LinkedHU_CENG
Configuration and Change Management Report

1 Introduction

During the life of a software system, it goes through some changes. These changes can take place during the development or use phases. A mistake that needs to be corrected, new requirements, new requests may cause. These changes should be planned and carried out in compliance with the software system.

Configuration management is the systematic management of the configuration components and resources of a software system in line with the system's processes, policies, and tools. Configuration management is critical for increasing the software system's performance, functional, and physical properties to the planned level and maintaining that level through all potential changes.

Change management refers to the process of analyzing a planned change and then planning that change as a result of that analysis. Aspects such as the contribution of the change to the system, the cost of the change, the necessity of the change, the methods considered, and efficiency are all discussed during this analysis process. In this process, the most logical and efficient change that can be applied is planned.

Configuration and change management are critical in ensuring that a change in a software system is managed and accurate. It's also crucial to keep track of how the team members approach the project and how well they get along with one another.

2 Purpose

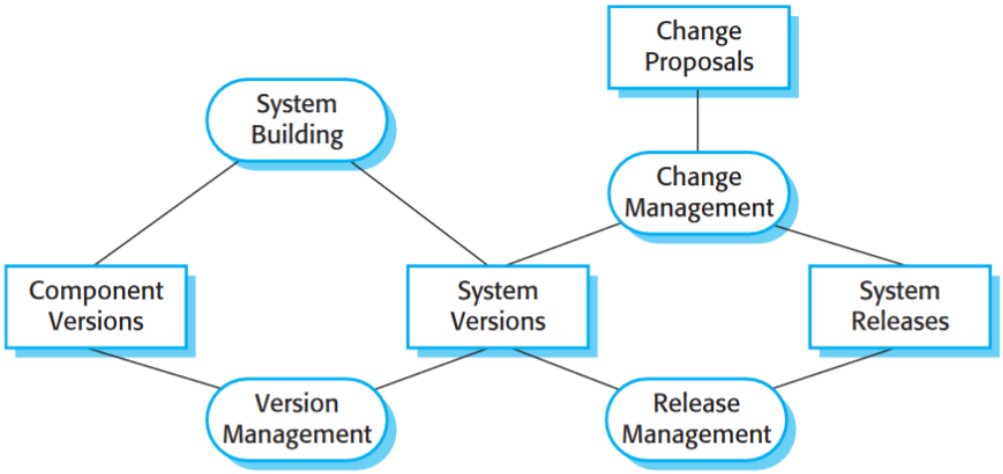
Configuration management has tasks such as defining configuration components such as network, server or computer resources, tagging them, ensuring their security, tracking changes, verifying and controlling configurations, and determining responsibilities. Change management has tasks such as determining the needs of changes and their effects on the system, planning and realizing the strategy of change.

The Git system will be used to ensure that this entire process is followed appropriately and that the system's versions are controlled. Git is a version management system that is free and open source. Changes will also be planned and reported using the Jira system. Jira is a bug tracking and project management system that can be accessed through Github. The main purpose of using these systems is to increase the efficiency of the cooperation in the project and to reduce the costs of mistakes that can be made.

Changes may be planned and exact times specified to each change task thanks to the Jira platform. This will allow team members to share work more efficiently. Team members will be able to work on the same project thanks to the Github platform, therefore there will be no conflict between the software packages used in the project. Changes can be done without affecting the main project by creating separate branches on Github, and the impacts of the changes on the system can be investigated.

As a result, this document is in charge of documenting the change's planning, implementation, and future effects, as well as guaranteeing team member compatibility and managing system versions.

3 Configuration and Change Management Specifications



Configuration management of a software system includes interrelated activities. These are Change management, Version management, System building, Release management.

1. Change Management:

   It is the activity of monitoring the plan, strategy and effects of the changes that can be made in the system due to reasons such as customer requests and system requirements.

   The versions of the project and the release times of the versions were determined. Strategies and applicability of the changes to be made were planned. Tests were prepared to ensure the accuracy of the changes, and the environment was created to monitor their compatibility with the system. Considered for the effects of changes on the system, and errors are reduced.

2. Version management:

   It is the activity in charge of maintaining the system's integrity and development, as well as providing version control and assuring developer harmony. This activity will be synchronized with this process thanks to the Github platform. A branch will be created for each task, and team members will be able to work with their own branches from there. A controlled working environment was created without damaging the system's main project.

3. System building:

   It is the process of assembling a program's components, data, and libraries into a unified whole and harmonizing it. Project components and changes to the project are time stamped on the Jira platform. The development process continues by estimating all possible errors and risks.

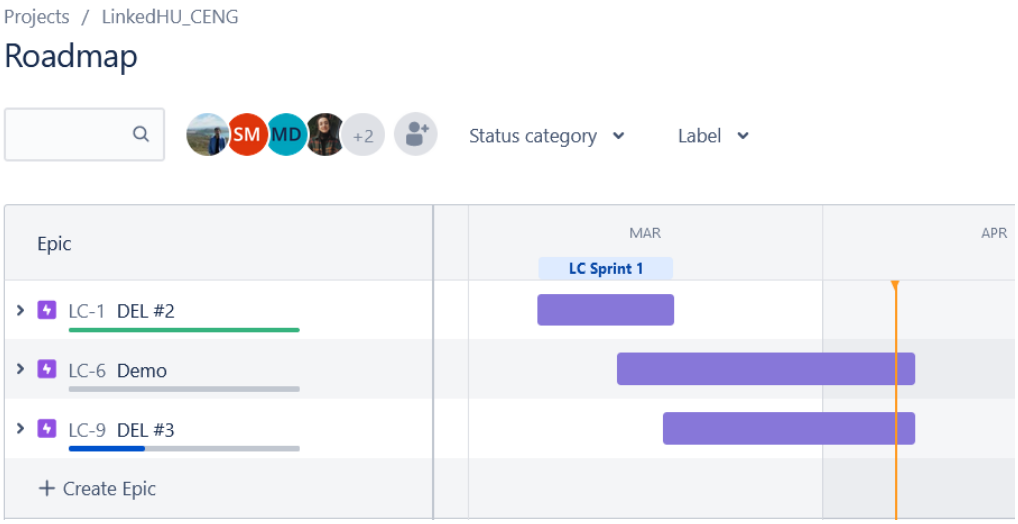
4. Release management:

