**CEBU INSTITUTE OF TECHNOLOGY**

**UNIVERSITY**

COLLEGE OF COMPUTER STUDIES

Software Project Management Plan

For

LOOP - Learning Object Organizer Plus V2.0

Signature

|  |  |  |
| --- | --- | --- |
| *Name* | *Position* | *Signature* |
| *Edwin Langga* | *Back-end Developer/Documentation* |  |
| *Marc Venry Aton* | *Team Leader/Back-end Developer/Documentation* |  |
| *Jessie Roxas Jr.* | *Front-end Developer/Documentation* |  |
| *Krizette Genn Quilaton* | *Front-end Developer/Documentation* |  |

Change History

|  |  |  |
| --- | --- | --- |
| Revision Number | Reused By | Revision |
| R1.0 | Edwin Langga | Use case diagramming |
|  |  |  |

Preface

The purpose of the Software Project Management Plan is to show a detailed report on how the specified project would be accomplished on the managerial level. All plans, budgets, schedules, etc. will be laid out on this document as a guide for both the client and the management staff of the intended software project for the entire software development process.

This document assumes that the reader is already fairly knowledgeable on the concepts of software engineering since this document is part of a software engineering process.

Contents will be ordered and classified according to the IEEE standard 1058-1998, which contain the standards for creating an SPMP.

Table of Contents

Signature 2

Change History 3

Preface 4

Table of Contents 5

List of Figures 6

List of Tables 7

Overview 8

1.1. Project Summary 8

1.1.1. Purpose, scope and objectives 8

1.1.2. Assumptions and constraints 8

1.1.3. Project deliverables 9

1.1.4. Schedule and budget summary 9

1.2. Evolution of plan 10

References 11

Definitions 12

Project organization 13

1.3. External interface 13

1.4. Internal structure 13

4.3. Roles and responsibilities 13

Managerial process plans 15

1.5. Start-up plan 15

1.5.1. Estimation plan 15

1.5.2. Staffing plan 15

5.1.3. Resource acquisition plan 15

5.1.4. Project staff training plan 15

1.6. Work plan 16

1.6.1. Work activities 16

1.6.2. Schedule allocation 17

5.2.3. Resource allocation 17

5.2.4. Budget allocation 17

5.3. Control plan 17

5.3.1. Requirements control plan 17

5.3.2. Schedule control plan 17

5.3.3. Budget control plan 17

5.3.4. Quality control plan 18

5.3.5. Reporting plan 18

5.3.6. Metrics collection plan 18

5.3.7 Risk management plan 18

5.3.8 Project closeout plan 18

6. Technical process plans 19

6.1 Process Model 19

6.2 Methods, tools, and techniques 20

6.3 Infrastructure Plan 20

6.3 Product Acceptance Plan 21

7. Supporting process plans 21

7.1. Configuration management plan 22

7.2. Verification and validation plan 22

7.3. Documentation plan 22

7.4. Quality assurance plan 23

7.5. Reviews and audits 23

7.6. Problem resolution plan 24

7.7. Subcontractor management plan 25

7.8. Process improvement plan 25

List of Figures

[Figure 1] External Organization Diagram – pg 13

[Figure 2] Internal Organization Diagram – pg 13

[Figure 3] Work Activities – pg 16

*[Figure 4] Process Model – pg 19*

List of Tables

[Table 1.1] Schedule and Budget Summary Table – pg 9

[Table 1.2] Evolution Plan – pg 10

[Table 4.3.1] Roles and Responsibilities – pg 14

[Table 4.3.2] Role Allocation – pg 14

[Table 5.1.2] Staff Plan – pg 15

[Table 5.2.4] Budget Allocation – pg 17

[Table 6.2.1] Methods and Techniques – pg 20

[Table 6.3.1] Methods and Techniques – pg 20

# Overview

## Project Summary

### 1.1.1. Purpose, scope and objectives

The system is intended to organize and filter learning objects, which are actually interactive tutorials, and videos that aid in learning a certain subject or topic. The purpose is to easily trap the learning objects that aren’t ready to be distributed for the student’s learning consumption. It would be a catastrophe if a learning object was distributed to students and it did not go through various reviews before it was approved. This is why a strong and reliable organizing and filtering system is required.

Registered developers may upload new learning objects and reviewers may review learning objects. There are specific features that a developer and a reviewer are able to use. A developer can upload new learning objects and delete the learning objects he/she has uploaded, search all the learning objects, and view all the learning objects that belong to him/her. On the other hand, a reviewer cannot upload new learning objects; he/she can only view, search, download and review learning objects with a rating of one (1), two (2) or three (3).

