Software Project Management Plan

For

A Software Subsystem of a Learning Management System

Software Engineering Team Project – CS 3321

Spring 2020

Team: Magdalena Hernandez, Rick Jenkins, Jose Ramos, Mario Rodriguez

May 1, 2020

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Updated By** | **Update Comments** |
| 0.1 | 2/6/2020 | Magdalena Hernandez, Rick Jenkins | Initial Document Template & Setup |
| 0.2 | 2/9/2020 | Project Team | 1st Draft |
| 0.3 | 4/15/2020 | Rick Jenkins | Update project delivery date to new due date of 5/3/2020 |
| 1.0 | 5/1/2020 | Rick Jenkins | Final Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Introduction 4](#_Toc39251614)

[2. Project Organization 6](#_Toc39251617)

[3. Managerial Process 8](#_Toc39251618)

[4. Technical Process 10](#_Toc39251619)

# Introduction

# 

This is software project management plan (SPMP) for a subsystem of a learning management system (LMS). Learning Management Systems (LMS) are software products that deliver educational courses, training programs, and/ or learning and development programs. LMS key features are designed to allow users to manage the administration, documentation, tracking, reporting, and delivery of courses.

This SPMP documents the development of a software system intended for a university that will deliver educational courses, include tracking and reporting features, and manage the administration of courses.

* 1. **Project Overview**

This document is intended for the members of the project describing the managerial aspects and technical aspects. The document is intended for planning and scheduling purposes and serves as a summary document of the deliverables expected from each of the teams.

Objective: To deliver a subsystem of an LMS that will: manage the administration of courses; store and retrieve student’s courses, id numbers, exam scores, and semester grade point average (GPA); and include tracking and reporting features.

Major work activities:

* Software Planning: Create SPMP, select team model for project, select architecture design, create use-case diagrams
* Software Analysis
* Create UML diagrams for software
* Building a database to hold course and student information for the current semester
* Design a Graphical User Interface (GUI) for software
  1. **Project Deliverables**

|  |  |
| --- | --- |
| **Deliverables** | **Date** |
| Software Project Management Plan | 5/3/2020 |
| UML Diagrams | 5/3/2020 |
| Executable Software | 5/3/2020 |
| Software Source Code | 5/3/2020 |
| Version Control Documentation | 5/3/2020 |
| Software Test Cases | 5/3/2020 |
| Data Storage Files | 5/3/2020 |

* 1. **Evolution of the SPMP**

The software project management plan is under version control. Proposed changes and new versions of the plan announced via email and are made available to all the project members on the project’s GitHub document repository. This document will be updated as the project progresses. Updates should be expected in the following sections:

* ***Reference Materials*** - updated as necessary
* ***Definitions, acronyms, and abbreviations*** - updated as necessary
* ***Project Organization*** - will be updated as team members are assigned for each function
* ***Technical Process*** - this section will be revised appropriately as the requirements and design decisions become clearer
* ***Schedule*** – as the project progresses, the schedule will be updated accordingly
  1. **Reference Materials**
* [Schach 11] Stephen R. Schach: Object-Oriented and Classical Software Engineering, McGraw-Hill, 8th Ed, 2011, ISBN-10: 0-07-337618-3
* [Weiss 10] Mark Allen Weiss: Data Structures & Problem Solving Using JAVA, Addison Wesley, 4th Ed, 2010, ISBN-10: 0-321-54140-5
* [Chang 20] Yuchou Chang: Project\_Assignment.PDF
* Project Team Repository: <https://github.com/UHD-Software-Engineering/cs3321-team-project>
  1. **Definitions, Acronyms and Abbreviations**

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

GPA - Grade Point Average

GUI - Graphical User Interface

JDK – Java Development Kit

LMS - Learning Management Systems

SPMP - Software Project Management Plan

UML - Unified Modeling Language

# Project Organization

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

* 1. **Process Model**

The project uses an object-oriented design methodology based on the Waterfall lifecycle process as shown in Fig 1 and uses UML for the development of the software. The development process is organized in several activities. The members of the project are organized in teams. At the end of each activity up to and including testing, each team submits documents describing the achievement of the activity. The individual approved documents produced by the teams are considered work products and are part of the software documentation. The team documents are under version control using GitHub. Links to the team documentation are available in this document. The activities and milestones are described in the next following sections.

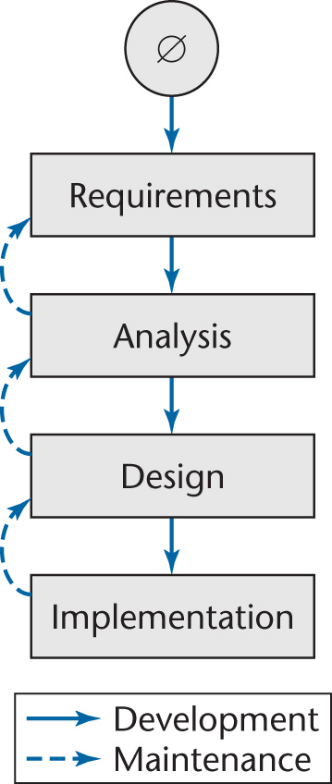


Fig. 1

* 1. **Organizational Structure**

Team members

* Rick Jenkins
* Jose Ramos
* Mario Rodriguez
* Magdalena Hernandez
  1. **Project Responsibilities**

Ultimately the whole complete development of project team is responsible for the successful delivery of the product. The team member tasks per deliverable according to expertise and the phases below:

* Project Plan & SPMP – Rick Jenkins
* Requirements Specification – Entire Team
* Analysis – Mario Rodriguez, Rick Jenkins, Jose Ramos,
* Architecture Spec – Magdalena Hernandez, Mario Rodriguez
* Component/Object Specification – Magdalena Hernandez, Mario Rodriguez
* Source Code – Jose Ramos, Rick Jenkins, Magdalena Hernandez
* Database SQL - Magdalena Hernandez, Jose Ramos
* Test Plan – Mario Rodriguez
* Final Deliverable – Entire Team.

