

Configuration Management Plan

Introduction

The **purpose of this configuration plan** is to track the expected versioning with successive software development, as well as to define the plan's limitations and the responsibilities, authorities, policies, resources, and schedule we will have, ensuring product integrity, facilitating change control, and optimizing system control throughout its entire lifecycle.

The **scope of the configuration plan** encompasses all software components, client requirements documents, the development phase, source code, and system manuals for both the client and the server.

The configuration elements will consist of the system components developed in each iteration. Each component will correspond to an implemented module or functionality and will be managed through specific branches in the Git repository, following the GitFlow model. This ensures that each iteration produces controlled and traceable versions of the components that make up the final product.

Criteria for Identifying Configuration Elements

Configuration management will be applied to all project artifacts, such as requirements documents, source code, metadata databases, scripts, and user manuals. Each configuration item will have a unique identifier and will be controlled through the Git repository, applying semantic versioning.

Limitations and Assumptions

The main limitation is the client's financial resources and a fixed budget for development. Other limitations and assumptions include the work schedule, which is defined by 8-hour workdays, 40 hours per week with rest days on Saturdays and Sundays.

Due to the tight schedule, to meet the final delivery date, it is necessary to work on two Saturdays: the Saturday of week 1 (November 15, 2025) and the Saturday of week 3 (November 29, 2025).

Three components will be completed per week, with an exception in weeks 3 and 4 where two components will be worked on simultaneously.

Although an iteration does not necessarily constitute a component, our organizational planning has resulted in one component being completed per iteration.

The team will consist of 7 people (2 requirements specialists and analysts, 3 designers and implementers and 2 testers), with a multi-device approach to meet the client's requirements.

The system's construction will be completed on December 5th, and from that date, a transition phase will begin during which preliminary versions will be generated to validate the product's quality. From December 8rd to 11th, the *alpha version* will be tested to detect general errors; from December 12th to 17th, the *beta version*, which is more stable and focused on verifying the system's actual functionality, will be evaluated. Finally, the final version will be delivered on December 19th, ensuring that the software has been reviewed and refined through several iterations to guarantee its stability and compliance with the requirements.

Plan Responsibilities and Authorities

The Configuration Management Authority is the project **coordinator**, responsible for verifying that configuration management activities are planned and executed, and also responsible for evaluating and approving configuration changes.

Furthermore, all members of the work team are responsible for ensuring compliance with this plan and guaranteeing the correct implementation of the software development of each component.

Scheduled

The project's timeline will follow the same schedule defined in the Excel spreadsheet, with no changes to the planned dates. Each use case has an initial version, V1.0.0, representing its first complete implementation.

The **Semantic Versioning** standard will be applied for version control, ensuring consistency and traceability in subsequent software updates.

Component versioning table

Component	Construction	Date completed
Component 1	v1.0.0	15/11/2025
Component 2	v1.0.0	15/11/2025
Component 3	v1.0.0	15/11/2025
Component 4	v1.0.0	21/11/2025
Component 5	v1.0.0	21/11/2025
Component 6	v1.0.0	21/11/2025
Component 7	v1.0.0	28/11/2025
Component 8	v1.0.0	28/11/2025
Component 9	v1.0.0	28/11/2025
Component 10	v1.0.0	2/12/2025
Component 11	v1.0.0	2/12/2025
Component 12	v1.0.0	2/12/2025
Component 13	v1.0.0	5/12/2025
Component 14	v1.0.0	5/12/2025
Component 15	Not implemented	-
Component 16	Not implemented	-

Component 17	Not implemented	-
Component 18	Not implemented	-
Component 19	Not implemented	-
Component 20	Not implemented	-
Component 21	Not implemented	-
Component 22	Not implemented	-
Component 23	Not implemented	-
Component 24	Not implemented	-
Component 25	Not implemented	-
Component 26	Not implemented	-
Component 27	Not implemented	-

CM Responsibilities

For **version control**, the project will use the GitFlow model, which allows separating development work from stable code. This system will facilitate collaborative work and continuous integration through dedicated branches for development, new features, testing, releases, and urgent fixes.

Before each release, a **configuration audit** will be performed to verify that all components meet the defined functional and non-functional requirements, that versions are correctly labelled, and that there are no inconsistencies in the repository.

Planned Activities, Agenda and Resources

Configuration management will follow the same timeline established in the overall project plan. Iteration 0 will be dedicated to planning and defining requirements, while subsequent iterations will focus on the incremental development of the various system components, with configuration reviews and audits before each release.

Git will be used as the primary version control and collaboration tool. Git will allow for maintaining traceability of changes, recording code and documentation versions, and ensuring an organized repository structure in accordance with the GitFlow model.

CM Plan Maintenance

This Configuration Management Plan will be a living document, subject to continuous updates as changes occur in requirements or the project structure. Periodic reviews will be conducted at the end of each iteration to adapt the plan to new needs or technical adjustments identified during development.

Furthermore, the plan will serve as a reference at the start of each new phase, ensuring that approved changes are feasible, properly documented, and maintain system integrity.