The system organizes learning objects into the following categories: “Newly Uploaded/For Review”, “Reviewed/For Review” and “Ready to Use”. Organizing the learning objects this way ensures that all the learning objects that are ready to use have been tested and reviewed. The expected users for our system would be the faculty and staff of CIT-University, depending on the course or topic of a particular learning object. This means that the system would be deployed on the CIT-U eLearning server and would be consequently only visible within the campus network. Students are out of our scope, they will be handled by Informatron CMS along with the feature to download the ready to use learning objects.

The system aims to create a cloud framework version of the already existing “Learning Objects” system that enables viewing of learning objects directly from the server. Instead of viewing the learning objects straight from the CIT-U eLearning server, developers and reviewers may view them on their respective devices through our system.

The objectives of the project are as follows:

* To complete the project by the project due date
* To work together as a team to complete the project
* To make a great web application that would organize and filter learning objects while having a user-friendly interface that still meets the industry standards.
* To enable an administrator to control the users, and their activities

### 1.1.2. Assumptions and constraints

The project will be planned with the following assumptions:

* This project is a component of a larger project
* This project will only deliver the software components of the larger project
* The larger project that this project is a part of has already defined the hardware that the

software will run on

The project will be planned with the following constraints:

* Time

o 5 months.

o Once the software is completed it will take about a week to integrate it with the other

systems.

* Staff

o One expert from the CIT-U eLearning Center will act as the project consultant.

* Maintenance

o The software will be maintained by the staff of the CIT-U eLearning Center.

### 1.1.3. Project deliverables

As part of the project, our team will deliver the following to the client:

- Project Proposal and Systems Requirement Specification (SRS) document.

- Project Management Artifacts such as the Software Project Management Plan.

- Status reports (throughout the project lifecycle) – on a weekly or case-by-case basis.

- Software Design Description (SDD).

- Software Testing Document (STD).

- A working version of LOOP – Learning Object Organizer Plus.

### 1.1.4. Schedule and budget summary

|  |  |  |
| --- | --- | --- |
| Project Deliverables | Deadlines | Budget |
| Project Proposal | 6/20/2014 | $700.00 |
| Software Requirements Specification (SRS) | 7/23/2014 | $1,000.00 |
| Software Project Management Plan (SPMP) | 8/6/2014 | $2,800.00 |
| Software Design Document(SDD) | 8/20/2014 | $200.00 |
| Software Test Document (STD) | 8/27/2014 | $1,900.00 |
| Software Output (Development) | 9/3/2014 | $11,600.00 |

*[Table 1.1] Schedule and Budget Summary Table*

All deliverables are given different deadlines. The project should be finished on the first week of October. The total duration of this project is 68 days, and the total budget cost is $21,200.00.

## Evolution of plan

| Version | Description of Version | Date Expected |
| --- | --- | --- |
| Draft | Rough draft created for initial submission and review comments. | August 13, 2014, 2014 |
| Preliminary | Second draft incorporating initial review comments, submitted for final review. |  |
| Initial | The initial SPMP. |  |

[Table 1.2] Evolution Plan

# References

SPMP standards - IEEE Std 1058-1998

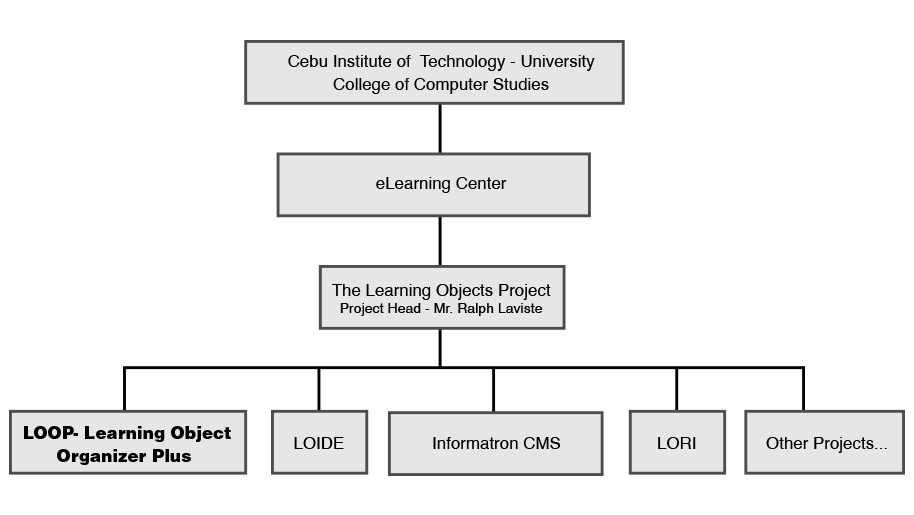
Sample SPMP - Karl E. Wiegers SOFTWARE REQUIREMENTS, 2003 Microsoft