# Managerial Process

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

* 1. **Management Objectives and Priorities**

The management objective is to deliver the product in time and of to specifications. The PM and the team work together to achieve this by respectively checking that progress is made as planned and monitoring the quality of the product at various stages.

* 1. **Assumptions, Dependencies, and Constraints**

In this project plan, a number of factors are taken into account. For these refer the following list shows the way milestones on various project phases that have scheduled:

* The team budget of 4 persons x 26 hours = 104 hours
* The project deadline of May 3rd
* The final presentation of May 3rd
* The peer evaluation deadline of May 3rd
* Two-week period of Spring Break March 9th – March 14th is closed

**NOTE:** Due to impact of the Covid-19 virus, the State and local government’s ordered shutdowns of campus and quarantine will have its reflection on the product, and not on the duration of the project. By assigning a priority of deliverables, selection deliverables may be dropped out if time runs out.

* 1. **Risk Management**

The goal of risk management is to identify and mitigate potential sources of expense or delay. Some risks are common to every project phase, and some risks are closely associated with a project phase. Risks for this project have been classified accordingly:

**Common Risks**

* **Team Member Unavailability**. During the course of a project, it is almost certain that some members of the team will be unavailable for certain project activities due to illness or emergency.

**Probability:** High

**Impact:** Low

**Prevention:** Team members should alert the team at the first opportunity regarding potential absences, and coordinate with other members of the team to cover their responsibilities for the duration of the absence.

**Correction:** Excessive (or unannounced) unavailability will trigger a team discussion.

* **Sponsor Unavailability.** The project sponsor has a large number of responsibilities and will not always be available.

**Probability:** High

**Impact:** Medium

**Prevention:** Sponsor meetings are on a standing schedule based on class and office times and announced in advance. Communication channels are available for the sponsor to notify the team of potential unavailability. The team will never assume sponsor availability without confirmation.

**Correction:** If circumstances dictate that the sponsor is unavailable to adegree that will negatively impact the project, the team asks that the sponsor appoint a replacement sponsor representative.

* **Miscommunication**. The volume of communication regarding this project almost guarantees that miscommunications will occur.

**Probability:** High

**Impact:** Medium

**Prevention:** The primary method of avoiding miscommunication is to document and verify verbal communications. For this project, the documentation and verification process will consist of meeting notes, meetup chat logs and emails. To avoid internal miscommunication, regular team meetings.

**Correction:** If miscommunications are occurring regularly, the sponsor and team will reassess the documentation and verification process.

* **Changes to Project Scope.** Changes to project scope are a common request but can derail project timelines.

**Probability:** Low

**Impact:** High

**Prevention:** The Project Assignment document and this document define processes for

handling change requests in a responsible manner. Because this a defined project assignment it is not expected that the scope will be subject to change.

**Correction:** The defined processes for change requests specify deadlines for such changes. Those deadlines will be honored to mitigate the risk of disruptive scope change.

* **Missed Deadlines**. As students, some of which are also employed full time, the project team face several external responsibilities that could detrimentally affect their ability to complete tasks on time.

**Probability:** Medium

**Impact:** High

**Prevention:** During weekly apportioning of responsibilities, team members are expected to anticipate their non-project-related time commitments and accept only work they can reasonably expect to accomplish by the given deadline.

**Correction:** Multiple failures to meet deadlines will trigger a team discussion.

* 1. **Monitoring and Controlling Mechanisms**

The monitoring of progress is done by the PM using the following means:

* Regular project status meetings after class
* Shared document repository
* Project tracking by MS project plan

# 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.

* 1. **Methods, Tools, and Techniques**

The Methods, Tools, and Techniques section aims to outline specific plans, methods, or tools to be used during this project.

* **Diagrams.** The Project Team will use standard UML diagrams to represent data, relationships, and requirements. Specific models and the requirements phase in which they are used can is shown below.

|  |  |  |
| --- | --- | --- |
| **Workflow** | **Description** | **Diagram** |
| Requirements | The development team review the Project Assignment document to identify the main ‘actors’ of the system and how they interact with the system. | Use Cases |
| Analysis | Those developing the system will analyze the various ‘entities’ involved in creating the system and how they interact with each other. These diagrams will be used to outline the requirements of the system. | Sequence Diagrams |
| Design |  | Class Diagram |

* **Programming Languages & Tools.** The Project Team will develop the program in Java and will utilize various tools such as Microsoft Project, GitHub, JavaSDK, MSSQL, ArgoUM, Visual Paradigm Online Diagrams, Apache Netbeans IDE, and Eclipse IDE
  1. **Software Documentation**

To ensure that the implementation of the software satisfies the requirements, the following documentation is required as a minimum:

* + 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.

* + 1. **Software Design Description (SDD)**

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

* + 1. **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.

1. **Schedule**