   It includes tracking the lifetime and adaptation of versions of the software at the time of development and customer use. Correct choices were made so that the version of the software used in the development process should be "long term supported". .NET 6.0 with the “LTS” label was preferred in the main software language.

4 Key Considerations

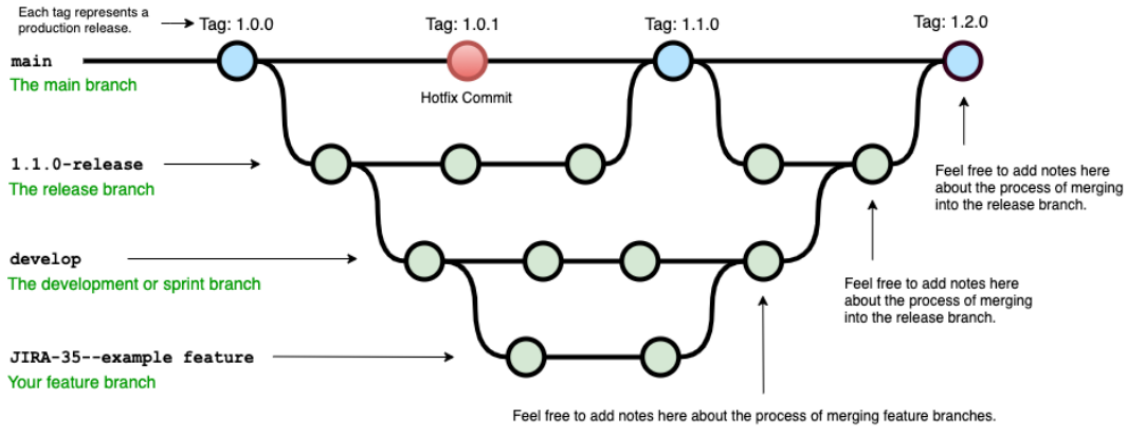
The development process is difficult and error-prone. Some rules must be established in order to better build this system. Jira and Github platforms are utilized in this system to solve this problem and organize the development process. All tasks that must be completed on the Jira platform must be entered on a daily basis, with members and times assigned. Thanks to Github integration, these activities provide a convenient setting for team members to collaborate.

Below is a sample section of the Jira platform.



On the Jira platform, it shows the stage of a task, when it will be completed, and by whom and by whom. With this planning, it is possible to take quick steps in the development process.

Below is an example of the map to follow on the github platform.



Some rules have been determined for the use of the Github platform in the system.

1. Create Branch

   As seen above, development will proceed with a specific map rule. It will have a Main branch. This branch will have the latest changes of the enhancements. It is the branch of our main project. There will be one develop branch. New feature branches will be opened on this branch. The names of these branches will be opened with the task and team member name details. Tasks of team members can be tracked. In this way, developments will be achieved without affecting the main branch, and team members will be able to work efficiently.

2. Add Commits

   During the development stages, operations such as adding, deleting or updating can be done. These operations are committed to branches. These operations can be recorded and tracked on the project. Special messages are written to the operations in all of these operations so that others can understand the changes. Benefits include defining the project's direction and preventing mistakes and confusion as a result of these messages.

3. Open a Pull Request

   After a development has been completed on the computer, the developer can open a pull request. An efficient step should be taken for the reviewers to check by adding a detailed comment with this request.

4. Review and Control Code

   After one of the team members completes the development, they send a pull request and a secondary check is done by the reviewers. Comments are made at the controls, and the suitability of the request is evaluated. It was decided by the team members that there should be more than one reviewer and that the review should be done faster for efficiency.

5. Merge

   After all controls are checked, the request is approved, and the merging process takes place. With this merge process, the project goes one step further. It is possible to reach the pre-merge project in a future error.

6. Deployment

   In order to avoid confusion in the project deploy phase, a release branch will be created and worked on. After the development on the release branch is completed, the merge authority for the deploy phase will be the responsibility of a single person. This is a step taken to minimize possible risks.

If the whole process is briefly summarized, a developer will work on the develop branch. The developer will create sub-branches (feature branch) for the things that she/he is in charge of. The developer will be able to merge into the develop branch once she/he's finished with these sub-branches. After the tests in the develop branch are completed, a pull request is sent to merge with the main branch. Reviewers will examine this request, and if it is found to be inappropriate, the issues will be fixed. The merge procedure will be carried out once the request has been accepted. During these processes, the developer will execute various tests in order to decrease all possible risks. When the main branch development is finished, a deployment can be done. This is a process that the team will continue to work on together and in communication throughout.

Distribution of Tasks

This document was prepared primarily in collaboration with Software Configuration Manager (Murat Çelik) and Software Analyst (Humeyra Uçar), with contributions from the entire team.