Software Configuration Management Plan

Game of Codes—Brogrammers 2.0.2 (Last Updated 4/3/18 by Vijayanta Jain)

# 1. Introduction

We are developing a web-based game that is a puzzle platformer which teaches players the fundamental logic behind programming. Our design encourages problem solving and critical thinking for the player to progress. The game will have different levels each being more complicated than the previous one and utilizing the concepts from the previous lesson. Our goal is to teach fundamental programming concepts to individuals while playing a fun puzzle platformer.

Following are the configuration items (CI) for our project with identifier -

* **Platform Document** (*Brogrammers 2.0.2\_Deliverable\_3\_Platform*)
* **Domain Model and Detailed Design Document** (*Brogrammers 2.0.2\_Deliverable\_3\_Domain-Design*)
* **Software Requirement Specification Document** (*Brogrammers 2.0.2\_Deliverable\_3\_SRS*)
* **Use Case Model Document** (*Brogrammers 2.0.2\_Deliverable\_3\_UseCaseDocument*)
* **GRL and UCM Models** (*Brogrammers 2.0.2\_Deliverable\_3\_URN, Brogrammers 2.0.2\_Deliverable\_3\_URN\_GRL.jucm, Brogrammers 2.0.2\_Deliverable\_3\_UCM.jucm*)
* **Sprint Review** (*Brogrammers 2.0.2\_Deliverable\_3\_SprintReview*)
* **Configuration Management Plan** (*Brogrammers 2.0.2\_Deliverable\_3\_Configuration\_Management\_Plan*)
* **Architecture Design and Design Patterns** (*Brogrammers 2.0.2\_Deliverable\_3\_SoftwareArchitecture*)
* **Test Plan Documents** (*Brogrammers 2.0.2\_Deliverable\_3\_Test\_Plan*)
* **Daily Scrum Report** (Brogrammers 2.0.2\_Deliverable\_3\_Daily Scrum Meetings)
* **Sprint Backlog** (Brogrammers 2.0.2\_Deliverable\_3\_SprintBacklog\_3)
* **Product Backlog (**Brogrammers 2.0.2\_Deliverable\_3\_ProductBacklog)

Many sub-sections in the document have been omitted because they are not applicable for our current scenario of development and SCM process.

# 2. SCM Management

## 2.1 Organization

Upon using our SCM plan, there will be three members assigned to the task of carrying out the SCM plan. One shall be the Project Manager, one shall be the Debugger, and one shall be the Tester.

## 2.2 Responsibilities

* **Product Manager**

Following are the roles of the product manager:

* Manage the product backlog.
* Order items in the product backlog.
* Make product backlog visible to all.
* Determine what is “done” and acceptable in sprint.
* Cancel a sprint if the sprint goal becomes obsolete.
* **Scrum Master**

Following are the roles of scrum master:

* Make sure the development team is practicing Scrum properly.
* Facilitate the product owner’s and development team’s work.
* Interface with external entities.
* **Developer**

Following are the roles for the developer:

* Determine the work to be done in a sprint.
* Work with the product owner to determine what “done” is.
* Implement the user stories in the sprint.
* Manage the sprint backlog.

## 2.3 Applicable Policies, Directive and Procedures

N/A

# 3. SCM Activities

## 3.1 Configuration Identification

Configuration Items -

* **Platform Document** (*Brogrammers 2.0.2\_Deliverable\_3\_Platform*)
* **Domain Model and Detailed Design Document** (*Brogrammers 2.0.2\_Deliverable\_3\_Domain-Design*)
* **Software Requirement Specification Document** (*Brogrammers 2.0.2\_Deliverable\_3\_SRS*)
* **Use Case Model Document** (*Brogrammers 2.0.2\_Deliverable\_3\_UseCaseDocument*)
* **GRL and UCM Models** (*Brogrammers 2.0.2\_Deliverable\_3\_URN, Brogrammers 2.0.2\_Deliverable\_3\_URN\_GRL.jucm, Brogrammers 2.0.2\_Deliverable\_3\_UCM.jucm*)
* **Sprint Review** (*Brogrammers 2.0.2\_Deliverable\_3\_SprintReview*)
* **Configuration Management Plan** (*Brogrammers 2.0.2\_Deliverable\_3\_Configuration\_Management\_Plan*)
* **Architecture Design and Design Patterns** (*Brogrammers 2.0.2\_Deliverable\_3\_SoftwareArchitecture*)
* **Test Plan Documents** (*Brogrammers 2.0.2\_Deliverable\_3\_Test\_Plan*)
* **Daily Scrum Report** (Brogrammers 2.0.2\_Deliverable\_3\_Daily Scrum Meetings)
* **Sprint Backlog** (Brogrammers 2.0.2\_Deliverable\_3\_SprintBacklog\_3)
* **Product Backlog (**Brogrammers 2.0.2\_Deliverable\_3\_ProductBacklog)

## 3.2 Configuration Control

**Requesting Changes**

Currently, only the developers can suggest changes. The protocol to suggest an issue is through Github’s Issue tab. They can create a new issue which the other members of the team can comment on and discuss. Finally, based on the votes on the issue, the product owner decides if the change should be added to the product backlog or not.