# Definitions

1. **Software Requirements Specification (SRS)** – Document that contains all requirements agreed upon by the client and our team (Team LOOP)
2. **Software Design Description (SDD) –** Document that specifies the system design to be followed during implementation.
3. **Software Testing Document (STD) –** Document that contains information on testing methods and standards to be used for project testing.
4. **(Synchronized) Build –** Partial or fully developed software that is ready for scenario testing
5. **Scenario Testing –** Refers to testing a system as a whole rather than by small parts.
6. **Interface –** a means of communication or interaction between separate entities.
7. **User Interface** **(UI) –** Usually refers to a system/program’s screen/s.
8. **Backlog –** Work that is overdue or behind schedule.
9. **Netbeans -** Fully-featured Java IDE written completely in Java, with many modules available, such as: debugger, form editor, object browser, CVS, emacs integration, etc.
10. **MySQL –** An open source database that is preferable to web applications.
11. **Javascript -** is an [interpreted](https://en.wikipedia.org/wiki/Interpreter_(computing)) computer [programming language](https://en.wikipedia.org/wiki/Programming_language). A popular programming language that is supported in all Web browsers and other Web tools. It enables interactive functions to be added to Web pages, which are otherwise static. JavaScript evolved from Netscape's LiveScript language.
12. **JQuery -** is a [JavaScript](http://www.techterms.com/definition/javascript) library that allows web developers to add extra functionality to their websites. It is [open source](http://www.techterms.com/definition/opensource) and provided for free under the MIT license. In recent years, jQuery has become the most popular JavaScript library used in [web development](http://www.techterms.com/definition/web_development).
13. **HTML5 -** is a [markup language](http://en.wikipedia.org/wiki/Markup_language) for structuring and presenting content for the [World Wide Web](http://en.wikipedia.org/wiki/World_Wide_Web) and a core technology of the [Internet](http://en.wikipedia.org/wiki/Internet). It is the fifth revision of the [HTML](http://en.wikipedia.org/wiki/HTML) standard (created in 1990 and standardized as HTML 4 as of 1997)
14. **CSS3 -** is a [style sheet language](http://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation semantics](http://en.wikipedia.org/wiki/Presentation_semantics) (the look and formatting) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language).
15. **PHP -** is a [server-side scripting](http://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](http://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](http://en.wikipedia.org/wiki/General-purpose_programming_language).
16. **AJAX -** an acronym for asynchronous [JavaScript](http://en.wikipedia.org/wiki/JavaScript) and [XML](http://en.wikipedia.org/wiki/XML), used on the [client-side](http://en.wikipedia.org/wiki/Client-side) to create [asynchronous](http://en.wikipedia.org/wiki/Asynchronous_I/O) [web applications](http://en.wikipedia.org/wiki/Web_application).
17. **Bootstrap –** Sleek, intuitive, and powerful front-end framework for faster and easier web development.
18. **CodeIgniter -** is an open source rapid development web application framework, for use in building dynamic web sites with PHP.
19. **LOOP –** acronym forLearning Object Organizer Plus.
20. **LOIDE –** acronym forLearning Object Integrated Development Environment.
21. **LORI –** acronym forLearning Object Review Instrument
22. **WBS –** Work Breakdown Structure
23. **IDE** - Integrated Development Environment

# Project organization

## External interface



[Figure 1] External Organization Diagram

## Internal structure

Ralph Laviste

Project Manager

Marc Venry Aton

Team Leader

Jessie Roxas Jr.

Team Member

Krizette Genn Quilaton

Team Member

Edwin Langga Jr.

Team Member

[Figure 2] Internal Organization Diagram

## Roles and responsibilities

This section describes the organization of team LOOP for the development of this system. This section will be updated as the project progresses.

| Role | Responsibilities |
| --- | --- |
| Project Manager | Supervises the team throughout the development process.  Validates all deliverables and reviews all documentation. |
| Team Lead | Software project planning and monitoring.  Milestone and schedule planning and monitoring.  Set and communicate the team meeting agendas.  Finalizes all project documentation subject to manager’s review. |
| Analyst | Makes sure the UI’s are well designed to meet user requirements.  Ensure the SRS has completely covered all user’s requirements  Responsible to track user’s requirements through entire project |
| Developer | Responsible for hard coding.  Ensures the completion of assigned coding tasks within allocated time.  Performs primary debugging tasks |
| Tester | Tests all modules/aspects of the system after development stage.  Performs secondary debugging tasks. |
| Deployment Team | Makes sure the system is deployed and ready to use for the client. |

[Table 4.3.1] Roles and Responsibilities

| Role | Persons Responsible |
| --- | --- |
| Project Manager | Mr. Ralph Laviste |
| Team Lead | Marc Venry Aton |
| Analyst | Jessie Roxas Jr. |
| Developer/s | Marc Venry Aton  Jessie Roxas Jr.  Edwin Langga Jr.  Krizette Genn Quilaton |
| Tester | Krizette Genn Quilaton |
| Deployment Team | Jessie Roxas Jr.  Edwin Lannga Jr. |

[Table 4.3.2] Role Allocation

# Managerial process plans

## Start-up plan

### Estimation plan

Team LOOPV2.0will use MS Office Project 2007 to estimate the cost and schedule of the development of this system. The basis for setting the schedules for the milestones will be based on the schedule of submission of deliverables as stated in *table 1.1*. Should there be any changes or backlogs the schedule will be updated as needed.

