**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. Provides an detailed overview of the procedures which go towards the completion of the software project. It surfaces the managerial and technical processes and showcases a general organization of a software product.

Software Project Management Plan for

<Project Title: LMS For Change>

<Contributors: Stephen Autry, Eric Anaya, Brian Ruiz, Gerardo, >

<Date: November 2018>

Version Release Responsible Party Major Changes Date

0.1 Initial Document Release for Comment

**TABLE OF CONTENTS**

Build the table of contents here. Inse1t it when you finish your document.

1. **Introduction / Overview**

This collaborative project seeks to create a foundational *platform* application for management of student accounts and gradebooks with multi-use for either k12 or universities.

* 1. **Project Overview**

This platform allows for the ergonomic management of registered students and there respective campuses administrators. Comparable to already established third party applications like *Blackboard.com.* The collaborators will work together to make this an easily accessible and free product, thus the budget will be limited.

* 1. **Project Deliverables**

Primary deliveribles for the client is to provide a product which will efficiently store student data with high consideration for security of the data.

* 1. **Evolution of the SPMP**

The programming team will go about developing the final product through means of an evolution tree life cycle model and will use cloud repositories to collaborate on the product accordingly.

**PROJECT ORGANIZATION**

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

2.**1 Process Model**

We decided to follow the evolution-tree life-cycle model that way we can go through the process and keep modifying the program till we get one that fits the criteria the best.

2.2 **Organizational Structure**

We went with the chief program style to make decision easier and safe time on discussion

**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 |
| Chief Programmer | Leads project team; | Stephen |

responsible for project

|  |  |  |
| --- | --- | --- |
| Programmers | Worked on writing the  The program and GUI | Eric Anaya  Oscar Duarte  Gerado |

|  |  |  |
| --- | --- | --- |
| Secretary | Was in charge of documentation | Brian |

**MANAGERIAL PROCESS**

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

3.**1 Management** **Objectives** and **Priorities**

Our goal is to create a program with a GUI that hit includes all that is asked by the client. The top priority is for it to be on schedule and have at least the basic needs the client wants. With the time we have left we will add any stuff we think the client would want.

Project Dimension Cost

Schedule

Scope (functionality)

Fixed Constrained Flexible

# x

x

# x

Table F-3: Flexibility Matrix

3.2 Assumptions, Dependencies, and Constraints

The project is dependent on us being able to work independently and be able to combine what we have toward the project. The member will have to look up any information or algorithm that is needed for the program.

3.3 Risk Management

What we did is that after a state of the program has been completed another member will test it out and give feedback on the program. The members will discuss potential solutions and any problems that may come up before writing or updating the program. A copy of the base program should always be saved in case something happens to the current program.

## 3.4 **Monitoring** and **Controlling** **Mechanisms**

Define the repo1ting mechanisms, rep01t 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 suppo1t 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 repo1ts 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 stait of the project.

Information From To Tin1e Period Communicated

Status repo1t Project Team Project Manager Weekly

Status repo1t Project Manger Software Manager, Project Weekly Team

Project Review Project Team Softwai·e Manager Monthly

<etc>

Table F-4: Communication and Repo1ting Plan

3.5 **Staffing Approach**

Members will be chosen I class.

**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**

For this project we came to the agreement to use netbeans, since most of the team is familiar with java. For the GUI netbeans has a program already included so we are going to use that. The programmers can use any technique and method if it is comprehendible to the others or the programmer can easily explain it to the others. The programmer is to finish a version of their program and hand it to another programmer for testing and to help updated it.

4.2 **Software** **Documentation**

For documentation we will have the SPMP and the UML to guideline our project.

* + 1. **Software** **Requirements** **Specification** (SRS)

The project should have the requirements including student’s name, student’s ID, registered courses in the current semester, each exam’s score in one course, GPA calculation in the current semester. The project should also have and admin to add a student and edit any changes.

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

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

* + 1. **Software** **Test** **Plan**

After a programmer is done with a program, the program should be run and tested by another programmer. Then program should fix any problems found and the programmer should integrate any input he thinks would benefit the program. The program should be tested after every new iteration of the program.

* 1. **User** **Documentation**

Describe how the user documentation will be planned and developed. (This may bejust 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.

This group will resort be resorting to Github for documentation, and the individual contributors seeked to provide detail solutions along with comments for others to comprehend and for easier trouble shooting through version control.

* 1. **Project Support Functions**

Provide either directly or by reference, plans for the supporting functions for the softwai·e project. These functions may include, but are not limited to, configuration management , software quality assurance, and verification and validation. Plans for project support functions ai·e 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 ftmction must be specified. The nature and type of suppo1t functions required will vaiy from project to project. The absence of a softwai·e 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 & 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 laborat01y 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, considering 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**

Ce1tain additional components may be required and may be appended as additional sections or subsections to the SPMP. Additional items of impo1iance 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 te1ms 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 suppo1ting details that could detract from the SPMP if included in the body of the SPMP. Suggested appendices include:

A. Current Top 10 Risk Chait

1. Current Project Work Breakdown structure
2. Current Detailed Project Schedule

END OF DOCUMENT