**The Learning Through Activities System**

Project Management Plan

**By**

Miss. Jirada Somanasak 542115009

Miss. Putsacha Owatsakul 542115037

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

**Project Advisor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aj. Chartchai Doungsa-ard**

**Document History**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Version** | **Status** | **Date** | **Viewable** | | **Editable** | **Responsible** | |
| **Documents** | | | | | | | | |
| **MOCK-Project Management Plan\_V.0.1. docx** | **0.1**   * Introduction * Infrastructure * Management Procedures * Quality Standard * Quality Planning * Schedule and Milestones * Software Configuration Management * Risk Management | Pre-  Released | 10-05-13 | | Prutya, Warat, Aj.Chartchai | Prutya, Warat, | | Prutya, Warat, |
| **MOCK-Project Management Plan\_V.1.0. docx** | **1.0**   * Review Document | Released | 05-06-13 | | Prutya, Warat, Aj.Chartchai | Prutya, Warat, | | Prutya, Warat, |
| **MOCK-Project Management Plan\_V.1.1. docx** | **1.1**   * Add more information in Identification topic. * Change schedule and milestone. | Released | 21-06-13 | | Prutya, Warat, Aj.Chartchai | Prutya, Warat, | | Prutya, Warat, |

**Table of Contents**

[Chapter One | Introduction 4](#__RefHeading__2377_2086945100)

[1.1 Identification 4](#__RefHeading__2379_2086945100)

[1.4 Work Products to be Develop 5](#__RefHeading__2383_2086945100)

[1.5 Acronyms and Definitions 5](#__RefHeading__2385_2086945100)

[Chapter Two | Infrastructure 6](#__RefHeading__2387_2086945100)

[2.1 Software Development Life Cycle 6](#__RefHeading__2389_2086945100)

[2.2 System Architecture 7](#__RefHeading__2391_2086945100)

[2.3 Development Tools 8](#__RefHeading__2393_2086945100)

[2.4 Hardware and Material Resources 9](#__RefHeading__2395_2086945100)

[Chapter Three | Management Procedures 10](#__RefHeading__2397_2086945100)

[3.1 Project Team Structure 10](#__RefHeading__2399_2086945100)

[3.2 Monitoring and Controlling Mechanisms 10](#__RefHeading__2401_2086945100)

[Chapter Four | Quality Standard 11](#__RefHeading__2403_2086945100)

[4.1 ISO 29110 for Very Small Entity (VSE) 11](#__RefHeading__2405_2086945100)

[Chapter Five | Quality Planning 15](#__RefHeading__2407_2086945100)

[5.1 Quality Factors 15](#__RefHeading__1778_1109928140)

[5.1 Reviews/Responsibility 16](#__RefHeading__2409_2086945100)

[5.2 Testing 16](#__RefHeading__2411_2086945100)

[Chapter Six | Schedule and Milestones 17](#__RefHeading__2413_2086945100)

[Chapter Seven | Software Configuration Management 17](#__RefHeading__2415_2086945100)

[7.1 Software Configuration Management 17](#__RefHeading__2417_2086945100)

[7.2 Filename Format 17](#__RefHeading__2419_2086945100)

[7.3 Change Management 17](#__RefHeading__2421_2086945100)

[7.4 Software Configuration Item Table 18](#__RefHeading__2423_2086945100)

[Chapter Eight | Risk Management 19](#__RefHeading__2425_2086945100)

[8.1 Risk Management Process 19](#__RefHeading__2427_2086945100)

[8.2 Risk Identification and Solutions 19](#__RefHeading__2429_2086945100)

# Chapter One | Introduction

## 1.1 Identification

This Project Management Plan is the document for planning, scheduling activities and evaluating overall of the project so that the project will complete as successfully as possible in spite of all the risks. The Project Management will lead us to see specific project reach fruition and allow us to work with it and see a project through from start to finish.