### Staffing plan

The table below shows the main individuals involved in the Learning Object Organizer Plus (LOOP) project.

All individuals mentioned below will be affiliated to the project for the whole duration of the project.

|  |  |
| --- | --- |
| Name | Affiliation to project |
| Marc Venry Aton | IT Student/Team Lead |
| Edwin Langga Jr. | IT Student/Team member |
| Jessie Roxas Jr. | IT Student/Team member |
| Krizette Genn Quilaton | IT Student/Team member |
| Mr. Ralph Laviste | Project Supervisor |

[Table 5.1.2] Staff Plan

### 5.1.3. Resource acquisition plan

The team will have access to school facilities that can aid in the acquisition of necessary software such as MS Office Project 2007 and Astah Community. The team will also be able to consult the Project Supervisor and the staff of the CIT-U eLearning center during the project development.

The team will be using personal laptops and other hardware for the project.

The team will be using open source technologies to ensure the tools needed for development will be available for download on the internet.

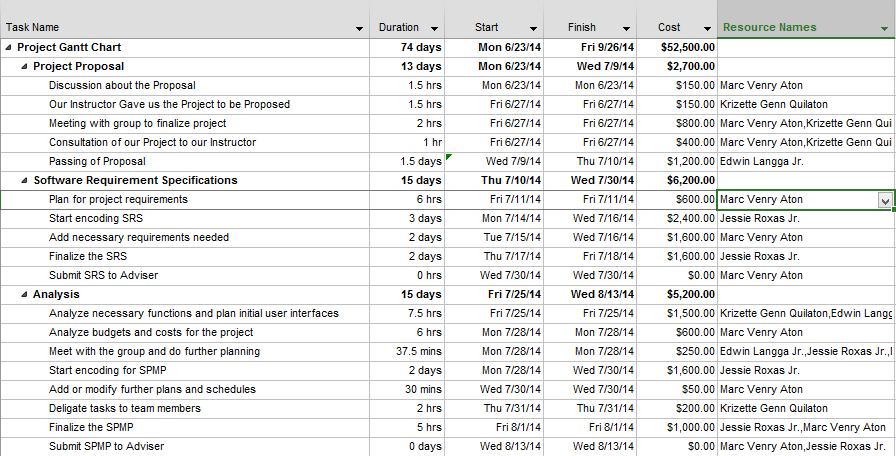
Furthermore, the team will consult will the other subsystems of the Learning Objects Project in order to develop the necessary interfaces.

