**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. The following is a template for the SPMP. It begins with a cover page that contains the version control and release information. Each section has a description of the information contained within.

Software Project Management Plan for

**UHD Learn ++ Learning Management System**

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

<date>

Version Release Responsible Party Major Changes Date

0.1 Initial Document Release for Comment

# Table of Contents

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

1. Introduction

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

* 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.**

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. Administrator will also be added software subsystem of learning management system into the giving them the ability to add or remove courses and users.

* 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 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 |

* 1. **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.

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**

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.

* IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans
* IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications
* IEEE Std 1016-1998, IEEE Recommended Practice for Software Design Descriptions
  1. **Definitions and Acronyms**

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

* 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**

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.

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

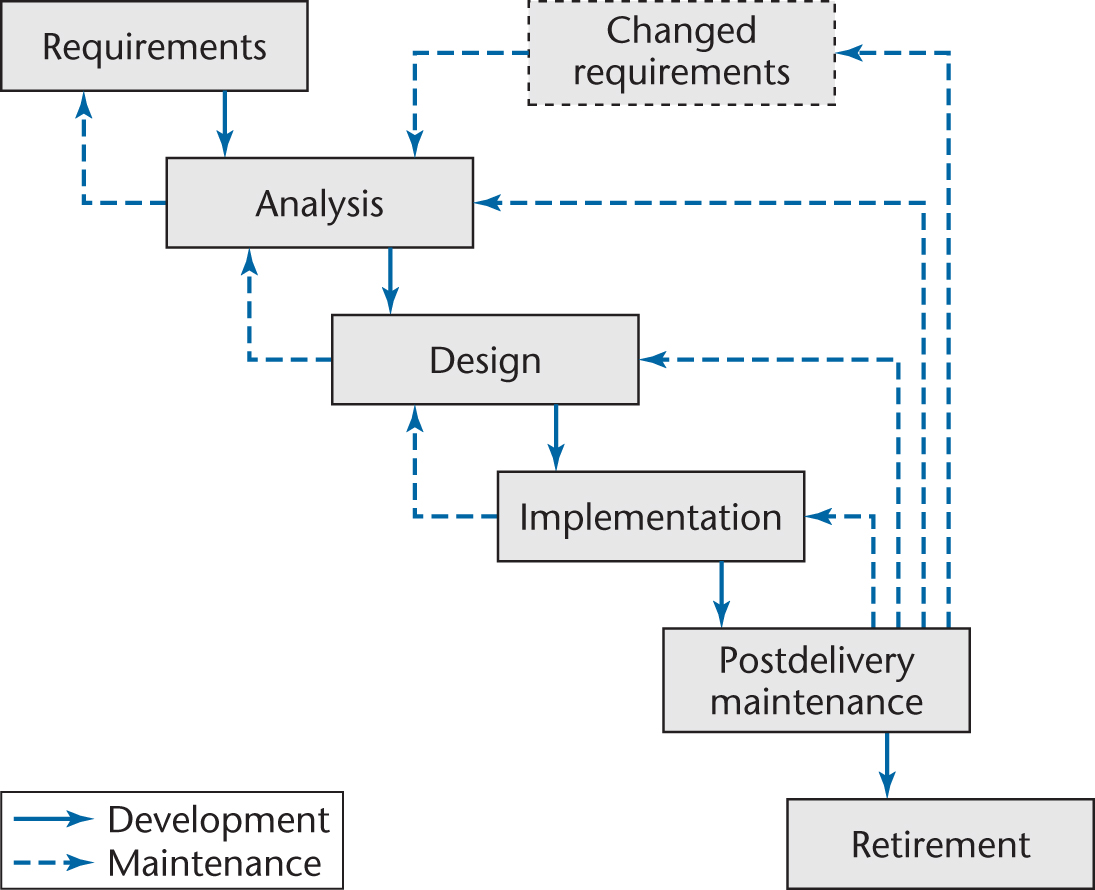


Figure 1. Waterfall Process Model

**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.

Andrew Truong

Team Lead

Hung Ly

Programmer

Daniel Fabela

Application Designer

Sumanth Pisipati

Programmer

Eduardo Rojas

Technical Writer

**Figure 2. 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 instructor, Professor Yuchou Chang.

**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.

|  |  |
| --- | --- |
| 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 | Quality Assurance Specialist – Manage the look-and-feel of the project, programs application, and ensure that the project’s GUI adheres to design standards |
| Eduardo Rojas | Secretary – 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 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. Project Dimensions**

**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 thi project.

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**

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.

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.**

Describe the types of skills required for the project, how appropriate personnel will be recruited, and any training required for project team members.

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

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

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.

**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.

To ensure that the 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.

**4.2.2 Software Design Description (SDD)**

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

**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.

**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.

**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. The absence of a software quality assurance, configuration management, or verification and validation plan, however, must be explicitly justified in project plans that do not include them.

# 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.

**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.

# 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:

1. Current Top 10 Risk Chart
2. Current Project Work Breakdown Structure
3. Current Detailed Project Schedule