**1.2 Project Overview**

The learning through activities system support the activity courses management in Chiang Mai University. The system is proposed to helps students plan about participation the activities. Also, it can help students analyze their situation and help instructors provide the grade for each student easily. The application run on web application and it use Facebook account to connect with website. The learning through activities system make efficiency for university and make usability for users.

**1.2.1 Purpose & Scope**

The learning through activities system is the web application. It is focus on instructors and students. For instructors, The system provides students information and activities participation overview of each student. For students, the system provides activities information and grade information. So, the learning through activities system can help students and instructors plan to achieve the course.

The learning through activities system has five scope to management, which are including these scope:

* User management
* Activities management
* Participate histories management
* Grade management
* Activity transcript management

**1.3 Document Overview**

The purpose of the Learning Through Activities system Project Management Plan is to guide project team members during the development of the Learning Through Activities system project.

In this document provides plan of progress 1, progress 2 and progress 3. For progress 1, the document consist of user management and activity management. For progress 2, the document consist of participate histories management and grade management. For progress 3, the document consist of activity transcript management.

## 1.4 Work Products to be Develop

**1.4.1 Deliverable**

| **No.** | **Deliverables/Release** | **Media** | **Copies** | **Date** |
| --- | --- | --- | --- | --- |
| 1 | **Project Proposal**   * Proposal Version 1.0 | Document | 4 | 7 th Mar 2014 |
| 2 | **Progress Report I**   * Project Management Plan Version 1.0 * Software Requirement Specification Version 1.0 * Software Design Document Version 1.0 * Test Plan Version 1.0 * Traceability Record Version 1.0 * Software Version 1.0 | Document  Document  Document  Document  Document  Source Code | 4  4  4  4  4  1 | 28th May 2013 |
| 3 | **Progress Report II**   * Project Management Plan Version 2.0 * Software Requirement Specification Version 2.0 * Software Design Document Version 2.0 * Test Plan Version 2.0 * Traceability Record Version 2.0 * Software Version 2.0 | Document  Document  Document  Document  Document  Source Code | 4  4  4  4  4  1 | 17h Jul 2013 |
| 4 | **Show Pro Event**   * Software Version 2.0 * 30 seconds Video * Poster size A1 | Files  File  Poster | 1  1  1 | 9th Sep 2013 |
| 5  5 | **Final Progress**   * Project Management Plan Version 3.0 * Software Requirement Specification Version 3.0 * Software Design Document Version 3.0 * Test Plan Version 3.0 * Traceability Record Version 3.0 * Software Version 3.0 * Software Source Code | Document  Document  Document  Document  Document  Source Code  CD-ROM | 4  4  4  4  4  1  1 | 11th Oct 2013  11th Oct 2013 |

## 1.5 Acronyms and Definitions

**1.5.1 Acronyms**

SRS Software Requirement Specification

SDD Software Design Document

OS Operating System

VSE Very Small Entity

PM Project Management

SI Software Implementation

IID Iterative and Incremental Development

SCI Software Configuration Item

**1.5.2 Definitions**

|  |  |
| --- | --- |
| Acceptance test | Test activities for sample checks to verify that a system (or product, solution) has the right quality for deployment or usage. Often acceptance test is done by the customer. [IEEE90] |
| Feature | Transformation of input parameters to output parameters based on a specified algorithm. It describes the functionality of a product in the language of the product. Used for requirements analysis, design, coding, testing or maintenance. [IEEE90] |
| IEEE | Institute for Electrical and Electronics Engineers. Biggest global interest group for engineers of different branches and for computer scientists. [IEEE90] |
| Plan | A documented series of tasks requires meeting an objective, typically including the associated schedule, budget, resources, organizational description and work breakdown structure. [IEEE90] |
| Project Management | The application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project. [IEEE90] |
| Project Plan | A formal, approved document used to guide both project execution and project control. The primary uses of the project plan are to document planning assumptions and decision, to facilitate communication among stakeholders, and to document approved scope, cost, and schedule baseline. [IEEE90] |
| Risk | An uncertain event or condition that, if it occurs, has a positive or negative effect on a project’s objectives. It is a function of the probability of occurrence of a given threat’s occurrence. [IEEE90] |
| Risk Management | The systematic application of management policies, procedures and practices to the tasks of identifying, analyzing, evaluating, treating and monitoring risk. [IEEE90] |
| Traceability | The ability to trace the history, application or location of an item or activity, or work products or activities, by means of recorded identification. The establishment and maintenance of relationships between such items. Horizontal traceability describes the relationship between work products of same type (e.g., customer requirements). Vertical traceability describes the relationship between work products which build upon each other or are derived from each other (e.g., from customer requirements to qualification test cases). Bidirectional traceability allows to directly following relationships in both directions. [IEEE90] |
| Unit test | A test of individual programs or modules in order to remove design or programming errors. [IEEE90] |

