Hacettepe University
Department of Computer Engineering
BBM478 - Software Engineering Laboratory

Configuration and Change Management Report

Group One

Ahmed Şamil BÜLBÜL (21426749, Project Manager)
Halil İbrahim ŞENER (21328447, Software Architect)
Naciye GÜZEL (21580841, Software Developer)

1. Purpose

This document describes the strategy of handling issues, changes, decisions, constraints, and other aspects of the system in the design and implementation. Software change management is the process of selecting which changes to encourage, which to allow, and which to prevent, according to project criteria such as schedule and cost.

2. Configuration and Change Management Specifications

Configuration management is a systems engineering process for creating a basis for product and keeping performance, functionality of product in that basis with requirements, design throughout product's life. Configuration management helps to check and confirms that system performs as intended. Configuration management has got 4 activities: Change management, Version management, System building, Release management.

Changes are inevitable in development processes. Systems keep evolving. However systems cannot accept every change. Change management deals with keeping track of requests for changes to the software from customers and developers, working out the costs and impact of making these changes, and deciding if and when the changes should be implemented.

Our main approach of handling configuration and change management will be done by using GitHub and TravisCI.

GitHub provides Version management by keeping records of changes files and file systems. With GitHub releases, we can also handle Release management.

Continuous integration is merging different works in order to get the product. By using continuous integration, errors and issues will be detected before they are integrated into the system which decreases the cost to check them after integration. The cost of integration decreases because of automation. Automation makes better use of resources and increases in quality of the product. It provides some standards. In addition to these, it provides faster integration. In Library Book Loan System, the project will use Travis CI. Travis CI provides a powerful integration tool. The way Travis CI works is, in every PR (Pull Request), GitHub sends a message to Travis CI that build is mergeable. Travis CI builds the project then it updates the PR.

Configuration management will be held in GitHub. Our repository holds the source code of the project and documentations. Our development approach is:

- Branch protection is applied. No direct commits to master branch.
- When there is a problem or issue that needs to be changed, one creates an issue that specifies problems clearly.
- Development of a specific feature or an issue will be done by creating a branch that clearly shows what is to be done.
- In development branches, one or more project members can work.
- Multiple branches can be created at the same time.
- If development is done in a branch then pull request should be made.
- In pull request, project members should review code and give "peer review", give comments, make changes or approve changes.
- If a branch passes peer review and gets approved, then GitHub triggers Travis CI.
- Merge operation is done.

Software configuration management is about tracking and controlling changes in the software, and it is a part of the larger cross-disciplinary field of configuration management. Revision control and establishment of

baseline are practices of software configuration management. Git is a Source Code Management (SCM) system. It allows revision and tracking and controlling changes. So by using GitHub and Travis CI, we are managing our main configurations and changes.

GitHub improved maintainability of the source code and reduced risk and liability. By providing such things it reduced the cost of processes in general.

Maven is a tool to provide a better build system. It is a build automation tool used primarily for Java. Project dependency problems are handled here. It also describes how software is built. With Maven, dependencies directories, plugins, build process are handled automatically. It helps to make changes on project easier and allow component changes in a better way.

To demonstrate change management, after accepting to develop a feature or to overcome the problem activity is like below: