|  |
| --- |
| V 1.0 | Pookas ETventure |



|  |  |
| --- | --- |
| Pookas ETventure | PROJECT PLAN |

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision Number | Date | Primary Author(s) | Comments |
| 1.0 | 4 Sep 2009 | Ivan Loh, Sarah Lam, Gillian Ng Lui Quan Fu, Goh Li-Xian | First Version |
|  |  |  |  |

Table of Contents

[1 Introduction 4](#_Toc240701203)

[1.1 Overview 4](#_Toc240701204)

[1.2 Goals and Objectives 4](#_Toc240701205)

[1.3 Scope 5](#_Toc240701206)

[2 Risk Management 6](#_Toc240701207)

[3 Team Organization 7](#_Toc240701208)

[3.1 Team Structure 7](#_Toc240701209)

[3.2 Roles and Responsibilities 7](#_Toc240701210)

[3.3 Communication 9](#_Toc240701211)

[4 Team Resources 10](#_Toc240701212)

[4.1 People 10](#_Toc240701213)

[4.2 Hardware and Software 10](#_Toc240701214)

[5 Project Schedule 11](#_Toc240701215)

[5.1 Work Breakdown Structure 11](#_Toc240701216)

[5.2 Task Durations and Dependencies 12](#_Toc240701217)

[5.3 Critical Path Analysis 13](#_Toc240701218)

[5.4 Activity Dependency Chart 14](#_Toc240701219)

[5.5 Gantt Chart 17](#_Toc240701220)

[5.6 Work Package 18](#_Toc240701221)

[6 Project Estimates 23](#_Toc240701222)

[6.1 Code Size Estimation using Function Points 23](#_Toc240701223)

[6.2 Efforts, Duration and Team Size Estimation 26](#_Toc240701224)

[6.3 Cost Estimates 30](#_Toc240701225)

[7 Process Modeling 32](#_Toc240701226)

[7.1 Lifecycle Model 32](#_Toc240701227)

[7.2 Capability Maturity Model 32](#_Toc240701228)

[7.3 Project Processes 33](#_Toc240701229)

[7 Product Checklist 38](#_Toc240701230)

[8 Best Practices Checklist 39](#_Toc240701231)

[8.1 Project Plan Checklist 39](#_Toc240701232)

[8.2 Requirement Checklist 40](#_Toc240701233)

[8.3 Architecture Design and Modeling 40](#_Toc240701234)

[8.4 Code issues 41](#_Toc240701235)

[8.5 System Testing 41](#_Toc240701236)

# 1 Introduction

## 1.1 Overview

The virtual classroom & Edventure project is a development of a virtual 3D environment where students can access classroom materials such as lecture notes or quizzes. Students are able to learn and interact with others through a platform with fascinating visual effects instead of the usual learning system. This development will involve two features packed product namely Blackboard Academic Suite and, ActiveWorld.

Edventure

Edventure is the name of the Blackboard Academic Suite that is used by Nanyang Technological University (NTU) as its learning system. Many students in academic institutions worldwide are currently using the same platform (Blackboard Academic Suite) to access lecture notes, course announcements, quizzes and discussion board.

ActiveWorld

ActiveWorld is a powerful virtual world building platform that allows user to create attractive 3D worlds that others can visit, mingle and chat in. The virtual classroom project will be built using the ActiveWorld client and Edventure will be integrated into the virtual classroom to provide functions that will enhance the way students learn.

Our team – Pookas ETventure is a five member’s team where each member possesses different skills and abilities that will allow the team to take on any challenge. The team will be implementing the virtual classroom project and the requirements specification for the project will be documented in the subsequent sections.

## 1.2 Goals and Objectives

The goals and objectives of the project are conjured as follow:

We aim to enhance students learning experience through the integration of a virtual world. We also seek to provide a platform for students to interact with each other in a multiuser environment found in ActiveWorld. This new learning platform will enhance the way academic providers’ deliver their education contents and materials.

## 1.3 Scope

The virtual classroom & Edventure project is a 3D multiuser environment built on ActiveWorld. The virtual classroom will allow student to attend lectures, participate in game quizzes and broadcast messages to other students or lecturer who are taking the same module.

The contents required in the virtual classroom will be retrieved from Edventure and will be limited to certain modules only. Students are required to login to their account before the contents can be retrieved to the virtual world.