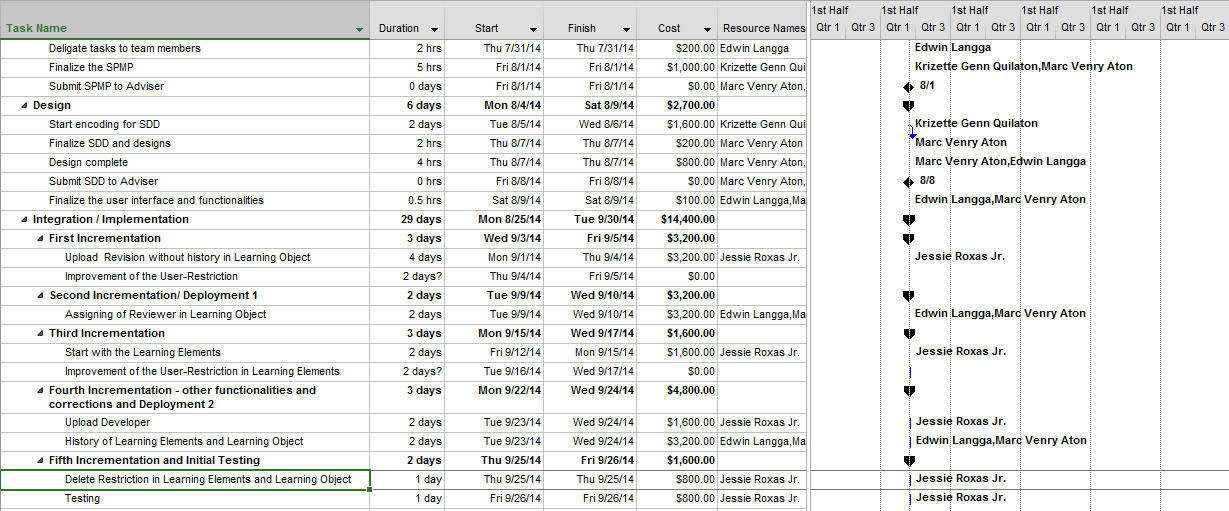
### 5.1.4. Project staff training plan

Team LOOP will not have training for disciplines such as Javascript, HTML, PHP or any other necessary technology. The members of the team already have ample knowledge on each of their assigned disciplines. Further knowledge on any new necessary technologies will be assigned for individual research to the corresponding team member/s or will be subject for consultation with the project supervisor and other authorities.

## Work plan

### Work activities





[Figure 3] Work Activities.

### Schedule allocation

The planned duration of this project will be from the project initiation on July 9, 2014 to October 15, 2014.

Target dates for final deliverables will be in the middle of June 16, 2014 to October 15, 2014.

### 5.2.3. Resource allocation

All team members of this project are students of the CCS417 course and will therefore be able to work on their respective tasks on their own time preference, aside from following the number of hours specified in the 5.2.1 Work Activities. An estimate of 4 hours maximum per weekday is allocated for personal time management inside and outside of class.

Therefore, the estimated maximum workload of each team member will be 4 hours per day (not including weekends). The weekends will be used to cover for backlog during the weekdays.

### 5.2.4. Budget allocation

|  |  |  |  |
| --- | --- | --- | --- |
| **Resource Name** | **Rate** | **Hours** | **Amount** |
| Management | $0 | 49 | - |
| Team Lead | $100.00/hour | 45 | $4,500.00 |
| Analyst | $100.00/hour | 15 | $1,500.00 |
| Developer | $100.00/hour | 116 | $11,600.00 |
| Testers | $100.00/hour | 16 | $1,600.00 |
| Deployment Team | $100.00/hour | 20 | $2000.00 |
| Miscellaneous Expense | $1,000.00 | - | $1,000.00 |
| Adjustment | $600.00 | - | $600.00 |
|  |  | **TOTAL** | $22,800.00 |

[Table 5.2.4] Budget Allocation

## 5.3. Control plan

This section will specify the metrics, reporting mechanisms, and control procedures necessary to measure, report, and control the product requirements, the project schedule and resources, and the quality of the work processes and work products.

### 5.3.1. Requirements control plan

The requirements will be specified and controlled by the use case diagrams found on the Software Requirements Specification document (SRS). To ensure consistency and accuracy, all the changes will be reflected on an updated version of the SRS. All versions of the SRS will be documented and kept for future reference and shall be useful in the Traceability Analysis mentioned at section 7.2 (Verification and validation plan). The requirements control plan is necessary in every project so the team really keeps it as a backbone for the further developments that will be done and conducted to LOOP.

### 5.3.2. Schedule control plan

The Team LOOP’s “Team Lead” will monitor the work progress of the project through weekly status reports to be documented by the team’s documentation officer. Any backlog will be reported to the team lead and adjustments to the work schedule will be applied accordingly to meet deliverable deadlines. Also, the current progress of each module shall be recorded to make sure that each week is a productive week for the project development.

### 5.3.3. Budget control plan

Any relevant expenses made by the team for the duration of this project will be recorded by the documentations officer and will be subject for review by the team lead and the project manager to ensure proper control of budgeted funds. The documentations officer shall keep a keen eye over the expenses.

### 5.3.4. Quality control plan

There will be constant supervision from the project manager to ensure the quality of each deliverable. Furthermore, the team will consult with the different development teams in charge of the other related learning object systems to maintain a balanced level of quality control throughout the main learning objects project. The team will constantly meet with the other learning object systems whenever there’s a need to discuss something – requirement changes, and functionality changes can be factors or possible reasons for the team meetings.

### 5.3.5. Reporting plan

#### Internal Reporting

Team members will submit a weekly report in Excel format that will serve as the form containing information on the work accomplished each week which will be checked by the team lead and documented by the documentations officer.

#### External Reporting

On every milestone, there will be a short presentation/discussion between members of the team and the Project Manager. Corrections/suggestions will be given if any. The suggestion will then be considered and will be discussed by the members within the team to apply to the project.

### 5.3.6. Metrics collection plan

Using the information collected from the internal reports as stated in section 5.3.5 of this document, project metrics will be analyzed and documented accordingly. The analyzed information from the gathered data will be used in the documents and will be useful in determining other possible factors that may affect the programs quality as well as the schedule when the deliverables should be submitted.

