Mälardalens universitet

INTELLIGENT
REPLANNING DRONE SWARM

Mälardalen University
M.Sc.Eng. Dependable Aerospace Systems
Västerås, Sweden

Project Course in Dependable Systems
22.5 credits

Configuration Management Plan

Responsible
Yonatan Michael Beyene
yme21001@student.mdu.se

Contributors

Claire Namatovu
cnu21001@student.mdu.se

Andrea Haglund
ahd20002@student.mdu.se

Esaias Målqvist
emt21001@student.mdu.se

Emily Zainali
ezi21001@student.mdu.se

Examiner: Luciana Provenzano

November 16, 2025

DOCUMENT APPROVAL

Name	Role	Version	Date	Signature
Andrea Haglund	Chief Engineer	1.1	2025-11-16	AH

DOCUMENT CHANGE RECORD

Version	Date	Reason for Change	Pages / Sections Affected
0.1	2025-09-30	Introduced	All
1.0	2025-10-2	Reviewed	Pages: 4,7,8,9
1.1	2025-11-13	Supervisors feedback	Pages: 4,5,6,8,9

Contents

Glossary

CE

Chief Engineer. 7

CI

Configuration Items. 5, 7, 8

CM

Configuration Manager. 5

CMP

Configuration Management Plan. 4, 5

CR

Change Request. 9

IRDS

Intelligent Replanning Drone Swarm. 4, 7

QAP

Quality Assurance Plan. 7

QCM

Quality & Configuration Manager. 8

QM

Quality Manager. 7, 8

QMP

Quality Management Plan. 7

RM

Requirements Manager. 7

SM

Safety Manager. 7

TMP

Technical Management Processes. 4

VVM

Validation & Verification Manager. 7, 8

1 Introduction

This document is the Configuration Management Plan (CMP) for the project Intelligent Replanning Drone Swarm (IRDS). The CMP outlines the systematic approach for managing and controlling system elements and their configurations throughout the project life cycle. As a key component of the Technical Management Processes (TMP), the CMP defines the strategy, management, and technical procedures necessary to ensure that each product remains consistent with its defined configuration.

CM ensures strict version control, traceability of all design decisions, alignment between requirements and implementation, and verification that only approved, validated configurations are deployed. This level of control is crucial for preventing configuration drift, enabling safe replanning decisions, maintaining system integrity during iterative development, and supporting the dependability and safety goals of the IRDS project.

1.1 Purpose

The purpose of this CMP is to manage system and system element configurations throughout the project life cycle. The purpose of this management is to ensure consistency, integrity, traceability, and control.

Effective configuration management is essential for a dependable drone swarm system, where multiple autonomous drones must operate safely, consistently, and in synchronisation.

1.2 Related Documents

The standards used to create this CMP are listed in Table 2. Other related documents referred to in this management plan are listed in Table 1.

Document ID	Document Title
PP-01	Project Plan [1].

Table 1: Related documents

Standard	Year	Title
ISO/IEC/IEEE 15288	2023	System life cycle processes [2]
ISO/IEC/IEEE 15289	2011	Content of life-cycle information items (documentation) [3]
ISO 9001	2015	Understand, Implement, Succeed! [4]

Table 2: Standards used to create this CMP.

2 Scope

This CMP applies to all project deliverables, including reports, management plans, test results, safety analyses, and code files. It covers creation, review, revision, approval, and baselining (a formally approved version of a configuration) of these items, all stored under version control in a working environment used by the team. Non-controlled items include personal notes and draft versions. The plan follows mostly the standards stated in table 2.

2.1 Objectives

The objective of this CMP is to define how configuration management activities will be executed throughout the system's lifecycle. This plan ensures the integrity, traceability, and control of all identified configuration items. Specifically, the CMP details how the team performs key Configuration Management activities, including:

- Configuration Identification: This includes process of identifying and classifying Configuration Items (CI), as well as the process for establishing baselines.

- Configuration Control: Define the process for managing, evaluating, approving, and documenting changes to configuration items to ensure that only authorised and verified modifications are implemented and that all changes are traceable.

2.2 Deliverables

The Configuration Manager (CM) shall deliver:

ID	Deliverable
CM-01	Configuration Management Plan.
CM-02	Configuration Item List.
CM-03	Change Request Logs/Records.

Table 3: Deliverables.

3 Methodology

The configuration management methodology for the IRDS project is based on a controlled, lifecycle integrated approach aligned with ISO/IEC/IEEE 15288, ISO/IEC/IEEE 15289, and ISO 9001. The methodology focuses on ensuring traceability, integrity, and disciplined control of all configuration items throughout the system's development.

This project uses a baseline-driven CM approach. All changes to baselined items require formal evaluation and approval. This approach is necessary because IRDS is a dependable autonomous drone swarm system, where inconsistencies in requirements, algorithms, communication protocols, or safety analyses can lead to unsafe or unpredictable swarm behaviour.

4 Activities

To support the successful delivery of this project and maintain high standards of quality, it is important to establish clearly defined deliverables under quality oversight, well-defined roles and responsibilities, and structured review processes. Together, these elements promote accountability, consistency, and traceability throughout the project lifecycle.

