DEPARTMENT OF COMPUTER ENGINEERING

BBM384 Software Engineering Laboratory

Online Book Management System
Configuration and Change Management Report

BOOKIFY

Ismet Okatar 21727542
Atakan Ayyıldız 21526681
Göktuğ CANDEMİR 21627064
Ege ÇINAR 21627136
Muhammed Aydoğan 21726952

Configuration and Change Management Report

1 Introduction

We will make changes to our original plans during the software development period. The Software Configuration and Change Management Plan has been developed so that we can identify the change, control the change, ensure that the plan is implemented correctly and report the change to others.

2 Purpose

The objectives of the Configuration and Change Management process are to:

• Respond to the customer's changing business requirements while maximizing value and reducing incidents, disruption and re-work.

• Ensure that changes are recorded and evaluated, and that authorized changes are prioritized, planned, tested, implemented, documented and reviewed in a controlled manner.

• Ensure that changes to configuration items are recorded in the Configuration Management System.

• Optimize overall business risk.
  - It is often correct to minimize business risk , but sometimes it is appropriate to knowingly accept a risk because of the potential benefit.

• Changes should be managed to:
  - Optimize risk exposure (supporting the risk profile required by the business).
  - Minimize the severity of any impact and disruption.
  - Achieve success at the first attempt.
  - Ensure that all stakeholders receive appropriate and timely communication about change so that they are aware and ready to adopt and support the change.

3 Configuration and Change Management Specifications

Change Management documents how changes will be monitored and controlled. The plan defines the process for managing change on the project.

A Configuration Management documents how configuration management will be performed. It defines those items that are configurable, those that require formal change control, and the process for controlling changes to such items.

3.1 Configuration Management:

Software Configuration Management is a process to systematically manage, organize, and control the changes in the documents, codes, and other entities during the Software Development Life Cycle.It is abbreviated as the SCM process in software engineering. The primary goal is to increase productivity with minimal mistakes.

The primary reasons for Implementing

Software Configuration Management System are**:**

- There are multiple people working on software which is continually updating
- It may be a case where multiple version, branches, authors are involved in a software project, and the team is geographically distributed and works concurrently
- Changes in user requirement, policy, budget, schedule need to be accommodated.
- Software should able to run on various machines and Operating Systems
- Helps to develop coordination among stakeholders
- SCM process is also beneficial to control the costs involved in making changes to a system

Any change in the software configuration Items will affect the final product. Therefore, changes to configuration items need to be controlled and managed.

3.2 Change Management:

There will be always further changes after the first specification of the software which will be build.

Change management is a systematic approach to dealing with the transition or transformation of an organization's goals, processes or technologies. The purpose of change management is to implement strategies for effecting change, controlling change and helping people to adapt to change. Such strategies include having a structured procedure for requesting a change, as well as mechanisms for responding to requests and following them up.

To be effective, the change management process must take into consideration how an adjustment or replacement will impact processes, systems, and employees within the organization. There must be a process for planning and testing change, a process for communicating change, a process for scheduling and implementing change, a process for documenting change and a process for evaluating its effects. Documentation is a critical component of change management, not only to maintain an audit trail should a rollback become necessary but also to ensure compliance with internal and external controls, including regulatory compliance.

Change management can be used to manage many types of organizational change. The three most common types are:

1. **Developmental change -** Any organizational change that improves on previously established processes and procedures.
2. **Transitional change -** Change that moves an organization away from its current

state to a new state in order to solve a problem, such as mergers and acquisitions and automation.

3. **Transformational change -** Change that radically and fundamentally alters the culture and operation of an organization. In transformational change, the end result may not be known. For example, a company may pursue entirely different products or markets.

Change Request Form Template			
Project Name	**Bookify**		
Requested By	Project Manager	Date	16.04.2020
Request No	3	Name of Request	Add Staff Log in Page
Change Description	Login page for staff who are different from the admin and customers.		
Change Reason	This is a new requirement from project manager.		
Impact of change	Extra editing and extra code costs are required for backend and frontend.		
Proposed Action	It will not be difficult to add this difference since we use the mvc model, but this request is not critical for the project at the moment.		
Status	In review	Approved	Rejected
			✓
Approval Date	N/A		
Approved By	N/A		

4 Key Considerations

1. **Version Control:** By using a version control it can be prevented potential code losses and disruptions. And the written codes can be reviewed more easily and in a more reliable way.
2. **Baseline and release information:** Every version released there will be also a baseline version which can be deployed at any time. Developers must branch to protect the main stream of development, they must check in code that does not break the build and the process of declaring a baseline must ensure integrity as future development builds on the baseline.
3. **Audits & Review:** Reviewing the process and the merge requests to manage the configurations and ensure that the code is correct and clean.
4. **Documented Process:** All the process will be made by following the previously specified documentation. And each baseline version will have documentation for the users which will be using our system.
5. **Build, Integrate and Deploy Scripts:** By end of each merge request there will always be a working product. The potential errors will be easily prevented by our error catching methods.