### 5.3.7 Risk management plan

Risks will be identified at the beginning of each phase and the documentations officer will assemble them into a prioritized risks list. The team will then formulate a contingency plan for each risk. Risks will be dealt with according to priority.

Some risks can only be indentified only after each implementation of a certain phase. In this case, the risks are documented and will be considered as one of the pre-identified risk. These risks will be included in the risks list and then will be dealt with according to priority. Each of these risks are likely to be solved as soon as possible as to stop chronic problems later on.

### 5.3.8 Project closeout plan

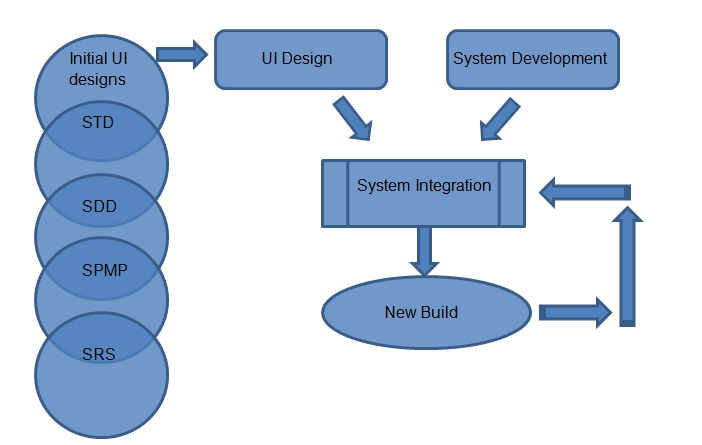
Team LOOP will ensure the proper closeout of the project in March 20, 2014 at the latest. All project documentations and source codes will be handed down to the maintenance staff of the CIT-U eLearning Staff.

6. Technical process plans

## 6.1 Process Model

Team LOOP will make use of the Microsoft Synchronize and Stabilize method for the implementation phase. Preceding phases will use the Agile Fountain Model. During requirements specification and design phases there will be a sequential order of activities that may overlap due to revisions or backlogs.

For the Implementation phase, the development team will be divided into 3 specialized work areas: UI Design, System Development, and System Integration. Team members assigned to each area will work in parallel for each build which would be tested and delivered to the client for verification.



*[Figure 4] Process Model*

## 6.2 Methods, tools, and techniques

The methods and techniques listed in this table will be evaluated and applied in specific areas of the project as appropriate:

|  |  |
| --- | --- |
| **Category** | **Methods and Techniques** |
| Requirements Elicitation | Meetings  Interviews  Brainstorming  Consultation |
| Formal Specification and Analysis | Formal models using UML to model structural aspects of the requirements and design.  Use cases to define requirements.  UI drafts to clarify client needs. |
| Prototype | Latest stabilized build to be submitted for verification. |
| Estimation | Cost per man-hours method. |

[Table 6.2.1] Methods and Techniques

|  |  |
| --- | --- |
| **Category** | **Tools** |
| Operating System | Windows XP, Windows 7 or higher |
| Development languages and databases | Databases: MySQL  Language: Javascript,JQuery,HTML5,CSS3,PHP, AJAX |
| Development Framework | Bootstrap, CodeIgniter |
| IDE | Sublime Text 2, Netbeans, Textpad |
| Document | All documents will be written using Microsoft Word, UML components will be constructed using Astah Community |
| Project Planning and Tracking | MS Office Project 2007 |

[Table 6.3.1] Methods and Techniques

## 6.3 Infrastructure Plan

This section specifies the plan for establishing and maintaining the development environment (hardware, operating system, network, and software), and the policies, procedures, standards, and facilities required to conduct the software project.

The hardware specifications required for establishing and maintenance of the development environment can be any PC that has basic functionalities and input-output devices. The PCs are running in Windows XP/7/8 and can support the common web browsers (i.e. Google Chrome, Mozilla Firefox, Safari, etc.). The team will also make use of internet as the general source of information in terms of research and in gathering the additional data to come up with a better solution to the current problem being focused in this software project. The team will make use of the text-editor called Sublime Text 3 mainly for the development process and testing procedures later on. Whenever it is necessary, the team will also make use of various software tools like Netbeans and Astah Community for the diagrams and use cases.

The team observes basic policy, procedures, and standards to make sure the quality of the software project. An example of this is to submit the deliverables on time - complete and with the right information or output. The structure of development for this project will consist of almost isolate work areas that communicate only when passing finished work to the team lead for integration. The team lead is in charge of keeping the finished modules or he may assign someone in the team to make a back-up of the project modules to an online file storage/hosting server (www.Mediafire.com).

As for the communication, the team uses the "group feature" of Facebook (a social networking site) as the medium for the team to post announcements or upload files related or necessary for the project development. The team may also send files or documents via emails. The team lead ensures that each member is aware of any updates made on a certain module or document especially during the development process.