# Chapter Two | Infrastructure

## 2.1 Software Development Life Cycle

The Iterative model approach is iterating on steps as the project progress with requirements. Iterative model iterates Requirement, Design, Build, and Test phases again and again for each requirement and builds a system iteratively until complete. The advantage of iterative model is building and improving the product step by step. Thus, developer can track the defects at early stages and avoid flow of defects.

![](data:None;base64,)

**Figure 1:** Iterative and Incremental Development Model

## 2.2 System Architecture

**Figure 1:** The overview architecture of Mock Object Code generation

**Class Analysis:** This part will use Java Reflection to inspect input code, then analysis the input code to Test Class objects. If users input wrong syntax of java code, the program will not allow users to test. Reflection uses a special kind of java class: Class. The object of the Class type can hold all information of the class and have getter method extract the information.

**Test Configuration:** When users create one test method, program will allow users to configure the behavior of mock object and the environment via GUI. The Program provides step-by-step input form, which helps users to understanding the configuration. Test Configuration module contains 4 phases. First one is creating a mock up data or the test input data. In the case of method that’s required non-primitive type, users have to instantiate that object here before using it in another phase. Next, the phase is stubbing the method. This phase users have to specify the method if necessary (in the case of non-void method) based on Mockito’s function that discussed before. Then the program is on the running phase. This phase is telling the mock object to perform the method that we stubbed before. Finally, users have to verify that behavior for confirming that method has been used correctly.

**Mockito:** This is the based mock framework part of this program. The method of configuration behavior in Test Configuration will fetch from this part.

**JUnit:** This is pattern of the test class, which uses for build pattern code in final test output. JUnit support Mockito framework, both frameworks work together in the way to make Mock Object unit test.

**JIBX XML Gen:** This module uses to generate XML metadata for holding information of Test Configuration in the form of Test Class object from Class analysis. JiBX read information then generates XML, which to be use in another iterations. After create metadata, this module will transform it to the java code. The transform part will convert information from XML with binding definition documents allow program convert XML back to code object. Then generate the code object to the .java JUnit test code.

**Test Code**: This is a final result of this program. Users will get the well-formed code and be able to test when users import the result file to their work in IDE. This code will serve with both Mockito framework and JUnit framework

## 2.3 Development Tools

* IntelIJ IDE 12.0.1
* NetBeans IDE 7.3
* Visual Paradigm UML 10.0
* MySQL Query Browser
* Apache Tomcat 7

## 2.4 Hardware and Material Resources

* Computers

**- MacBook Pro 13-inch**

**Processor:** 2.5GHz quad-core Intel Core i5

Turbo Boost up to 3.1GHz

**Memory:** 4 GB 1600MHz DDR3

**Graphics:** Intel HD Graphics 4000 1024 MB

**Operating System:** OS X 10.9.2(13C1021)

- **MacBook Pro 13-inch (ของเช้ง)**

**Processor:** 2.5GHz dual-core Intel Core i5

Turbo Boost up to 3.1GHz

**Memory:** 4 GB 1600MHz DDR3

**Graphics:** Intel HD Graphics 4000 512 MB

**Operating System:** OS X 10.8.3 (12D78)

# Chapter Three | Management Procedures

## 3.1 Project Team Structure

|  |  |
| --- | --- |
| **Participants** | **Activities** |
| Miss. Jirada Somanasak  and  Miss. Putsacha Owatsakul | Feasibility Study |
| Project Proposal |
| Project Requirements |
| Project Plan |
| Project Design |
| Implementation |
| Testing |

