

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

DOY!

Configuration and Change Management Report

1 Introduction

This Configuration and Change Management Report outlines the comprehensive processes, tools, and methodologies to be implemented for maintaining consistent work products throughout the software development lifecycle. It establishes a robust framework for tracking changes, maintaining version control, and ensuring that all stakeholders have access to the most current and accurate project artifacts.

The effective management of configuration items and changes is critical to project success, particularly in complex software development environments where multiple team members work simultaneously on different aspects of the system. This report addresses how the project will identify configuration items, control versions, manage changes, track issues, and ensure the integrity of all project deliverables.

Our approach integrates industry best practices with tools specifically selected to meet the unique needs of our development environment. By implementing these configuration and change management procedures, we aim to reduce risks associated with uncontrolled changes, improve communication among team members, enhance productivity through process automation, and ultimately deliver higher quality software products.

This document serves as the authoritative reference for all team members regarding how artifacts will be managed, how changes will be controlled, and what procedures must be followed to maintain system integrity throughout development and maintenance phases. It also establishes clear roles and responsibilities to ensure accountability for configuration management activities.

2 Purpose

The purpose of configuration and change management in our project is multifaceted, addressing the technical, operational, and strategic aspects of software development. Our comprehensive approach aims to:

- **Maintain Product Consistency**
 - Maintain a consistent set of work products as they evolve through development iterations
 - Establish a single source of truth for all project artifacts
 - Prevent unauthorized or undocumented modifications to controlled items
 - Ensure team members always work with the most current versions of

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

artifacts

- Preserve the history of all configuration items to support auditing and rollback capabilities

- **Ensure Build Integrity**

- Ensure reliable and reproducible software builds across all environments
- Implement automated build verification to detect integration issues early
- Maintain configuration control of build components, including third-party dependencies
- Document build procedures to enable consistent execution by any team member
- Support parallel development through proper branching and merging strategies

- **Facilitate Adaptability**

- Provide an efficient mechanism to adapt to changes and address issues as they arise
- Establish clear protocols for emergency fixes versus planned enhancements
- Enable impact analysis prior to implementing changes to understand dependencies
- Streamline the review and approval process to maintain development velocity
- Support concurrent work on different aspects of the system without interference

- **Improve Work Management**

- Enable effective tracking and planning of work items throughout their lifecycle
- Provide visibility into the status of all change requests to stakeholders
- Establish priorities based on business value, technical urgency, and resource availability
- Facilitate resource allocation through clear identification of pending work items
- Support coordination between team members working on related components

- **Enhance Project Visibility**

- Support measurement of project progress through consistent versioning
- Enable meaningful reporting on change volumes, types, and resolution

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

times

- Provide metrics to assess team performance and process efficiency
- Establish traceability between requirements, changes, and implementations
- Create an audit trail for compliance and governance requirements
- **Reduce Project Risk**
 - Minimize risk by controlling changes to configuration items
 - Prevent regression issues through controlled implementation of changes
 - Establish fallback positions through baseline management
 - Support disaster recovery through secure repository management
 - Enable root cause analysis through comprehensive change history

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

3 Configuration and Change Management Specifications

Our configuration management approach addresses the management of project artifacts, their versions, and configurations throughout the software development lifecycle.

- **Version Control System**

- **Tool:** Git with GitHub as the central repository
- **Repository Structure:**
 - Main branch for stable releases
 - Development branch for ongoing work
 - Feature branches for individual features/tasks
 - Hotfix branches for critical fixes
- **Access Control:** Role-based permissions will be implemented to ensure appropriate access levels

- **Configuration Items**

The following items will be under configuration management:

1. Source code
2. Documentation
 - Requirements specifications
 - Design documents
 - User manuals
 - Test plans
3. Build scripts and configuration files
4. Database schemas
5. Third-party libraries and dependencies
6. Test data sets

- **Builds and Releases**

- **Build Management Tool:** CI/CD pipeline
- **Build Frequency:** Daily automated builds from the development branch
- **Release Nomenclature:** [Major].[Minor].[Patch] format
- **Release Verification:** Automated test suite execution before release approval
- **Change Management**

Our change management process establishes how changes to configuration-

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

controlled artifacts are requested, evaluated, approved, and implemented.

