Software Project Management Plan (SPMP)

The basic template to be used is derived from IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans.

**Software Project Management Plan**

**for**

***Air Traffic Controller LCMS***

***<Greg Friedman, Mark Holson, Brennan McPherson, and Howard Wheeler>***

***<2/27/2015>***

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Release Date** | **Responsible Party** | **Major Changes** |
| 0.1 | 3/6/2015 | Team 1 | Initial Document Release for Comment |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

***Table of Contents***

Build the table of contents here. Insert it when you finish your document.

***1. Introduction***

1.1 Project Overview

Include a concise summary of the project objectives, major work activities, major milestones, required resources, and budget. Describe the relationship of this project to other projects, if appropriate. Provide a reference to the official statement of product requirements.

In the FAA project, a system has to be put in place to modernize the training of air traffic controllers. The system will comprise of two state of the art training modules. Managers will be able to create “smart” lesson plans that are based upon trainees performance on quizzes and simulators.

1.2 Project Deliverables

|  |  |
| --- | --- |
| Deliverable | Date |
| User Requirements Document | February 17 |
| Configuration Management | TBD |
| Use Cases | February 17 |
| Risks & Mitigation | TBD |
| Project Schedule | TBD |
| Software Project Management Plan | March 6 |
| Software Process Selection | TBD |
| Design | TBD |
| Source Code | TBD |

1.3 Evolution of the SPMP

Describe how this plan will be completed, disseminated, and put under change control. Describe how both scheduled and unscheduled updates will be handled.

This document is subject to multiple changes and edits as the project continues. When circumstances cause changes in the constraints, requirements, or planning, then these new additions will lead to a SPMP with an updated version number. If these changes cause a change in a milestone, then they will be discussed with the Senior Management prior to incorporating them into the document. Weekly meetings will ensure that all project members are up to date with all changes and how they affect the SPMP.

1.4 Reference Materials

Provide a complete list of all documents and other sources of information referenced in the plan. Include for each the title, report number, date, author, and publishing organization.

1.5 Definitions and Acronyms

Define or provide references to the definition of all terms and acronyms required to properly interpret the SPMP.

LCMS: Learning Content Management System

SPMP: Software Project Management Plan

***2. Project Organization***

This section specifies the process model for the project and its organizational structure.

2.1 Process Model

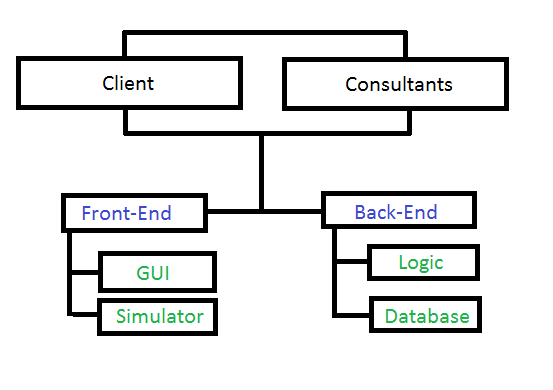
Specify the life cycle model to be used for this project or refer to an organizational standard model that will be followed. The process model must include roles, activities, entry criteria and exit criteria for project initiation, product development, product release, and project termination.

Use-cases:

* User tries to log in, but forgets password and needs to generate a new one
* Manager logs in and adds a new simulation to the program
* Manager logs in and views reports for a specific trainee
* Manager logs in and adds a list of new training goals
* Trainee logs in, takes his first test simulation, views the report of the simulation, and then logs out
* Trainee logs in, starts a simulation, but has an emergency and quits the program unexpectedly
* Trainee logs in and takes a simulation, fails the simulation and tries a new simulation
* Auditor logs in and views reports for a specific trainee
* Auditor logs in and views list of total time spent for a specific trainee

2.2 Organizational Structure

Describe the internal management structure of the project, as well as how the project relates to the rest of the organization. It is recommended that charts be used to show the lines of authority.



2.3 Organizational Interfaces

Describe the administrative and managerial interfaces between the project and the primary entities with which it interacts. A table may be a useful way to represent this.

|  |  |  |
| --- | --- | --- |
| **Organization** | **Liaison** | **Contact Information** |
| Customer: FAA | Professor Broadwater | rbroadwater@towson.edu |

2.4 Project Responsibilities

Identify and state the nature of each major project function and activity, and identify the individuals who are responsible for those functions and activities. Tables of functions and activities may be used to depict project responsibilities.

|  |  |  |
| --- | --- | --- |
| **Role** | **Description** | Person |
| Project Manager | leads project team; responsible for project deliverables | <name> |
| Technical Team Leader(s) | <define as locally used> | <name> |
| Front-end programmer | designs and manages front-end GUI and simulator |  |
| Back-end programmer | writes program logic and manages database |  |

Table F-2. Project Responsibilities.

***3. Managerial Process***

This section of the SPMP specifies the management process for this project.

