**Indian Institute of Information Technology Pune**

NAME: Deepesh Patil

GROUP: 2

MIS: 112215055

YEAR/SEM: 3rd Year/5th Sem

**Software Configuration Management(SCM) Plan**

1. Introduction

1.1. Purpose

The Software Configuration Management(SCM) plan will enumerate procedures and processes under which change in software will be controlled, version management effected, and all software artifacts maintained with integrity and traceability for the "Online Course Reservation System".

1.2 Scope

This policy addresses all source code, configuration files, libraries, databases, and documentation within the Online Course Reservation System. It shall be applicable to all environments-development, test, and production-and to all people who are involved in the development of the systems, including developers, testers, project managers, and configuration managers.

2. Configuration Management Team Roles and Responsibilities

2.1. Configuration Manager

Preparation and maintenance of SCM procedures;

Leading the change control process.

Everything in the system is identified, documented, and versioned .

2.2 Developers

SCM procedures

Maintain local copy of code & commit only after their changes are approved

All changes made to the software should be appropriately documented and linked to the correct configuration items.

2.3 Testers

Confirm that the right versions of software components have been used & mark any problems related to versions and configurations

2.4 Project Manager

Getting Clear Approvals for Major changes to configurations not affecting the project's schedule & its deliverables

3. Configuration Identification

3.1. Configuration Items (CI)

Configuration items (CIs) of the "Online Course Reservation System" are among others

Source Code: All source programming files Java, Python etc.

Databases: Database schematics, scripts

• Libraries : third party and self developed libraries

Documentation : user guide, system architecture

Configuration Files : Files which determine the application behavior are different between development and production environment

3.2. Naming Conventions

All CIs shall have a standardized naming convention to avoid ambiguity and traceability.

Examples :

Source Code : module-functionality-version.extension

Database Schemas : db-entity.sql

Documents : project-version.ext

4. Version Control:

4.1. Version Control System (VCS)

The system will use a distributed version control system such as Git to manage versions of the software. Every CI shall have a name that is unique in terms of version number and change history.

4.2. Versioning Strategy

• Major Versions (1.x.x): Complete overhaul or function reconstruction

• Minor Versions (x.1.x): Incremental changes or updates

• Patch Versions (x.x.1): Bug fixing or minimal modification

5. Change Management

5.1 Change Control Board (CCB)

All major changes must have Change Control Board (CCB), which consists of the Project Manager, Configuration Manager, and technical leads, approve.

5.2 Change Request Process

1. Request submission: A Change Request (CR) might be submitted by any interested party.

2. Impact Analysis: The impact on the system is analyzed by the Configuration Manager .

3. Approval/Denial: Based upon analysis, CCB analyzes the impact and accepts or declines the request .

4. Implementation: Once the request is accepted, the change is implemented by the development team .

5. Review: The change is tested and reviewed before it is integrated in the production environment.

6. Configuration Audits and Reviews

6.1. Audits

SCM policies will be checked periodically through configuration audits. The configuration audits will ascertain the fact that all items of software have been correctly identified and versioned and stored appropriately.

6.2. Audits

Configuration auditions will be conducted at significant milestones for ensuring the proper adherence of SCM practices.

**Risk Management Plan**

1. Introduction

1.1. Purpose

The purpose of this document is to determine, evaluate and control any risk factors that may hamper the "Online Course Reservation System" project. Proactive planning mitigates risks that are identified early in the project lifecycle, and proper risk management helps bring the project to a successful close.

2. Risk Management Process

2.1. Identification of Risks

Risk identification involves the following:

Brainstorming sessions with the project team.

Previous project lesson learned

Expert judgment from a domain expert.

Review of project documentation

2.2 Risk Categories-

Risks can be Technical Risks like System architecture, technology stack, or integration issues.

Operational Risks : These are risks related to the operational execution of the project, typically including availability or a skills gap in the team.

Financial Risks : this relates to the budget overruns and cost escalations.

Schedule Risks : deadlines not met

Legal and Compliance Risks : these relate to failure to abide by regulatory or legal requirements.

3. Risk Assessment

3.1. Risk Probability and Impact

Risks will be assessed based on:

•Probability: Chances of its occurrence (High, Medium, Low).

•Impact: Effect when the risk happens: Project impact would be as Critical, Significant or Minor.

3.2. Risk Matrix

A risk matrix will be used to prioritize risks based on their probability and impact.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Description | Probability | Impact | Risk Level |
| 1 | Data breach from external hacking | High | Critical | High |
| 2 | Delay in integration with payment gateway | Medium | Significant | Medium |
| 3 | Insufficient server capacity for peak load | Low | Minor | low |

4. Risk Mitigation Strategies

4.1. Avoidance

Action taken to eliminate the risk. Example : Choosing a tried-and-tested payment gateway service provider who will avoid integration issues.

4.2. Mitigation

Decrease the likelihood or effect of the risk. Example : Increasing security as multi-factor authentication that lessens the likely risk of data breaches.

4.3. Transfer

Passing on the risk to a third party.

For instance : Server capacity during peak loads by using cloud services.

4.4. Acceptance

Acknowledging the risk and going ahead. Illustration: Acceptance of the fact that minor UI bugs are likely to be present in a beta version launch.

5. Monitoring and Review

5.1. Risk Log

Every risk identified is noted down in a risk log & the risk log is updated from time to time. Every risk entry should have a description, a risk owner, and a status.

5.2. Risk Reviews

Risk is to be reviewed regularly during project meetings to assess the current status and amend mitigation strategies if it's necessary.