- **Change Request Process**

1. **Initiation:** Any team member can submit a change request via the issue tracking system

2. **Documentation:** Each change request must include:

- Description of the proposed change
- Justification for the change
- **Impact assessment**
- **Resources required for implementation**

3. **Review:** The change control board (CCB) reviews requests weekly

4. **Prioritization:** Changes are prioritized based on:

- Business value
- Technical urgency
- Risk assessment
- Resource availability

5. **Implementation:** Approved changes are assigned, scheduled, and tracked

6. **Verification:** Changes are verified against requirements before closure

- **Change Control Board**

- **Composition:**

- Project Manager (Chair)
- Technical Lead
- QA Representative
- Business Analyst
- Customer Representative (when applicable)

- **Meeting Frequency:** Weekly or as needed for urgent changes

- **Decision Authority:** Consensus-based with final authority resting with the Project Manager

- **Work Items Tracking**

- **Tool:** GitLab issue tracking system

- **Work Item Types:**

- Feature requests

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

- Bug reports
- Tasks
- Technical debt items
- Documentation updates
- **Workflow States:**
 - New
 - Approved
 - In Progress
 - In Review
 - Testing
 - Closed

4 Key Considerations

As the project grows it gets harder to maintain changes and configurations. At this point we need some configuration management tools. Within the scope of the Doy! project we will be using GitLab, a web-based version control and collaboration platform. Also, we will follow GitLab Flow to keep change and configuration management activities. GitLab flow is a lightweight, branch-based workflow. Following GitLab Flow includes the following steps:

- 1. Create a Branch:** By creating a branch, one creates a space to work without affecting the default branch. Additionally, this gives collaborators a chance to review the work.
- 2. Make Changes:** A branch is a safe place to make changes. If any mistakes occur, one can revert their changes or push additional changes to fix the mistake. Changes will not affect the default branch until it merges to default branch.
- 3. Create a Pull Request:** Create a pull request to ask collaborators for feedback on changes. Pull request review is so valuable that some repositories require an approving review before pull requests can be merged. When create a pull request, include a summary of the changes and what problem they solve.
- 4. Merge Pull Request:** Once pull request is approved, merge pull request. This will automatically merge one's branch so that their changes appear on the default branch. GitLab retains the history of comments and commits in the pull request to help future contributors understand the changes. Branch protection settings may block merging if pull request does not meet certain requirements.

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

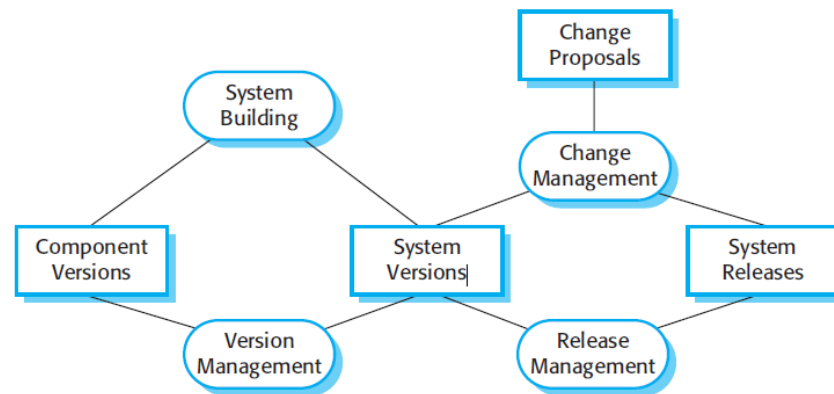


Figure 25.1
Configuration
management activities

- **Risk Management**

- Regular backups of all configuration items
- Disaster recovery procedures for repository loss
- Change impact analysis before implementation of major changes
- Automated testing to detect regression issues

- **Tool Integration**

- Integration between Git, Jenkins, and GitLab
- Automated notifications for build failures
- Traceability between code commits and work items
- Reporting capabilities for change metrics
- Manual debugging procedures integrated with issue tracking workflow

- **Team Training**

- **All team members will receive training on:**
 - Version control system usage
 - GitLab issue submission and tracking
 - Change request workflow
 - Build system interaction
 - Configuration management best practices
 - Manual debugging techniques and documentation
 - Integration between debugging and issue tracking

- **Compliance and Auditing**

- Periodic audits of configuration management processes

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

- Compliance verification with organizational standards
- Documentation of all changes for regulatory purposes
- Retention policies for configuration items

5. Traceability Table

Del3 Task / Team Member	Bariş Yıldız	Abdussamet Tekin	Muzaffer Berke Savaş	Said Çetin	Mehmet Oğuz Kocadere
Front-end Development of Critical Use Cases			X		X
Back-end Development of Critical Use Cases	X			X	
Database Query and Entity Development of Critical Use Cases		X			
Source Code Management	X				
Architectural Design Documentation		X			
Test Planning				X	
Build Configuration and Configuration-Deployment Documents					X
User Documentation			X		
Total Hours	192	192	192	192	192

DOY!	
Configuration and Change Management Report	Date: 12/04/2025

6. Prompts

The following key prompts will guide configuration and change management processes:

- **Code Commit Prompt:**
 - Which GitLab issue is addressed?
 - Have tests and debugging been completed?
 - Is the change properly isolated?