BoomBook Bookstore
Configuration and Change Management Report

1 Introduction

Configuration management (CM) is the management of the system changes to software product. It is also concerned with the policies, processes, and tools for managing changing software systems. CM has become particularly important within the industry because if we do not have effective configuration management procedures in place, we may waste effort modifying the wrong version of a system.

Change Management plan is a generic plan that guides the Project Manager in terms of making any kind of change on project. It involves 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 when the changes should be implemented. It is also important within the industry because of the cost benefits.

2 Purpose

The purpose of this report is discussing some activities of configuration management and change management. We have already used GitHub while developing the project and it will be used throughout the project. Some operations are performed while working with GitHub such as branch creation, commit, pull request. In this report, a few examples of these activities from the project will be discussed.

3 Configuration and Change Management Specifications

Managing a change in software is accomplished by understanding the origin of possible variables. Software changes are caused by many different reasons and we follow some decision points to manage them. We used GitHub to facilitate this management because it is a project management and version control system and it has helped us to track the changes made in the code.

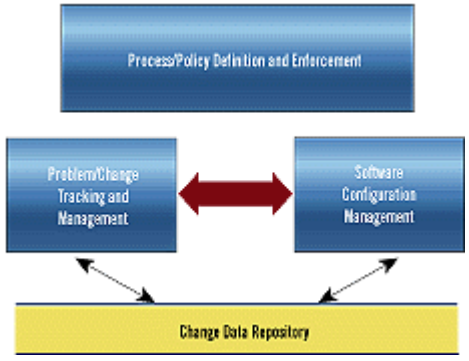

Figure1: Software Change Management

When a member of the project wants to change some aspects of the project, or when a new idea comes from them, he will explain the ideas with logical and understandable reasons and the change request will be evaluated by all members in the project. If the ideas are approved, we assign a priority to ensure appropriate resources and urgency are applied. The required changes are documented, and the old documents are updated if necessary. When all updated documents and changes are completed within the branch, we informed that we have pushed the changes to the main repository thanks to the pull request. Other members of the project can accept or reject the pull request. All members should analyze how long this change will take because this analysis can convince us to revise our priorities. If we are all agree with the idea, the pull request will be accepted.

For example, at the beginning of our project, we decided to use angular.js for the front-end of the application but as the project progresses, we switched to bootstrap which is easy to use for programmer. We also included frameworks such as jQuery where JavaScript is required.

When the change is folded into the planned development effort, a change request is completed. During coding, we conduct separate implementation and testing to verify the resolution for any unplanned changes. After testing, we still review some aspects of the change to ensure it will not negatively affect other parts of the application.

While we are following the decision points, we can decide that the change request should be rejected or postponed. In this situation, we keep the change request and all associated documentation so that we can do recovery operation easily.

While we are advancing the project, we may not have enough time for full analysis and decisions if a problem is big. In this situation, our process of change management will include an emergency path that is separated from our decision points. We must focus this process on an immediate resolution that eliminates shutdown. We can update the change request to document the quick fix.

Since our project is team project where several developers are working at the same time on a software system, configuration management (CM) is necessary. The use of a configuration management system will ensure us to have access to information about a system that do not interfere with each other’s work. Throughout the project, we will also follow some activities to achieve configuration management of the project.

Figure2: Software Change Management

Configuration management is handled by GitHub. Our main repository holds the source code of the project and documentations. New documents and changes of code modules is tracked here. Throughout the project, each project member will commit the changes of the code that they have developed on their branches and sent a pull request to combine it with the central repository when the job is done. Since we apply version management activity, these changes do not interfere with each other.

While we are following these activities configuration of the control process will ensure us that version of systems and components are maintained so that all changes are managed, and all versions of components are identified for the project.

4 Key Considerations

Change and configuration management lets us to maintain software evolution and provides the process improvement. The data that we collect under a consistent process will support estimating and planning, reducing risk, and making development more predictable. Understanding the origins of possible change variables that is discussed in part 3, the decision points, and the roles in the decision process will ensure us to control and manage the project, rather than just watch the software when it changes.