**Table of Contents**

1. **Introduction………………………………………………………………………………………..**
   1. **Purpose**
   2. **Scope**
   3. **Key Terms**
   4. **References**
2. **SCM management………………………………………………………………………………….**

**2.1 Organization**

**2.2 Responsibilities**

**2.3 Applicable Policies, Directives, and Procedures**

**3. SCM Activities……………………………………………………………………………………...**

**3.1 Configuration Identification**

**3.2 Configuration Control**

**3.3 Configuration status accounting**

**3.4 Configuration evaluation and reviews**

**3.5 Interface control**

**3.6 Subcontractor/vendor control**

* 1. **Release Management and Delivery**

**4. SCM Schedules……………………………………………………………………………………...**

**4.1 Sequence and coordination of SCM activities**

**4.2 Relationship of key SCM activities to project milestones or events**

**5. SCM Resources……………………………………………………………………………………...**

**5.1 Identify environment, infrastructure, software tools, techniques, equipment, personnel, and training**

**5.2 Key factors for infrastructure**

**6. SCM Plan Maintenance…………………………………………………………………………….**

**6.1 Who is responsible for monitoring the plan?**

**6.2 How frequently updates are to be performed?**

**6.3 How changes to the plan are to be evaluated and approved?**

**6.4 How changes to the plan are to be made and communicated?**

**6.5 History of changes made to the plan**

# Introduction

# Purpose

This application provides a place for both tutors and tutees to connect, interact, and work together in an organized manor.

# Scope

* Tutor Application
* Tutor and Tutee account creation
* Profiles
* Payment
* Tutor resources
* Help resource

# Key Terms

SCM – Software Configuration Management

JavaScript Component – application feature contained in a JavaScript class

# References

Create-React-App <https://create-react-app.dev/>

Github <https://github.com/>

Zenhub <https://www.zenhub.com>

# SCM Management

# Organization

# *Organizational context (technical and managerial) within which the configuration management activities are implemented.*

* Product Owner
* Scrum Master
* Development Team

# Responsibilities

# *Each of these jobs are changed based on the time for each deliverable. Every time there is a new Deliverable the jobs are reassigned.*

# *Product owner*

* Manages the product backlog.
* Orders the items in the product backlog.
* Makes the product backlog visible to all.
* Determines what is “done” and is acceptable in the sprint.
* Creates required documents.
* May cancel a sprint if the sprint goal becomes obsolete.

# *Scrum Master*

* Makes sure the development team is practicing in Scrum properly.
* Facilitates the product owner’s and development team’s work.
* Creates required documents.
* Interfaces with external entities

# *Development Team*

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

# Applicable Policies, Directives, and Procedures

# *External constraints placed on the SCMP.*

Sprints – 2-week interaction implementing a set of prioritized user stories.

Sprint Planning – Sprint goal given by Product Owner. Includes what can be done in each sprint.

Weekly Scrum – meetings between the roles above to communicate, show progress, and plan.

# SCM Activities

# Configuration Identification

# *Identify configuration items (events, items, procedures)*

*3.1.1 Software*

Software components are contained in tutor-app folder within the GitHub repository. Various application features are implemented as JavaScript files within the scr folder that is located inside of tutor-app. Other configuration items necessary to run the program and created by create-react-app are located within tutor-app. More information about create-react-app can be found at <https://create-react-app.dev/>. Software files are created and edited then pushed onto the Github repository.

*3.1.2 Documentation*

Each document follows a strict naming convention *TutorGroup\_Deliverable\_i\_DocumentName.* The “i” signifies which deliverable the documents are associated with. The “i” would be replaced with a number (ex. 1) signifying the deliverable.Documentation files are created then pushed onto the repository. The files creation is based upon whose role it is to create the file based upon the responsibilities of each job/role.

# Configuration Control

# *Requesting changes*

Requests for changes are put onto the Kanban board where the request is visible to all roles (Product Owner, Scrum master, and Development Team).

# *Evaluating changes*

Changes are reviewed by jobs relevant to the change.

# *Approving or disapproving changes*

Meetings between the associated roles allow communication to be made to decide if the changes should be approved.