## 6.3 Product Acceptance Plan

The project manager will inspect each deliverable on every milestone of the project. Any corrections or errors will be applied accordingly if any. Once approved by the project manager, a document deliverable will be documented as final and a software deliverable will be ready for scenario testing. The documentation of this project will be accepted once it is fully tested and approved by the project manager.

The acceptability of the product and the project as a whole depends when all the requirements are met. Every item specified in the documentation serves as the contract between the client and the developers of the software project. Each item should be implemented with the appropriate functions.

The client signing the appropriate document accepts every document of the project formally. At the end of the project development, the product will be presented to the client representative, in the person of Mr. Ralph Laviste, to demonstrate and showcase its functionalities. The client will then get to use and experience the software and may test it based on the purpose of the software project as stated in the documentations. This may determine if the client will request for changes or improvements regarding the software application.

7. Supporting process plans

## 7.1. Configuration management plan

This section contains the configuration management plan for the software project, to include the methods that will be used to provide configuration identification, control, status accounting, evaluation, and release management.

The baselines will be based on the projects specified milestones and each build from the implementation phase. Testers of Team LOOP will identify any defect or suggest any improvements of each build apart from the results of consultation to the project manager.

Changes in any aspect of the project must be made within the period/phase for which the change is to take place and is to be reflected in the corresponding documents or project builds. E.g. Changes in requirements specifications will be reflected on an updated SRS document.

## 7.2. Verification and validation plan

The members of Team LOOP will conduct preliminary tests on each new method/function completed during implementation. The Team Lead will then build and synchronize the new methods into the next build and tests it as a whole. The new build will then be presented to the Project Manager for finalization and verification.

Several tasks collectively make up continuing activities that go across the different life cycle phases. The general activities are traceability analysis, evaluation, interface analysis, and testing. These activities are horizontal threads that tie the subsequent phase activities together and allow verification to be more effectively conducted.

**Traceability analysis**

In LOOP, the traceability helps to indentify the affecting factors for the outcome of the software. It will provide a good record of the development of the project, module by module. This will also be reflected at the “change history” of the document as well as the whole documentation itself for any changes done in the software in its development process will be included as well. The team must ensure to record and report whatever changes done in the software.

**Evaluation**

In LOOP, evaluations are performed by many persons across all phases or in doing the actual builds and/or modules, on both interim and final software products, and may be either a comprehensive or selective assessment of a system. This helps the members as well as the project manager to ascertain that LOOP meets its specifications. The evaluation process is kind of similar to the strategy the Team LOOP will be taking for the Traceability Analysis. The team needs to keep track of all factors that may affect the software in its development stage later on. The team lead may ask someone in the team to prepare a form in which the member working on a specified module can record the things that he/she has been dealing with in the said module respectively.

**Interface analysis**

To ensure the completeness, accuracy, and consistencies of the interfaces, the team should do well in keeping all the interfaces updated as soon as possible every time there’s to be revised in the system. The interfaces are dynamic and are inevitably subject to change as the software development progresses. So the team’s strategy to ensure consistency is to update the interfaces immediately. Each member will be assigned something to work on with by the team lead. To ensure the accuracy, the team leader and the document officer will monitor the deliverables and carefully check them before any submission or archiving.

In the context of software verification and validation, testing can be defined as the testing that is performed. These objectives may differ from those of the developer. Testing is performed at several points in the development phase, starting from the requirement phase up to the test phase. The various test activities are listed below:

**Component Testing**

Testing conducted to verify the implementation of the design for one software elements or a collection of software elements.

**Integrating Testing**

An orderly progression of testing in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated.

**System Testing**

It is the process of testing an integrated hardware and software system to verify that the software product meets its specified requirements.

**Acceptance Test**

Formal testing conducted to determine whether or not a software product of this project satisfies its acceptance criteria and to enable the client to determine whether or not to accept the software.

The test activities mentioned are standard but are all applicable to LOOP’s development and testing stage. The team considers the test activities valid for the LOOP project.

## 7.3. Documentation plan

There are a number of documents that will be produced during the lifetime of the project. All documents are placed under the responsibility of the documentation officer. The list of all documents that will be created and maintained under version control include:

- Software Requirements Specification (SRS)

- Software Project Management Plan (SPMP)

- Software Design Description (SDD)

- Software Testing Document (STD)

- Weekly status reports (to check the current standing of a document or a process)

- System/Code documentation

- Risk Analysis Reports

- Liquidation Reports for Miscellaneous Expenses

*Note: Not all listed here are to be submitted/delivered to the client.*

## 7.4. Quality assurance plan

This section will describe the plans for assuring that the quality of delivered work products is consistent with what is expected for the project.

