**Software Project Management Plan**

**for**

**UHD Learn ++ Learning Management System (LMS)**

Andrew Truong, Daniel Fabela, Eduardo Rojas, Hung Ly, and Sumanth Pisipati

**Version 1.0**

2/5/2019

# Table of Contents

**1. Introduction**

1.1 Project Overview

1.2 Project Deliverables

1.3 Evolution of the SPMP

1.4 Reference Materials

1.5 Definitions and Acronyms

**2. Project Organization**

2.1 Process Model

2.2 Organizational Structure

2.3 Organizational Interfaces

2.4 Project Responsibilities

**3. Managerial Process**

3.1 Management Objectives and Priorities

3.2 Assumptions, Dependencies, and Constraints

3.3 Risk Management

3.4 Monitoring and Controlling Mechanisms

3.5 Staffing Approach

**4. Technical Process**

4.1 Methods, Tools and Techniques

4.2 Software Documentation

4.3 Project Support Functions

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

5.1 Dependencies

5.2 Resource Requirements

5.3 Budget and Resource Allocation

5.4 Schedule

1. **Introduction**

This section of the SPMP provides an overview of the project.

* 1. **Project Overview**

The project is to create a software subsystem of learning management system meaning a web service application similar to Blackboard. The Blackboard web service is for students and instructors during their time in school, giving a way to look at student’s courses, material of the course, the grades made in the courses, and a way of communicating with others. Instructors also have similar access but having more editing tools for their course in the semester, such as assign homework to the class or showing the students their grades. This project will be a smaller scaled down version of Black Board, providing students a list of their courses, grades and GPA in the course, their name, and student ID. Instructors will be able to add and look at assignments in the course, view and edit grades, view student’s information, and calculate student’s GPA.

* 1. **Project Deliverables**

List the primary deliverables for the customer, the delivery dates, delivery locations, and quantities required satisfying the terms of the project agreement.

|  |  |  |
| --- | --- | --- |
| Deliverable | Description | Dates |
| Software Project Management Plan (SPMP) | A complete formal project plan, including technical and managerial processes that will be implemented in the development and delivery of the system. | 3/21/2019 |
| Requirement artifacts, analysis artifacts, and presentation of project’s progress. | UML Diagrams of use cases, class diagram, collaboration diagram, and sequence diagram. | 3/21/2019 |
| Product Software | The executable code for the LMS. | 4/28/2019 |
| Final Presentation | A demonstration of the product software. | 4/30/2019 |

**Table 1.2.1 Project Deliverables**

* 1. **Evolution of the SPMP**

All members of the project team have agreed upon this SPMP. Any team member may make changes to the SPMP, but the entire project team must approve all changes. This document may be periodically updated during the lifecycle of the product to reflect changes in the project schedule. All changes will be documented in order to keep the SPMP current. Andrew Truong will be responsible for coordinating and finalizing all changes to the SPMP.