## 3.2 Monitoring and Controlling Mechanisms

**3.2.1 Project Meeting**

|  |  |
| --- | --- |
| **Participants** | **Roles** |
| Miss. Jirada Somanasak | Development team member |
| Miss. Putsacha Owatsakul | Development team member |
| Aj. Chartchai Doungsa-ard | Project advisor |

# Chapter Four | Quality Standard

## 4.1 ISO 29110 for Very Small Entity (VSE)

ISO 29110 is the Software Life Cycle Profiles and Guidelines for Very Small Entities (VSEs) standards and technical reports are targeted at Very Small Entities (VSEs). A Very Small Entity (VSE) is an enterprise, organization, department or project having up to 25 people. ISO 29110 concerns on project management process and software implementation process.

**4.1.1 Project Management (PM) process**

* **Purpose**

The purpose of the Project Management process is to establish and carry out in a systematic way the tasks of the software implementation project, which allows complying with the project’s objectives in the expected quality, time and costs.

* **Objectives**
* **PM.O1.** The *Project Plan* for the execution of the project is developed according to the *Statement of Work* and validated with the Customer. The tasks and resources necessary to complete the work are sized and estimated.

**- PM.O1. Tasks in this project:**

1. Create the Project Plan related with the Project Proposal.

* **PM.O2.** Progress of the project is monitored against the *Project Plan* and recorded in the *Progress Status Record.* Corrections to remediate problems and deviations from the plan are taken when project targets are not achieved. Appropriate treatment is taken to correct or avoid the impact of risk. Closure of the project is performed to get the Customer acceptance documented in the *Acceptance Record.*

**- PM.O2. Tasks in this project:**

1. Record the project status in Project Status Record for each progress.

2. Establish the Acceptance Record before submitting final progress.

* **PM.O3.** The *Change Requests* are addressed through their reception and analysis. Changes to software requirements are evaluated for cost, schedule and technical impact.

**- PM.O3. Tasks in this project:**

1. Analyzing the change.

2. Setting the change request form.

3. Approving the change request by project advisor.

4. Change the project follow by approved change request.

* **PM.O4.** Review meetings with the Work Team and the Customer are held. Agreements are registered and tracked.

**- PM.O4. Tasks in this project:**

1. Meeting with team members and project advisor.

2. Evaluate meeting results.

* **PM.O5.***Risks* are identified as they develop and during the conduct of the project.

**- PM.O5. Tasks in this project:**

1. Identify the risks.

2. Analyse the risks.

3. Plan for managing the risksin the Project Plan.

* **PM.O6.**A *Software Version Control Strategy* is developed. Items of *Software Configuration* are identified, defined and baselined. Modifications and releases of the items are controlled and made available to the Customer and Work Team including the storage, handling and delivery of the items.

**- PM.O6. Tasks in this project:**

1. Identify SCI.

2. Create SCI table.

3. Record the change of each SCI in the SCI table.

* **PM.O7.** Software Quality Assurance is performed to provide assurance that work products and processes comply with the *Project Plan* and *Requirements Specification*.

**- PM.O7. Tasks in this project:**

1. Create tasks follow ISO29110 for VSE to the Project Plan and Requirements Specification.

**4.1.2 Software Implementation (SI) process**

* **Purpose**

The purpose of the Software Implementation process is the systematic performance of the analysis, design, construction, integration and tests activities for new or modified software products according to the specified requirements.

* **Objectives**
* **SI.O1.** Tasks of the activities are performed through the accomplishment of the current *Project Plan*.

