Software Engineering Laboratory
Online Book Store System

Group Name : DEVAMKEE
- Muhammed Said Kaya 21627428
- Muhammet Eren Taşdemir 21727743
- İhsan Kürşad Ünal 21627758
- Emre Hancı 21604552
- Ali Kayadibi 21727432

Prudentia Bookstore
Configuration and Change Management Report

1 Introduction

Configuration management (CM) is considered with policies, processes, and tools for managing software systems. Need to manage evolving systems because it is easy to lose track of what changes and component versions have been incorporated into each system version.

The configuration management of a software system product involves four closely related activities:

1. *Change management* This involves keeping track of requests for changes to the software from customers and developers, working out the cost impact of making these changes, and deciding if and when the changes should be implemented.

2. *Version management* This involves keeping track of the multiple versions of system components and ensuring that changes made to components by different developers do not interfere with each other.

3. *System building* This is the process of assembling program components, data, and libraries, and then compiling and linking these to create an executable system.

4. *Release management* This involves preparing software for external release and keeping track of the system version that has been released for customer use.

2 Purpose

The overall objective of a Configuration Management (CM) Plan is to document and inform project stakeholders about CM with the project, what CM tools will be used, and how they will be applied by the project to promote success. The "Prudentia Bookstore" CM Plan defines the project's structure and methods for

- Controlling modifications and releases of CIs
- Reporting and recording status of CIs and any requested modifications
- Ensuring completeness, consistency, and correctness of CIs
- Controlling storage, handling, and delivery of the CIs

The intended audience of the CM Plan is the project manager, project team, project sponsor and any senior leaders whose support is needed to carry out communication plans.

3 Configuration and Change Management Specifications

Configuration management is handled by Git-Hub. Changes and updates can be tracked from Git-Hub repository. In Git-Hub master branch is our main-line and other branches are base-lines. Code-lines in our local devices. Because all developers have their own SCI. SCI is shared to the configuration manager by other communication providers like e-mail or Discord.

Change requests made by using a shared "Change Request Log" file in Google-Documents.

Change logs so far :

Change Request Log					
Project Name	Prudentia Bookstore				
Request No.	Name Of Request	Owner	Last Updated	Status	Next Update Due
1	Change of database attribute name	PM,CM	14.04.2020	Accepted	N/A
2	Database charset change	PM,CM	16.04.2020	Accepted	N/A
3	Database functions move to java	PM	18.04.2020	Accepted	N/A
4	Change of frontend template platform change	SA	20.04.2020	Rejected	-
5	Change of software run platform (from local to heroku)	PM	22.04.2020	Accepted	N/A

Request Template

Request Name	SHORT SPECIFICATION OF THE REQUEST
Request Owner	NAME AND PLACE IN PROJECT
Request Reason	DETAILED SPECIFICATION OF THE REQUEST
Estimated time to change	ESTIMATED TIME TO DONE REQUEST (IN HOUR)

4 Key Considerations

We are using only one version at the same time. So the master branch in Git-Hub is our last version. For some reason all developers have their own branch with their own number. Commits representing actual works. This project is not too large so we passed some levels between mainline and code-line. Heroku repository is always updated but Git-Hub may not be updated. Periodically Git-Hub updated, because Git-Hub branches last two levels of the management up to these layers; updates may take a long time. But Heroku repository will be always updated.