**Evaluating Changes**

Changes are evaluated through Pull Requests (PR) in Github. After a developer implements a change, he creates a PR on the repository webpage. Those changes are then evaluated by at least two other developers who evaluate the performance of the PR and the provide feedback. After the feedback is implemented, the changes are merged to the main branch.

The main criteria to evaluate changes are:

* Effectiveness of the change
* Compatibility with the current baseline

**Accepting/Rejecting Changes**

Based on the criterias mentioned above we accept or reject changes. It is the product owner’s authority to accept or reject changes made.

## 3.3 Configuration Status Accounting

N/A

## 3.4 Configuration Evaluation and Reviews

N/A

## 3.5 Interface Control

Any changes made to the CI are informed to the Product Owner and the Scrum Master which accordingly discuss with the development team to assign the task of updating the interfacing item. It is then added to the Product Backlog

## 3.6 Subcontractor/Vendor Control

N/A

## 3.7 Release Management and Delivery

The software is tested and updated at the end of every sprint. The build of the software is not necessary as the software is a web-application.

# 4. SCM Schedule

Every item in the SCM schedule is completed within a sprint - starting from the Version Management till the Release Management of the software. Before the start of the sprint, every SCM activity is discussed at the sprint meeting and added to the sprint backlog; Based on thorough discussion every story gets assigned someone. All the major activities of the SCM are achieved in the serial order of version control followed by build control followed by release management followed by change management. The changes are discussed and reviewed (see section 3.2 for details) and added to the product backlog at the end of the sprint review.

**Baseline Establishment**

The baseline of the CI is established at the end of every sprint and new baseline is started from the previous establishment. At the beginning of every sprint, target for a new baseline is established.

**Implementation of changes**

Based on the new target baseline, the changes are broken down into tasks which are then completed by the assignees. Each completion leads towards achieving a new-baseline of CI

**Timeline**

The CI is audited at the end of the sprint at the sprint review meeting where changes made are audited and the new baseline is established

# 5. SCM Resources

Following are the resources being used for the SCM

1. Software Tools
   * IDE(s)/Code Editors - VS Code, Notepad++, Netbeans IDE are the main tools being used to edit and develop code.
   * Version Control - We use GitHub and git as our version control softwares. GitHub is used to maintain the central repository and git is used to manage our local version control. We also GitHub for change management.
   * Languages/Libraries - We are currently using JSP, Javascript, HTML/CSS to develop our game
2. Techniques
   * We are using agile software management to develop our code. We are using scrum for the process of developing the software and Kanban board to manage the workflow.
3. Training
   * Training of all the personnel is carried out during the classroom as well as during the development phase. There is no formal training process, everyone learns independently as well as collaboratively. New knowledge learned is also added to the Wiki!
4. Personnel
   * According to the scrum guidelines, the personnel can be divided into three individuals and groups. The Scrum Master ensures that all the team is managing the scrum effectively. The Product owner manages the tasks and the sprint goals of the software. The development team does the development of the software. Because of lack of personnel, every team member is also part of the development team.

# 6. SCM Plan Maintenance

**Maintainer**

The SCMP is maintained by the Scrum Master. Based on the discussions done at the Sprint Review, the Scrum master accordingly makes changes and updates the plan.

**Updates**

All the updates to the plan are performed as and when required. All the changes made are done during the sprint.

**Evaluation & Approval**

The changes are mainly evaluated by the product owner and select individuals from the development team that review the changes made by the scrum master. The metric for evaluating the documents are:

* Accuracy - Ensuring that the changes made are accurately described.
* Consistency - There are no conflicts with the document itself or the process (i.e. accurate).
* Brevity - The changes made are short but descriptive. The purpose of the document is to inform an individual rather than hinder one from learning.

The changes made are then communicated at the sprint review which are then reviewed and updated accordingly.

**History of Changes**

|  |  |  |
| --- | --- | --- |
| Name of Reviser | Date | Description of Change |
| Felix Perez | 3/9/18 | Initial Document Creation |
| Vijayanta Jain | 4/4/18 | Adding content to document |
|  |  |  |

*This SCM Plan conforms with the requirements of IEEE Std 828-2005.*