Constraints

The project has a tight deadline and has to be completed within 9 weeks and this may not be enough to produce a fully functional system. Furthermore, all members have involvements in other projects and assignments, thus, they will not be able to fully devote their time into the project. However, we do note that each member will devote about 8 hours per week.

ActiveWorld is a relatively new system building platform for all members. The team members will have to use some time from the tight schedule to learn about the software before they could start designing on their project. Thus, reducing the time required for implementing the project.

# 2 Risk Management

The following risks may affect the project negatively.

Lack of Full Access Rights to Edventure

|  |  |
| --- | --- |
| Impact Severity | High |
| Probability | 30% |
| Impacts | Without Full Access rights to Edventure, we are unable to determine if we could perform the tasks we require in Edventure. This may affect functions that are only possible with Administrator Account. |
| Risk Reduction | For functions that require rights greater than our Students’ account, we will approach our lecturers’ for their access rights to try out. We will also contact the department in NTU that manages Edventure, Blackboard to resolve access rights issue. |

Lack of experience with ActiveWorld

|  |  |
| --- | --- |
| Impact Severity | Low |
| Probability | 75% |
| Impacts | This could hinder the pace of development, and the structure of the program code might not be optimal |
| Risk Reduction | Start learning to use the ActiveWorld API as early as possible. This would allow more time in case program structure needed to be changed |

System size underestimated

|  |  |
| --- | --- |
| Impact Severity | Very High |
| Probability | 15% |
| Impacts | Due to not knowing how to use the ActiveWorld API, the amount of effort needed to produce the software product may be vastly underestimated. The current schedule might be overly optimistic, and more time might be needed to implement the required features. |
| Risk Reduction | Learn about the basic functionality the ActiveWorld API offers, as well as the previous projects, before deciding on the final features to implement. Monitor progress to ensure that the project is on track |

Lack of experience with project planning and documentation processes

|  |  |
| --- | --- |
| Impact Severity | Very High |
| Probability | 15% |
| Impacts | The amount of documentation required is not very clear, but if the team has to produce a large amount of documentation, it might hinder the development of the software product, and this planning might be abandoned under pressure |
| Risk Reduction | Monitor progress so that the project is on track. When planning, ensure that it is feasible to implement the requirements, given the required processes involved, and avoid requirement gold-plating |

No full-time dedicated team members

|  |  |
| --- | --- |
| Impact Severity | Medium |
| Probability | 100% |
| Impacts | The team members have educational commitments. This would definitely mean that the actual staff size is smaller |
| Risk Reduction | When defining feasible requirements, take into consideration the smaller staff size |

# 

# 3 Team Organization

## 3.1 Team Structure

Figure - Team Structure Diagram

## 3.2 Roles and Responsibilities

Project Manager

|  |  |
| --- | --- |
| Assigned to | Ivan Loh |
| Responsibilities | Managing the project plan |
| Assign tasks to team members |
| Ensure project is progressing on time |
| Update members on project status |
| Coordinate team meetings |

Lead Developer

|  |  |
| --- | --- |
| Assigned to | Lui Quan Fu |
| Responsibilities | Responsible for overall system implementation |
| Develop system modules according to the specification in the detailed design document |
| Integration of coded modules into functioning system |

Developer

|  |  |
| --- | --- |
| Assigned to | Goh Li-Xian |
| Responsibilities | Assist Lead Developer in the system implementation |
| Code system modules according to design specifications |
| Debug after testing |

****Quality Assurance Engineer****

|  |  |
| --- | --- |
| Assigned to | Sarah Lam |
| Responsibilities | Design and develop test plan |
| Apply testing strategies |
| Report bugs in system for developer to improve |

System Architect

|  |  |
| --- | --- |
| Assigned to | Gillian Ng |
| Responsibilities | Manage overall system design based on requirement specifications |
| Responsible for high level design document |
| Converts high level design document into classes for the developers |
| Refine the high-level modules to low-level modules |

Client

|  |  |
| --- | --- |
| Assigned to | Wong Ee Kian |
| Responsibilities | Provide requirements for the project |
| Involve in acceptance testing for the system |
| Provide requirements for the project |

## 3.3 Communication

Our team communicates are performed in the following ways:

* We hold weekly meetings for discussions and to distribute work package among team members.
* Online instant messaging for communicating to each other when the group is not meeting physically.
* The team uses a mailing list to send emails to the entire team. The mailing list will track each new email topic and a weekly digest will be sent to the team. Each email topic and its subsequent replies can be viewed via the mailing list web portal, allowing each member to understand each topic easily.
* Source codes and documents are checkout via a SVN installed in a web server so that every version will be able for reverting and the team can share source codes easily.