3.1 Management Objectives and Priorities

Describe the philosophy, goals, and priorities for managing this project. A flexibility matrix might be helpful in communicating what dimensions of the project are fixed, constrained and flexible. Each degree of flexibility column can contain only one “X”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Dimension** | **Fixed** | **Constrained** | **Flexible** |
| Cost |  |  | X |
| Schedule | X |  |  |
| Scope (functionality) |  | X |  |

Table F-3: Flexibility Matrix

3.2 Assumptions, Dependencies, and Constraints

State the assumptions on which the project is based, any external events the project is dependent upon, and the constraints under which the project is to be conducted. Include an explicit statement of the relative priorities among meeting functionality, schedule, and budget for this project.

Due to the deadline of May 2015, the project team is under a fixed schedule. This being the only fixed project dimension, it holds the highest priority. The user requirements detailed out the functionalities of the project, each of which will be assigned a priority. In case of a rapidly approaching deadline, certain functionalities can be omitted that are less important.

3.3 Risk Management

Describe the process to be used to identify, analyze, and manage the risk factors associated with the project. Describe mechanisms for tracking the various risk factors and implementing contingency plans. Risk factors that should be considered include contractual risks, technological risks, risks due to size and complexity of the product, risks in personnel acquisition and retention, and risks in achieving customer acceptance of the product. The specific risks for this project and the methods for managing them may be documented here or in another document included as an appendix or by reference.

This section will detail possible major risks foreseen for the project in addition to steps to be taken in order to avoid the risks. They are divided into 4 categories:

1. Risks with respect to the work
2. Risks with respect to the management
3. Risks with respect to the resources
4. Risks with respect to the customer

Each category will detail possible risks for the project and steps for prevention.

### 3.3.1 Risks with respect to the work

1. Miscommunication

Prevention: Weekly meetings should be used to go over all current projects thoroughly in detail. Meeting minutes are kept so team members know who is working on what section. In case of a miscommunication arising, all members know who to go to with questions or concerns.

1. Time shortage

Prevention: Because the deadline is fixed, dealing with a time shortage puts pressure on all members of the team. To avoid issues, user requirements are ranked based upon priority. With clearance from the client and management, requirements with low priority are put aside.

1. Errors in design

Prevention: Proper planning and understanding of design process. Approach design in incremental steps.

1. Absent team members

3.4 Monitoring and Controlling Mechanisms

Define the reporting mechanisms, report formats, review and audit mechanisms, and other tools and techniques to be used in monitoring and controlling adherence to the SPMP. Project monitoring should occur at the level of work packages. Include monitoring and controlling mechanisms for the project support functions (quality assurance, configuration management, documentation and training).

A table may be used to show the reporting and communication plan for the project. The communication table can show the regular reports and communication expected of the project, such as weekly status reports, regular reviews, or as-needed communication. The exact types of communication vary between groups, but it is useful to identify the planned means at the start of the project.

**Weekly Group Meeting**

The project team will have group meetings every week on Thursday’s at 7pm. Meetings are held to review and refine the work from the week before, plan for the next week in regards to deadlines of deliverables, and the progress towards each milestone.

3.5 Staffing Approach.

We will need at least two programmers, one that will focus on front-end (GUI), one that will focus on back-end (logic). We already have the team set, so we do not need to recruit anyone else. Programming team will need to learn how to integrate the flight simulator into the GUI as well as the programming logic.

***4. Technical Process***

This section specifies the technical methods, tools, and techniques to be used on the project. It also includes identification of the work products and reviews to be held and the plans for the support group activities in user documentation, training, software quality assurance, and configuration management.

4.1 Methods, Tools, and Techniques

Identify the computing system(s), development method(s), standards, policies, procedures, team structure(s), programming language(s), and other notations, tools, techniques, and methods to be used to specify, design, build, test, integrate, document, deliver, modify or maintain the project deliverables

This project will be designed in Java. As an air traffic control simulator, we are going to use an open source software called jATC (found here: <http://sourceforge.net/projects/jatc/>) It will allow us to modify difficulties and other functionalities of the simulator. To keep track of version control, we will be using GitHub. The project repository can be found at: <https://github.com/Lifeboat13/Software-Engineering-Team-1>. Ruby on rails will be used to implement the database framework.

For development methodology, we will be taking an Agile approach, allowing for significant change in project requirements, which was suggested by the customer.

4.2 Software Documentation

Specify the work products to be built for this project and the types of peer reviews to be held for those products. It may be useful to include a table that is adapted from the organization’s standard collection of work products and reviews. Identify any relevant style guide, naming conventions and documentation formats. In either this documentation plan or the project schedule provide a summary of the schedule and resource requirements for the documentation effort.