4.1 Configuration Items

The IRDS project consists of various deliverables from all managers on the team. These deliverables fall under configuration management and are referred to as Configuration Items (CI). The CI are as follows:

- Safety Manager (SM)
  - Safety Management Plan .
  - Risk Level Analysis.
  - Safety goals and requirements.
  - Safety analysis.
- Requirements Manager (RM)
  - Requirements Management Plan.
  - Requirements specification.
  - Requirements Review Protocol.
  - Traceability Matrix.
- Validation & Verification Manager (VVM)
  - Validation Verification Plan.
  - Risk Analysis.
  - Verification Protocol.
  - Test Specification.
  - Validation Report.
- Chief Engineer (CE)
  - Project Plan.
- Configuration Manager
  - Configuration Management.
  - Configuration Item List.
  - Change Request Logs/Records.
- Quality Manager (QM)
  - QMP & QAP
  - Review Reports
  - Quality Impact Report
- Simulation Code

4.2 Configura Responsibilities & Roles

The configuration related responsibilities of each manager are defined below. These roles and responsibilities were established after consulting with the respective managers and by referencing ISO 9001 [4].

- CE
  - Ensure technical deliverables are properly versioned and documented.
  - Notify Configuration Manager when a new baseline should be created.
  - Use GitHub workflow (branches, etc.) correctly.
- RM

  - Ensure requirements are complete, consistent, and uniquely identified.
  - Provide traceability between requirements.
  - Ensure the specifications and project requirements are under configuration management including state, version and authorization status.
  - Submit requirement reports to QM for review before submitting.
- VVM
  - Ensure test cases, test scripts, and test reports are registered as configuration items.
  - Track versions of test artifacts.
  - Ensure test results link back to correct baselines.
- Quality & Configuration Manager (QCM)
  - Quality
    - Maintain process compliance with applicable standards and regulations.
  - Configuration
    - Ensure all Configuration Items are properly identified, controlled, and documented (Maintain traceability and clarity over project materials).
    - Maintain version integrity and baseline control (Avoid confusion over which version of a CI is "official").
    - Ensure all changes follow approved processes.

4.3 Working Environment

For the storage of files, links and relevant information for this project, GitHub is being used. GitHub is a cloud-based platform that enables developers to store, manage, and track changes to their code [5]. The reason it was chosen is because it gives integrity, security and safety to the project.

- Integrity - The items can not be corrupted or altered accidentally.
- Security - Only authorized people can access/change the items.
- Safety – Items are always stored in a way that prevents loss. Even if an accident occurs where an item is accidentally deleted or modified, it is always possible to recover or restore it to a specific point in time.

GitHub is an effective tool for managing code files and other small-sized files, but it is not well suited for large files such as report PDFs. For that purpose, Microsoft SharePoint is used. SharePoint is a platform that allows users to store, organize, share, and access information from any device [6]. GitHub is used exclusively for simulation code and database implementation, where traceability is maintained internally through commit history, branching, and links to the project's requirements database. SharePoint is used only for controlled documents such as management plans, reports, and analyses, where traceability is maintained through document versioning, approval records, and baselines. Since the artefacts stored in each platform belong to separate configuration domains, their integrity and traceability are preserved within their respective environments rather than across them. For planing and task allocating Jira was used. Jira is a project management tool that enables teams to track issues, organize projects, and streamline workflows through project. [7].

4.4 Item Identification and Tracking

To identify and track all deliverables created in this project, each deliverable is assigned a unique identification code. These codes are stored on GitHub along with a brief description of their purpose. This system ensures proper tracking of all materials and prevents duplication of identification codes. The codes consist of a combination of letters and numbers (AA-00).

All the items are available at SharePoint if anyone needs to accesses someone else work to maybe review it etc.

4.5 Version Control and Change Approval

Deliverables may require updates after submission. To maintain integrity and traceability, all changes shall follow a formal change control workflow. A version control system is used to record the history of modifications, and the version number is updated every time a deliverable is changed. All modifications must follow the controlled process described below.

4.5.1 Change Control Workflow

Any team member who identifies the need for a modification on the baseline of the deliverables needs to submits a Change Request (CR). The CR must describe:

- The reason for the change
- The changes
- An impact analysis (defining what could be impacted by the changes.)

The impact analysis will evaluate the technical impact, safety impact, requirements impact, verification impact, quality impact, or configuration impact. This is done by the configuration manager by confronting the team individually or together if the changes will affect their work.

In case disarrangement appears between the managers if the change request shall be approved or not, the final decision lies with the Chief Engineer, who has final technical authority for the project. The Safety Manager may recommend rejection for safety-critical issues, but the Chief Engineer has the formal decision-making authority.

Once the CR is approved by the chief engineer and configuration manager (make sure that the change follows CM rules), the manager who sent the CR can start modifying the deliverable.

To ensure transparency; a revision history table is maintained and updated whenever a new version is created. This table can be found on each delivery.

References

[1] A. Haglund, *Project Plan*, Intelligent Replanning Drone Swarm, Sep. 2025.

[2] ISO, IEC, IEEE, *Systems and software engineering — Content of life-cycle information products (documentation)*, Nov. 2011.

[3] ISO, *Quality management — Guidelines for configuration management*, ISO, Mar. 2017.

[4] Jarvis. A, Palmes. P, *Understand, Implement, Succeed!*, ISO, Mar. 2015.

[5] J. Holcombe, "What is github? a beginner's introduction to github," https://kinsta.com/blog/what-is-github/, 08 2025, (Accessed on 23-09-2025).

[6] Microsoft, "What is sharepoint?" https://support.microsoft.com/en-us/office/what-is-sharepoint-97b915e6-651b-43b2-827d-fb25777f446f, 08 2025, (Accessed on 25-09-2025).

[7] ProductPlaner, "Jira," productplan.com, Accessed: Sep. 29, 2025. [Online]. Available: https://www.productplan.com/glossary/jira/