensure transparency.

5. Configuration Audits

- Test Case ID
- Test Description
- Input Data
- Expected Output
- Actual Output
- Pass/Fail Status

5.1 Functional Configuration Audit

- Ensures that the system functions match approved requirements.
- Verifies that all approved changes are correctly implemented.

5.2 Physical Configuration Audit

- Verifies that all configuration items are properly documented.
- Confirms version numbers, file names, and storage locations.
- Ensures consistency between documents, source code, and executables.

