ICES4HU
Configuration and Change Management Report

1 Introduction

Configuration management is a critical aspect of software development that ensures consistency and synchronicity across the project.   The settings and components of the software system are specified by the software configuration. Configurations need to be managed and tracked in order to continue product development and deployment, which is where configuration and change management come into play.

Controlling and monitoring changes to a changing product, like software, is a function of configuration management. It includes procedures like version control and auditing that make sure all configurations are accurate and current. Even so, change management focuses on the changes themselves and includes standards for when and where they should be executed.

This Configuration and Change Management Report's objective is to present a thorough strategy. In the following sections, we will discuss the specifications and key considerations that are essential to effective configuration and change management.

2 Purpose

This Configuration and Change Management Report's purpose is to offer recommendations and requirements for preserving a synchronized and consistent system configuration throughout the software development cycle.

- ■ The benefits of implementing a configuration and change management plan include:

    - Tracking proposals for changes

    - Reducing costs in terms of time and effort

    - Easy addition, removal, and modification of features

    - Detecting faults and bugs

    - Facilitating teamwork

    - Enabling the roll-back to a previous system state if required

■ This report aims to address the specific reasons for changes in our project and design revisions, such as:

- Errors detected by software testers

- Design revisions

- Implementation of new features that need to be integrated with the entire project without affecting other progress.

The report attempts to prevent redundant shifts and make sure that all developers are aware of the changes made by following the prescribed configuration routines and documenting any modifications performed. In the end, this report intends to assist the software system's maintainability and evolution in a regulated and effective manner.

3   Configuration and Change Management Specifications

Configuration and change management is an essential component of software development that guarantees the system's quality, dependability, and maintainability. We employ a proven set of tools and procedures in our project to efficiently manage the configuration and change management process.

Version Control System:

We use the distributed version control system (DVCS) Git and the remote private Git server provider GitHub as the project's primary tools and services. Together, Git and the GitHub service provide a solid and long-lasting platform for remotely and collaboratively maintaining our software system. We use GitHub to remotely maintain our system, and each team member has their own branches in that repository. These branches serve as development and testing branches before we merge those changes into our main branch.
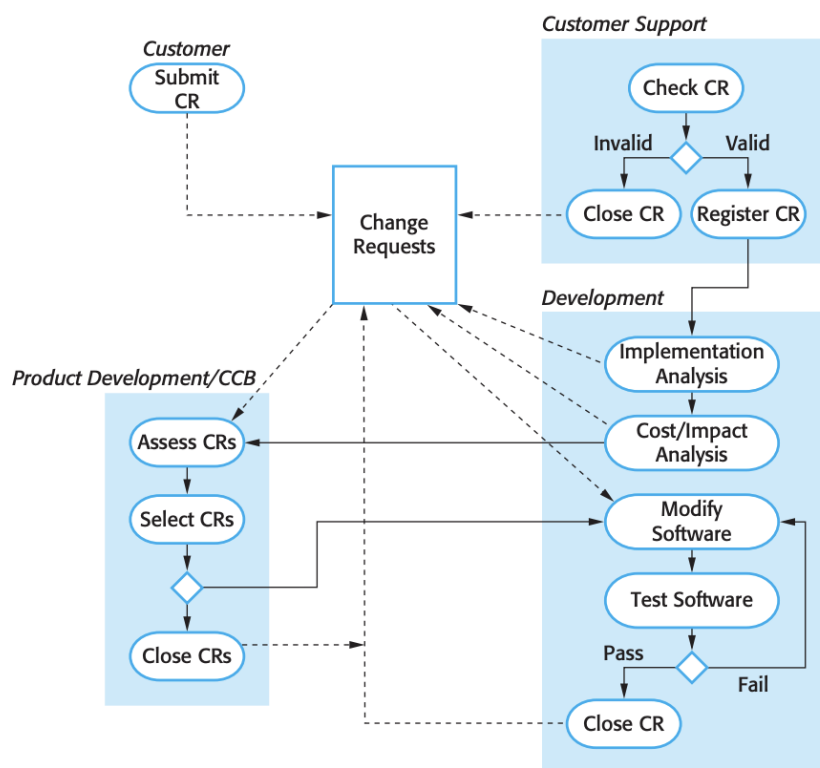
Change Request Process:

Each time a team member thinks of a potential addition or modification to any adjustable components, they commit this update to their branch. Members of the team examine and debate the suggested adjustment or addition. The project manager approves the pull request if there is agreement on the modification, and it is then incorporated into the main branch. It may also be denied or delayed as an alternative. If necessary, during this procedure, the documentation that is relevant to the altered component can also be updated.

Monitoring and Testing:

During the coding process, we perform independent installation and monitoring to ensure that all unplanned modifications are resolved. We check specific areas of the update during testing to ensure that it does not have a detrimental impact on other elements of the application.

The change management process

Change Management Plan:

Our change management plan consists of the following steps:

1. Identification of the change request: This step involves identifying the need for a change or addition to the system and submitting a request for review.
2. Review and analysis of the change request: The change request is reviewed and analyzed to determine its feasibility and impact on the system.
3. Approval or rejection of the change request: The change request is either approved, rejected, or postponed based on its feasibility and impact.
4. Implementation of the change request: If the change request is approved, it is implemented by making the necessary changes to the system.
5. Testing and verification of the change: The change is tested and verified to ensure that it does not have a detrimental impact on other elements of the application.
6. Documentation and communication: Documentation related to the change is updated, and communication is sent out to the relevant stakeholders informing them of the change.

To guarantee the quality, dependability, and maintainability of a software system, effective configuration and change management is essential. To manage the configuration and change management process successfully in our project, we combine a number of technologies and procedures, such as Git and GitHub, the change request procedure, monitoring and testing, and change management plan.

4 Key Considerations

Key considerations for implementing a configuration and change management plan include selecting the appropriate tools and technologies, establishing clear policies and procedures for handling changes, providing adequate training and support for team members, ensuring effective communication and collaboration among team members, and regularly reviewing and updating the plan to reflect changes in project requirements and objectives. It's crucial to establish a procedure for recording changes, which should include the justifications behind them, the people involved, and their effects on the system as a whole. By taking into account these factors, we can more effectively manage their software development projects and make sure that adjustments are made in a controlled and methodical way, ultimately resulting in more effective and successful project outcomes.

5 Distribution of Task

- Config change management written by Zübeyde Civelek.
- The whole team has checked the document.