# *Implementing changes*

Kanban board is updated with the changed being enacted and the associated role with the change performs the change.

# Configuration Status Accounting

# *Metrics to be tracked and reported and type of report.*

Each member contributions are tracked based upon which deliverable is being worked on. Once a new deliverable approaches the contributions are rerecorded.

# *Storage and access control of status data.*

Files are stored on a repository available to all members through GitHub.

# Configuration Evaluation and Reviews

Configuration evaluation and review is done by Sepideh Ghanavati, and Sanonda Datta Gupta. Once each deliverable has been completed, they are submitted to the two people mentioned before who review our work and give a grade to signify if the work is acceptable. After grading, the team fixes any mistakes given by the two people assessing our deliverable.

# *At minimum an audit on a CI prior to its release.*

# *Defines objective, schedule, procedures, participants, approval criteria etc.*

# Interface Control

# *Coordination of changes to CIs with changes to interfacing items outside of the scope of the Plan.*

There are currently no changes to interfacing items outside of the scope of the Plan.

# Subcontractor/Vendor Control

# *Incorporation of items developed outside the project environment into the project CIs.*

There are currently no incorporation of items developed outside the project environment.

# Release Management and Delivery

# *Description of the formal control of build, release and delivery of software products.*

The application is available through a Github repository. Once downloaded, the application can be run in terminal with “npm start”. There is a build file available for windows user’s. The application will be available from a url once finished for all users.

# SCM Schedules

# Sequence and coordination of SCM activities

Code components are completed in line with sprint expectations. Progress is recorded on Github and issue’s arising in software development are reported on the Kanban.

# Relationship of key SCM activities to project milestones or events, such as:

# *Establishment of configuration baseline*

Configuration of the baseline occurs throughout the development process. New component versions are added to the baseline as they are pushed to the Github repository.

# *Implementation of change control procedures*

In the event of a change to software, this change is immediately pushed to the Github repository once the current working session on it is complete. To ensure no previous commits are overwritten, developers always work with an up-to-date repository.

# *Start and completion dates for a configuration audit*

# Schedule either as absolute dates, relative to SCM or project milestones or as sequence of events.

# Graphical representations can be used here.

# SCM Resources

# Identifies environment, infrastructure, software tools, techniques, equipment, personnel, and training.

The application environment and infrastructure is a that of create-react-app, more information can be found at <https://create-react-app.dev/> . Software and development tools include code editors such as Visual Studio Code, and version control management system such as Github (https://github.com/).. Development issues are recorded on KanBan (https://www.zenhub.com).

Contributing and advising personnel include Chase Pisone, Calvin Mueller, Hunter McDaniels-Rosseiter, Klei Bendo, Hannah Yelle, Sepideh Ghanavati, Sanonda Datta Gupta, and Stephan Kaplan. React-bootstrap information and resources are used in software development as well.

# Key factors for infrastructure:

# *Functionality, performance, safety, security, availability, space requirements, equipment, costs, and time constraints.*

This application was made over a period of 8 weeks. The application has no costs and is free for all users. Security is available through the creation of a username and password and being a Umaine student.

# Identify which tools are used in which activity.

# SCM Plan Maintenance

# Who is responsible for monitoring the plan?

The Product Owner oversees keeping track of who does what in each deliverable. During each scrum meeting work is divided up between the team. Some people volunteer to do certain parts others just split work. Once work has been decided the Product Owner and Scrum master check up on the progress of each person’s work.

# How frequently updates are to be performed?

Updates are performed every time one has finished their work. The work one has done is pushed to GitHub allowing others to see then the group talks about what still needs to be done and how much time is left for the deliverable submission. This can be as often as once a day or multiple times a day for each member.

# How changes to the Plan are to be evaluated and approved?

Changes to the plan are evaluated and approved after discussion (on zoom or through slack/text) is made. Once the members have agreed upon a new updated plan. That plan is then put into motion and takes effect.

# How changes to the Plan are to be made and communicated?

Changes to the plan are communicated and made through slack/zoom. We discuss each other’s roles and our jobs and what these new changes to the plan means for each person and their work.

# Also includes history of changes made to the plan.

History of changes is stored on the KanBan, and also in the sprint Review/Backlog.