**- SI.O1. Tasks in this project:**

1. Develop software comply with the current Project Plan.

* **SI.O2.** Software requirements are defined, analysed for correctness and testability, approved by the Customer, baselined and communicated.

**- SI.O2. Tasks in this project:**

1. Analyse the requirements.

2. Accomplish the Software Requirements Specification.

* **SI.O3.** Software architectural and detailed design is developed and baselined. It describes the software items and internal and external interfaces of them. Consistency and traceability to software requirements are established.

**- SI.O3. Tasks in this project:**

1. Create Software Design Document that covers all of Software Requirements.

2. Create Traceability Record to trace the items in Software Design Document with the software requirements.

* **SI.O4.** Software components defined by the design are produced. Unit test are defined and performed to verify the consistency with requirements and the design. Traceability to the requirements and design are established.

**- SI.O4. Tasks in this project:**

1. Create Unit test that is comply with requirements and design after software components are produced.

2. Perform the unit test.

3. Traceability record is created for tracing Unit test with the requirements and design

* **SI.O5.***Software* is produced performing integration of software components and verified using *Test Cases and Test Procedures*. Results are recorded at the *Test Report*. Defects are corrected and consistency and traceability to *Software Design* are established.

**- SI.O5. Tasks in this project:**

1. Design Test Cases from Software Design.

2. Test the software components.

3. Record the Test Cases results at the Test Report.

4. Create traceability record.

* **SI.O6.** A *Software Configuration*, that meets the *Requirements Specification* as agreed to with the Customer, which includes user, operation and maintenance documentations is integrated, baselined and stored at the *Project Repository*. Needs for changes to the *Software Configuration* are detected and related *Change Requests* are initiated.

**- SI.O6. Tasks in this project:**

1. Analyze the change.

2. Create the change request form.

3. Approve the change request by project advisor and upload in our repository that is Dropbox.com.

4. Change the project complies with approved change request.

* **SI.O7.** Verification and Validation tasks of all required work products are performed using the defined criteria to achieve consistency among output and input products in each activity. Defects are identified, and corrected; records are stored in the *Verification/Validation Results*.

**- SI.O7. Tasks in this project:**

1. All works are traceable and have tested.

# Chapter Five | Quality Planning

## 5.1 Quality Factors

|  |  |
| --- | --- |
| **User Requirement Specification** | **Related quality factor** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

## 5.1 Reviews/Responsibility

|  |  |  |  |
| --- | --- | --- | --- |
| **Stage Exit Review** | | | |
| **No.** | **Stage** | **Review Item** | **Responsibility** |
| 1 | Project Planning | Project Plan | Jirada and Putsacha |
| 2 | Requirement Specification | Software Requirement Specification | Jirada and Putsacha |
| 3 | Architecture and Detailed Design | Software Design Document | Jirada and Putsacha |
| 4 | Software Testing | Software Testing Documents | Jirada and Putsacha |
| 5 | Project Monitoring and Control | Traceability Record | Jirada and Putsacha |

## 5.2 Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Stage Exit Review** | | | |
| **No.** | **Test** | **Verification** | **Responsibility** |
| 1 | Unit Testing | Jirada, Putsacha, and Aj.Chartchai | Jirada and Putsacha |
| 2 | System Testing | Jirada, Putsacha, and Aj.Chartchai | Jirada and Putsacha |
| 3 | Acceptance Testing | Jirada, Putsacha, and Aj.Chartchai | Jirada and Putsacha |

# Chapter Six | Schedule and Milestones

**Figure 2: Schedule and Milestones**

# Chapter Seven | Software Configuration Management

## 7.1 Software Configuration Management

Software Configuration Management is a set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made. In other words, SCM is a methodology to control and manage a software development project.

## 7.2 Filename Format

For the filename format that we using for all project documents is:

[Project name]-[Document name]\_[Version].file type

## 7.3 Change Management

Change management manages all of the changes in the project during development process. All of the change requests will be record into the change request document.