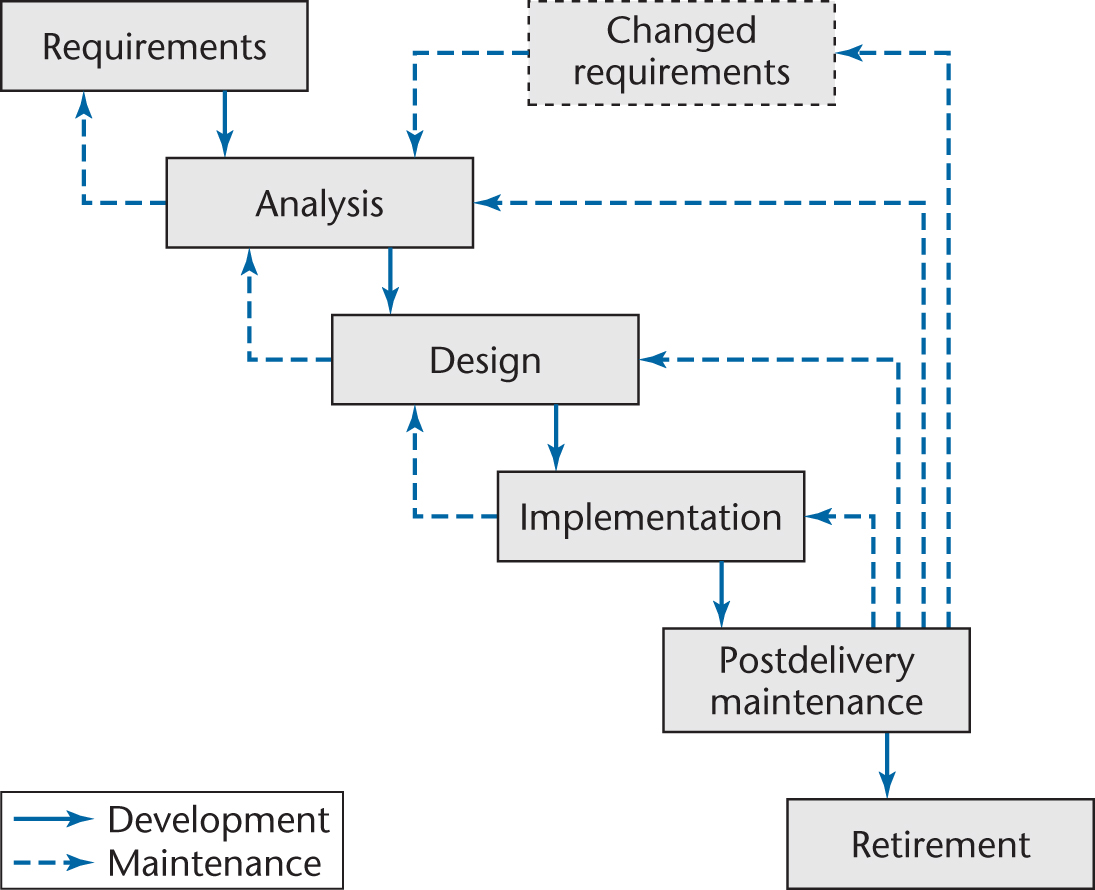
* 1. **Reference Materials**
* IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans
* Schach, Stephen R. *Object-Oriented and Classical Software Engineering*. McGraw-Hill, 2011.
  1. **Definitions and Acronyms**
* LMS: Learning Management System
* Slack: A project collaboration tool that helps our team keep track of: important documents, tasks that need to be completed, etc.
* Git Hub: A web-based hosting service for version control.
* Graphic User Interface (GUI): A category of interfaces that allows the user to visually interact with software (usually via a mouse or touch screen).
* C#: General-purpose, multi-paradigm programming language

# Project Organization

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

**2.1 Process Model**

The main process for this project will be the waterfall model.



**Figure 2.1.1 Waterfall Process Model**

**2.2 Organizational Structure**

Andrew Truong

Team Lead

Hung Ly

Programmer

Daniel Fabela

Application Designer

Sumanth Pisipati

Programmer

Eduardo Rojas

Technical Writer

**Figure 2.2.1 Organization Chart**

**2.3 Organizational Interfaces**

The project team will perform all of the work on the project. Questions regarding requirements and deliverables that cannot be answered from within the team will be directed to the client, Professor Yuchou Chang.

**2.4 Project Responsibilities**

The following table lists the roles necessary for the completion of the project. Each role has an assigned member who is responsible for the associated responsibilities. Although each member is assigned a role, all team members will serve as support staff for every other role as needed.

|  |  |
| --- | --- |
| Name | Role & Responsibilities |
| Andrew Truong | Team Lead – Act as main point of contact with instructor, assign duties to the rest of the team, ensure assignments are done on time, and assist in programming as needed. |
| Daniel Fabela | Application Designer – Manage the look-and-feel of the project, programs application, and ensure that the project’s GUI adheres to design standards |
| Eduardo Rojas | Technical Writer – Keep track of the paperwork and manage documentation as well as assist in programming. |
| Sumanth Pisipati | Programmer – Programs the application. |
| Hung Ly | Programmer – Programs the application. |

**Table 2.4.1 Team Roles**

# 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".

The foremost objectives of the project are the following:

1. All deliverables are completed and submitted to the client by 4/28/2019.
2. The final product software meets the requirements.
3. The client is left feeling satisfied with the deliverables that the team produced. Measures of the client’s satisfaction will be the grades that the team receives on project deliverables, as well as the final grades that the team receives.
4. The final product software is of good quality and maintainable.

|  |  |  |
| --- | --- | --- |
| **Project Dimension** | **Fixed** | **Flexible** |
| Resources | X |  |
| Hours Worked |  | X |
| Schedule | X |  |
| Scope |  | X |

**Table 3.1.1 Project Dimensions**

**3.2 Assumptions, Dependencies, and Constraints**

The project will be developed under the following organizational assumptions:

* The development team consists of six members who will contribute approximately even amounts of time to completing the project.
* Each team member will be willing and able to contribute at least six hours a week to the project.
* The team members are responsible for having the necessary tools to complete the project. Any thing outside of our scope will be brought to the client’s attention.
* The development team has enough experience as a whole to complete the project.
* The development team will cooperate to complete the project.

The project will be developed under the following project assumptions:

* The users of the system will be students and system administrators.
* The data for the system will be created by the team members.
* The system will properly run on Microsoft Windows operating system.
* The system will meet the requirements of the client.

**3.3 Risk Management**

The main risk to the success of the project is if each phase takes more time than anticipated. This risk is believed to be high since the project team has not worked together before or produced a product that meets the same requirements. If a phase is found to take significantly longer than expected, then the team will see if the scope needs to be reduced in order to meet the deadline for the project.

Another major risk is the possibility of losing a member of our team due to them dropping this course due to unforeseen circumstances. This would result in at least a 20% loss in manpower. This would affect the productivity and efficiency of the team. In order to mitigate this risk, tasks are divided evenly amongst the members. If a member drops, then their workload will be distributed to the remaining members. If the dropped member was in charge of a significant part of the project, then it will be given to another member.

Another risk is that the team has agreed to program this application in using C# language. Not every team member is knowledgeable in it. Most of the members know C++. This is not a huge risk due to the similarity between the two languages. This risk is mitigated through weekly meetings to demonstrate and learn what is necessary for the application. Constant communication will help cover any uncertainties and misunderstandings. Slack is an application the team is using to be in constant communication with one another and share files. Team members are expected to express any troubles they may be having so that they can be helped.

**3.4 Monitoring and Controlling Mechanisms**

Team meetings will be held weekly to determine the progress and work cooperatively on the deliverables. Any major problems will be reported as soon as discovered and handled as soon as possible. Reporting of these problems will be done at the meeting.

The meeting will also be used to work on the project cooperatively. This way, if a team member has an issue or question on what they are working on, they may ask then and receive help.

A Slack workspace has been created as a way to communicate with one another and share any files. Git Hub is used for version control. Every team member commit when they have worked on their part. If there are any huge errors, an earlier version of the application may be received and can be worked on.

**3.5 Staffing Approach.**

The five members have joined together to work on the project because of their varying knowledge and skills. The team should have the necessary skills to complete the project. There will be no recruitment of other members.

# Technical Process

The following sections describe the technical standards to be used in the project.

**4.1 Methods, Tools, and Techniques**

LMS will be developed using the waterfall method. This process is well suited due to the straightforward nature of the method and it falls in line with the scope of the project.

LMS will be developed on the team’s personal computers running either Windows Operating System 7 or later. The developers will use Visual Studio Integrated Development Environment (IDE) for coding, integration, compiling, and debugging. It will be a Windows Form Application in C# programming language. The data will be stored on csv files and be read into the application when needed. If changes are made to the data, it will be written onto the file. An alternative, if time permits, is to have a database run on MySQL or similar.

**4.2 Software Documentation**

All documentation for LMS will be created using Microsoft Word. The final document deliverables will be in Microsoft Word format.

The following IEEE standard document formats will be used for the final document deliverables:

* IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans

Finally, all presentation slides will be created using Microsoft Powerpoint.