# 4 Team Resources

## 4.1 People

#### 4.1.1 Ivan Loh

Ivan Loh is a final year student pursing his degree in Nanyang Technological University (NTU) Computer Science. Specialized in Web technologies and open source software, he has developed various commercial websites for clients. He has also proposed to clients on which open source software to build their websites on based on their inputs and requirements.

#### 4.1.2 Lui Quan Fu

Lui Quan Fu is a fourth year student in Computer Science. He has extensive programming language skills in C++, VB6, Java, VBA, and also well equipped with knowledge on database system such as mySQL, Access, Oracles. Beside application based system, his ability also includes web scripting HTML, PHP, Javascript, ASP and image/video processing tools like Adobe Photoshop, Fireworks, FFmpeg.

#### 4.1.3 Gillian Ng

Gillian Ng is a final year student in NTU majoring in Computer Science. She has programming experience in Java and C++, and her software development skills were developed from a few short internship experiences.

#### 4.1.4 Goh Li-Xian

Goh Li-Xian is a fourth year student in NTU majoring in Computer Science. She has programming knowledge on C++ and Java and also has experience in using applications such as Photoshop, Flash and 3D Max.

#### 4.1.5 Sarah Lam

Sarah Lam is currently a final year student studying Computer Science in NTU. She has experience in developing web applications and is proficient in web languages such as PHP, Javascript and JSP. She also has knowledge in programming languages such as C++ and Java.

## 4.2 Hardware and Software

The team will be using 5 personal laptops for all documentation works. The laptops are either running on Windows XP or Vista. The applications required for creating the documentations are Microsoft Word, Visual Paradigm and Microsoft Project.

The implementation of the virtual world will be done in the computers available in the Software Engineering Lab (N4-B1C-14) in School of Computer Engineering. The computers are mostly running on Windows XP. The Active World Virtual Environment is installed in the lab computers and will be required for building the virtual classroom project.

# 5 Project Schedule

## 5.1 Work Breakdown Structure

A large, complex project is organized and comprehended by breaking them into progressively smaller pieces until they are a collection of well defined tasks. The Work Breakdown Structure (WBS) is used to provide the framework for organizing and managing the work among the rest of our project team.

The diagram below shows the first tier being broken down into 5 different phases.

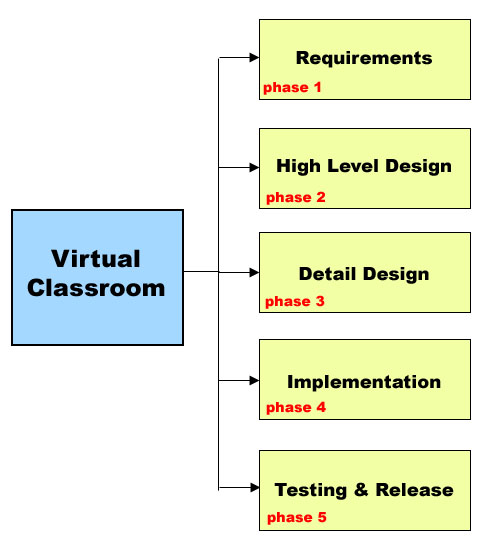


Figure - Work Breakdown Structure Diagram

The WBS will live on throughout the project, in the project schedule and often is the main path for reporting project costs.

## 5.2 Task Durations and Dependencies

The project is further breakdown into the 2nd and 3rd tier, each task showing the estimated time, efforts and resources required to complete.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phase** | **Node** | **Activity** | **Duration** | **Dependencies** |
| Phase 1 | 2 | Brainstorming Ideas | 1 | None |
| 3 | Business Planning | 3 | 2 |
| 4 | Write Proposal | 3 | 2 |
| 5 | Budget And Approval | 1 | 4 |
| Phase 2 | 7 | Use Case Diagram | 2 | Phase 1 |
| 8 | Class Diagram | 2 | 7 |
| 9 | Activity Diagram | 2 | 7 |
| 10 | Sequence Diagram | 1 | 9 |
| 11 | Activity Dependency Chart | 3 | Phase 1 |
| 12 | White Box Test Unit | 2 | 9 |
| Phase 3 | 14 | Review Specific Goal | 2 | Phase 2 |
| 15 | Evaluate Project Specification | 2 | Phase 2 |
| 16 | Indicate Data Schema | 2 | 15 |
| 17 | Prepare Programming Framework | 2 | Phase 2 |
| 18 | Define Coding Standards | 3 | 17 |
| Phase 4 | 20 | Research AW & Edventure API | 1 | Phase 3 |
| 21 | Research UI Changes | 2 | Phase 3 |
| 22 | Develop Re-Useable Process | 3 | 20;21 |
| 23 | Implement Selective Boardcast | 4 | 22 |
| 24 | Implement Quiz Games | 6 | 22 |
| 25 | Implement Video Lectures | 3 | 22 |
| 26 | Develop Beta Version | 2 | 23;24;25 |
| 27 | Integration With Alice Chatbot | 3 | 26 |
| Phase 5 | 29 | Review Test Cases | 2 | Phase 4 |
| 30 | Defects Handling And Bug Fix | 6 | 29 |
| 31 | Quality Assurance | 2 | 30 |
| 32 | Final Release | 2 | 31 |
| 33 | Documentation | 4 | 31 |