We have the strategy for manage the changes by following these steps:

1. Analyzing the change.
2. Setting the change request form.
3. Approving the change request by project advisor.
4. Change the project follow by approved change request.

17

## 7.4 Software Configuration Item Table

| **No.** | **Item** | **File name** | **File Type** | **Owner** | **Path** | **Baseline version** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Project Proposal | MOCK-Project Proposal\_V.1.0 | .docx | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Proposal | 1.0, 2.0, 3.0 |
| 2 | Project Management Plan | MOCK-Project Management Plan\_V.1.0 | .docx | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Project Management Plan | 1.0, 2.0, 3.0 |
| 3 | Software Requirement Specification | MOCK-SRS\_V.1.0 | .docx | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\SRS | 1.0, 2.0, 3.0 |
| 4 | Software Design Document | MOCK-SDD\_V.1.0 | .docx | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\SDD | 1.0, 2.0, 3.0 |
| 5 | Test Plan | MOCK-Test Plan\_V.1.0 | .docx | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Test Plan | 1.0, 2.0, 3.0 |
| 6 | Traceability Record | MOCK-Traceability Record\_V.1.0 | .docx | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Traceability Record | 1.0, 2.0, 3.0 |
| 7 | SoftwareSource Code | Mock Object Code Generation\_V.1.0 | .zip | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Code | 1.0, 2.0, 3.0 |
| 8 | 30 seconds video | MOCK-Showpro Video | .avi | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Showpro | 1.0 |
| 9 | Poster size A1 | MOCK-Poster | .png | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Showpro | 1.0 |
| 10 | Software Product | Mock Object Code Generation\_V.1.0 | .zip | Prutya  and  Warat | Users\MacintoshHD\Desktop\MOCK\Code | 1.0 |

# Chapter Eight | Risk Management

Risk management is concerned with identifying risks and drawing up plans to minimize their effect on the project.  
 A risk is probability that some adverse circumstance will occur.  
 -Project risks affect schedule or resources.

-Product risks affect the quality or performance of the software being developed.

-Business risks affect the project team during developing or procuring the software.

Identified risks at the start of project and at the start of development phase. All identified risks are documented and assessed in the Risk Management Process by the Project Team. In the Risk Management Process defines the possible risks and solution of them, and who is responsible for.

## 8.1 Risk Management Process

![Screen Shot 2555-02-20 at 12](data:None;base64,)

**Figure 3:** Risk Management Process Model

1. Risk identification: identify project, product and business risks.  
 2. Risk analysis: Assess the likelihood and consequences of the risks.  
 3. Risk planning: Draw up plans to avoid or minimize the effects of the risks.  
 4. Risk monitoring: Monitor the risks throughout the project.

## 8.2 Risk Identification and Solutions

|  |  |  |
| --- | --- | --- |
| **No.** | **Risk statement** | **Risk Solution** |
| 1. | The requirements might be change. | * Meeting and discuss and do a priority of changed requirements. * Design system with changed requirements and related with the other requirements. |
| 2. | During the implementing, the Internet maybe out of order or slow. | * Change the working place. |
| 3. | The deliverables maybe delay. | * Try to study more hard than previous work. * Ask a professional to make faster understand. |
| 4. | Team member maybe get sick and can’t develop the project. | * Assign a work to left team member who doesn’t get sick. |
| 5. | Team member maybe get accident and can’t develop the project. | * Assign a work to left team member who doesn’t get accident. |
| 6. | Budget of developing may not enough. | * Ask for more budgets from project advisor. |
| 7. | Work products are not submitted on time. | * Establish the project plan. * Develop project follow the project plan. |
| 8. | Work products are not traceable. | * Create the traceability record. |
| 9. | Mockito framework can be replaced by other better/new mocking framework. | * Separate application to small modules/components. * If they need to change mocking framework, developers can implement a new configure module and a new test code module. |