**4.3 Project Support Functions**

All members of the team have joined a Slack workspace specifically created for this project. A Git Hub repository was created for version control. All document deliverables must be committed to Git Hub 48 hours before the deadline so that all team members can review the document and make corrections or give suggestions. All corrections, suggestions, and comments must be emailed or messaged through Slack 24 hours before the deadline so that corrections can be made on time. All team members must sign off final documentation 12 hours before the deadline.

When the coding phase starts, every Friday by midnight the developers will deliver software updates and commit their progress. Andrew Truong will oversee the progress and offer comments, suggestions, and direction.

# Work Packages, Schedule, and Budget

**5.1 Dependencies**

The dependencies for the project are outlined below.

|  |  |
| --- | --- |
| **Task** | **Dependencies** |
| Software Project Management Plan (SPMP) | None |
| Test scenarios | SPMP |
| Coding | SPMP |
| Testing | SPMP, Coding |
| User guide | SPMP |
| Final demonstration | All of the above |

**Table 5.1.1 Task Dependencies**

**5.2 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.

The project team will do all of the work on the project. No additional resources will be required. The following table details the expected person-hours for each major task.

The time are estimates based on the scope of the project, experience with other projects in application building, and the technology used on the project.

|  |  |
| --- | --- |
| **Task** | **Effort in Person-Hours** |
| Software Project Management Plan (SPMP) | 10 |
| Coding | 100 |
| Testing | 20 |
| User guide | 10 |
| Final Presentation | 15 |
| Project Planning | 20 |
| **Total Effort** | **175** |

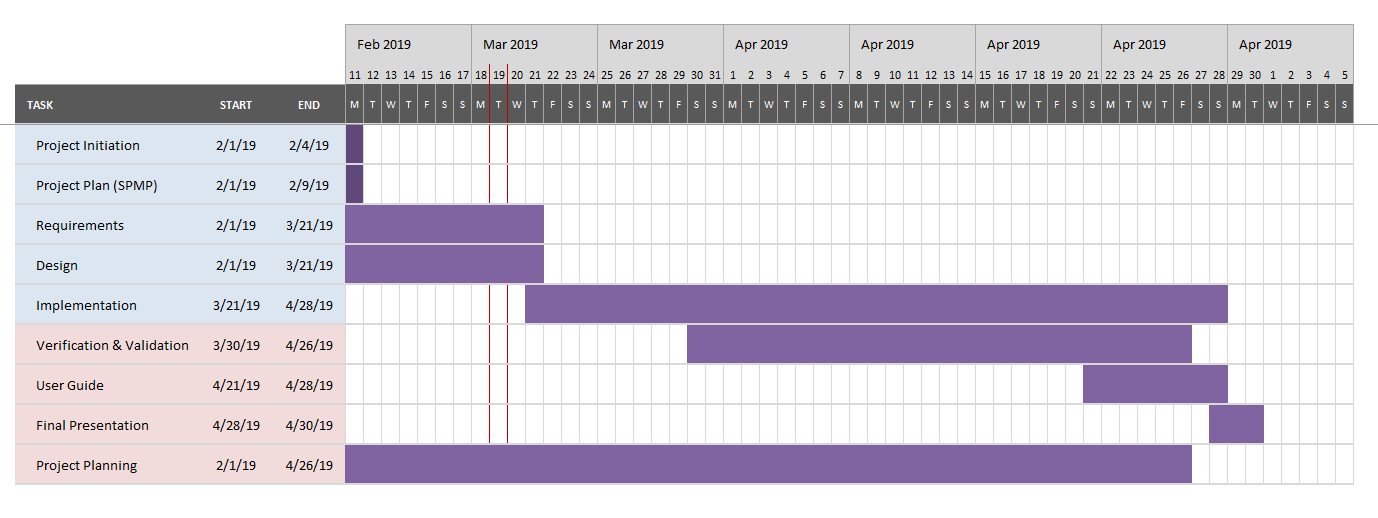
**Table 5.2.1 Task**

**5.3 Budget and Resource Allocation**

The project team will perform all of the work on the project. Resources in the form of members will be allocated evenly throughout the project.

**5.4 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.



**Figure 5.4.1 Task Schedule**