**Scope**

The processes used to create the following products will be tracked:

Software Requirements Specification (SRS)

Software Project Management Plan (SPMP)

Software Design Document (SDD)

Software Test Document (STD)

In addition, the quality assurance measures will be done in parallel to the verification and validation plan. (*Please refer to section 7.2*)

## 7.5. Reviews and audits

This section will describe the schedule, resources, methods and procedures used to conduct project reviews and audits. Every document in the project is subject for review and approval.

Some aspects of the system will be subject to review from other learning object system developers who have interfaces with this system. Such as the metering system, uploading system and the learning object players.

**Reviews**

Quality reviews will ensure that documentation products adhere to the standards on which they are based (as per section 7.3), and that non-documentation work products adhere to the plans/designs that are laid out by their input prerequisites.

Quality reviews of in-scope documentation work products will be conducted once the products are complete. Reviews of in-scope non-documentation work products will take place weekly during the periods that their production is active.

Each quality review will be in a meeting format and will generally require all the members of Team LOOP involved in the production of work products to attend.

A closure review will be held after all work products have been delivered. This review will be in a meeting format and will be for the purpose of gathering “lessons learned”, and identifying process improvement opportunities.

**Audits**

Brief, informal functional audits of in-scope work products will be held during the software testing and integration phases and findings will be documented.

Physical audits of software source code will be performed in order to assure that a minimum level of documentation quality exists. In addition, a quantity (% of documentation to code) will be taken to provide an indicator as to whether there is sufficient internal documentation being written.

Scheduled audits of other work products will not be held. However, audits may be performed at the request of the project manager or team lead. This is usually done to verify adherence of procedures described in the other project plans (i.e. SDD, STD, etc.)

## 7.6. Problem resolution plan

This section specifies the resources, methods, tools, techniques, and procedures to be used in reporting, analyzing, prioritizing, and processing software problem reports generated during the project. It also indicates the roles of development, configuration management, the change control board, and verification and validation in problem resolution work activities.

The project work schedule is designed with extra hours allocated for handling any problems that may arise during any of the phases of the project. Backlogs and bugs in the synchronized builds are already considered in the WBS.

The project's problem resolution plan includes the following items stated below:

**Problem reporting**

All problems must be reported to the project manager using the problem reporting form designated for use on the project. When complete, the form or report should be submitted electronically, via e-mail or may be uploaded on the “Software Project Group” in Facebook.

**Problem analysis**

Reported problems will be analyzed to determine the risk they pose to the project, and the short and long-term impact they will have on project resources, schedule, and budget.

Depending on the nature and reach of the problem, the appropriate team members will be engaged to properly analyze the problem, determine resolution steps, and estimate time required to resolve the problem. As time is more important than budget or resources on this project, emphasis will be on determining the problem’s impact on project schedule. This must include an analysis of the impact of diverting resource attention away from planned project activities toward resolving problems.

**Problem prioritizing**

Based on analysis of the problems, and given that time is the most important factor on this project, the problems will be prioritized based on the extent of their impact to schedule if they are allowed to persist. The problems will be classified as follows:

• **Critical (highest priority)**: problem will impact and/or has impacted delivery time of activities on the critical path

• **High**: problem has impacted and continues to impact delivery time of activities not on the critical path; will affect critical path if not resolved

• **Medium**: problem has an ongoing impact to schedule but is not expected to affect critical path

• **Low (lowest priority)**: problem has/had a one-time impact, and/or is so minor that critical path will never be affected

The effects of the problems that are encountered must be determined whether it will affect the critical path in order to identify which problem must be solved first, hence, to avoid delays of the entire project.

## 7.7. Subcontractor management plan

Though there are many related subsystems that interface with this one, our team will not need any subcontractors for this project.

## 7.8. Process improvement plan

This section includes plans for periodically assessing the project, determining areas for improvement, and implementing improvement plans.

**Problem resolution input**

Project process improvements are those that result from the problem resolution efforts described in section 7.6. The QA (Quality Assurance) will work with the project manager and other key resources directly involved with the process in question to develop changes to the problematic process IF a root cause analysis is performed and justified process improvements are identified.

If an organizational process is at fault, a temporary workaround that will be devised by the same participants which will last for the duration of the project. The problems and temporary workarounds will be documented so that the organizational process that caused the problem can be inspected to determine whether the changes used in this project may be of benefit to the organizational process. If the workaround was only of particular application to the current project, the documentation will be stored in the project repository so that future project managers will be aware of changes required in the process for projects of a similar nature in future. A full software back-up may also be done to ensure there's no loss of data and the development can continue.

The team lead will use the weekly status reports to assess the productivity or efficiency of the team members. These reports will be the basis for process improvement measures that would be implemented any time needed within the project development.

## 8.0 Additional Plans

There are no additional plans.