Implementation of the software satisfies the requirements, the following documentation is required as a minimum:

4.2.1 Software Requirements Specification (SRS)

The SRS clearly and precisely describes each of the essential requirements (functions, performances, design constraints, and attributes) of the software and the external interfaces. Each requirement is defined such that its achievement is capable of being objectively verified and validated by a prescribed method, for example, inspection, analysis, demonstration, or test.

|  |  |
| --- | --- |
| Requirement | Validation |
| Software must allow users to login to the program | Demonstration |
| Software must differentiate users between: “Training Managers, Employees, and Auditors” | Inspection |
| Software must store user information including: Login, Password, First Name, Last Name, Employee ID (EID) and Department of Employment | Test |
| Software must store lessons and exercises based on specific Air Traffic Control information. | Test |
| Software shall allow Employees to select “lesson goals” based off of lessons stored in the program. | Demonstration |
| From those “lesson goals,” software will compile numerous lessons and exercises related to the chosen lesson goals to create a unique “Lesson Plan” for the employee to study. | Demonstration |
| Software shall launch an Air Traffic Control simulation for the employee to take and grade the employee on performance as well as elapsed time of the simulation. | Demonstration |
| Software shall save grades and elapsed time with association to the employee. | Test |
| Software shall allow Employees to view their scores for each simulation taken. | Demonstration |
| Software shall also allow Training Managers to view scores of every employee in the same department as the Training Manager. | Test |
| Software shall allow Auditors with valid “EID” to view times of employees simulation trials for legal reasons. | Test |
| Software must be fully documented | Inspection |
| Software must be privately accessed in house or via secure remote connection | Demonstration |
| Software must have an intuitive GUI with easy to follow instructions | Demonstration |
| Software must be able to be maintained and kept up to date | Inspection |

4.2.2 Software Design Description (SDD)

The SDD describes the major components of the software design including databases and internal interfaces.

The LCMS will be a tool that can be accessed from the web or locally, that will include lesson plans and interactive training for air traffic controllers. A third-party open source ATC simulator will be used to train users and gather information on their performance. A database will be used to store all users’ information such as personal information, lessons completed, and simulation results.

4.2.3 Software Test Plan

The Software Test Plan describes the methods to be used for testing at all levels of development and integration: requirements as expressed in the SRS, designs as expressed in the SDD, code as expressed in the implemented product. The test plan also describes the test procedures, test cases, and test results that are created during testing activities.

We will use unit testing throughout the software and ensure that every commit to the repository passes all unit tests.

4.3 User Documentation

Describe how the user documentation will be planned and developed. (This may be just a reference to a plan being built by someone else.) Include work planned for online as well as paper documentation, online help, network accessible files and support facilities.

The LCMS will provide online documentation and manuals how to use the system. In addition, the code will be well commented and documented for future updates.

4.4 Project Support Functions

Provide either directly or by reference, plans for the supporting functions for the software project. These functions may include, but are not limited to, configuration management, software quality assurance, and verification and validation. Plans for project support functions are developed to a level of detail consistent with the other sections of the SPMP. In particular, the responsibilities, resource requirements, schedules and budgets for each supporting function must be specified. The nature and type of support functions required will vary from project to project, however, the absence of a software quality assurance, configuration management, or, verification and validation plan must be explicitly justified in project plans that do not include them.

***5. Work Packages, Schedule, and Budget***

Specify the work packages, dependency relationships, resource requirements, allocation of budget and resources to work packages, and a project schedule. Much of the content may be in appendices that are living documents, updated as the work proceeds.

5.1 Work Packages

Specify the work packages for the activities and tasks that must be completed in order to satisfy the project agreement. Each work package is uniquely identified. A diagram depicting the breakdown of project activities and tasks (a work breakdown structure) may be used to depict hierarchical relationships among work packages.

Setup the data structures required

Create the database

Create functionality according to use-cases

Create GUI and tie together functionality (no simulator)

Create/modify simulator to our needs and integrate

5.2 Dependencies

Specify the ordering relations among work packages to account for interdependencies among them and dependencies on external events. Techniques such as dependency lists, activity networks, and the critical path method may be used to depict dependencies among work packages.

5.3 Resource Requirements

Provide, as a function of time, estimates of the total resources required to complete the project. Numbers and types of personnel, computer time, support software, computer hardware, office and laboratory facilities, travel, and maintenance requirements for the project resources are typical resources that should be specified.

5.4 Budget and Resource Allocation

Specify the allocation of budget and resources to the various project functions, activities, and tasks.

5.5 Schedule

Provide the schedule for the various project functions, activities, and tasks, taking into account the precedence relations and the required milestone dates. Schedules may be expressed in absolute calendar time or in increments relative to a key project milestone.

***6. Additional Components.***

Certain additional components may be required and may be appended as additional sections or subsections to the SPMP. Additional items of importance on any particular project may include subcontractor management plans, security plans, independent verification and validation plans, training plans, hardware procurement plans, facilities plans, installation plans, data conversion plans, system transition plans, or the product maintenance plan.

6.1 Index.

An index to the key terms and acronyms used throughout the SPMP is optional, but recommended to improve usability of the SPMP.

6.2 Appendices

Appendices may be included, either directly or by reference, to provide supporting details that could detract from the SPMP if included in the body of the SPMP. Suggested appendices include:

A. Current Top 10 Risk Chart

B. Current Project Work Breakdown Structure

C. Current Detailed Project Schedule