[Table of Content]

# 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

# References

React, JavaScript, and HTML

# 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 (Ideally you will have subsections 3.1.1, 3.1.2, …)

# Configuration Identification

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

Each document follows a strict naming convention *TutorGroup\_Deliverable\_i\_DocumentName*

# *Name configuration items (unique identifiers)*

The “i” signifies which deliverable the documents are associated with. The “i” would be replaced with a number (ex. 1) signifying the deliverable.

# *Acquiring configuration items (physical procedures)*

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.*

# Subcontractor/Vendor Control

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

# Release Management and Delivery

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

# SCM Schedules

# Sequence and coordination of SCM activities

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

# *Establishment of configuration baseline*

# *Implementation of change control procedures*

# *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.

# Key factors for infrastructure:

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

# Identify which tools are used in which activity.

# SCM Plan Maintenance

# Who is responsible for monitoring the plan?

# How frequently updates are to be performed?

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

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

# Also includes history of changes made to the plan.