## 5.3 Critical Path Analysis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Phase** | **Node** | **ES** | **EF** | **LS** | **LF** | **Slack** |
| **Phase 1** | 2 | 0 | 1 | 0 | 1 | 0 |
| 3 | 1 | 4 | 2 | 5 | 1 |
| 4 | 1 | 4 | 1 | 4 | 0 |
| 5 | 4 | 5 | 4 | 5 | 0 |
| **Phase 2** | 7 | 5 | 7 | 5 | 7 | 0 |
| 8 | 7 | 9 | 9 | 11 | 2 |
| 9 | 7 | 9 | 7 | 9 | 0 |
| 10 | 9 | 10 | 10 | 11 | 1 |
| 11 | 5 | 8 | 8 | 11 | 3 |
| 12 | 9 | 11 | 9 | 11 | 0 |
| **Phase 3** | 14 | 11 | 13 | 14 | 16 | 3 |
| 15 | 11 | 13 | 12 | 14 | 1 |
| 16 | 13 | 15 | 14 | 16 | 1 |
| 17 | 11 | 13 | 11 | 13 | 0 |
| 18 | 13 | 16 | 13 | 16 | 0 |
| **Phase 4** | 20 | 16 | 17 | 17 | 18 | 1 |
| 21 | 16 | 18 | 16 | 18 | 0 |
| 22 | 18 | 21 | 18 | 21 | 0 |
| 23 | 21 | 25 | 23 | 27 | 2 |
| 24 | 21 | 27 | 21 | 27 | 0 |
| 25 | 21 | 24 | 24 | 27 | 3 |
| 26 | 27 | 29 | 27 | 29 | 0 |
| 27 | 29 | 32 | 29 | 32 | 0 |
| **Phase 5** | 29 | 32 | 34 | 32 | 34 | 0 |
| 30 | 34 | 40 | 34 | 40 | 0 |
| 31 | 40 | 42 | 40 | 42 | 0 |
| 32 | 42 | 44 | 44 | 46 | 2 |
| 33 | 42 | 46 | 42 | 46 | 0 |

Nodes (Task) which have 0 slack values are the ones included in the critical path.

## 5.4 Activity Dependency Chart

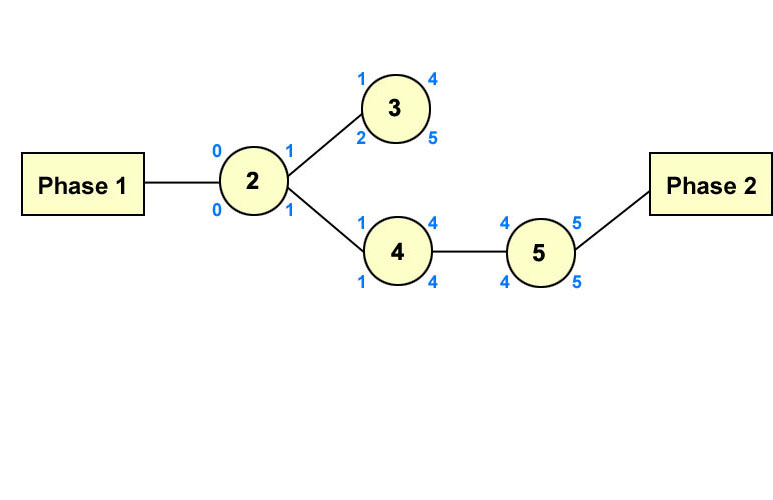


Figure - Phase 1 to Phase 2

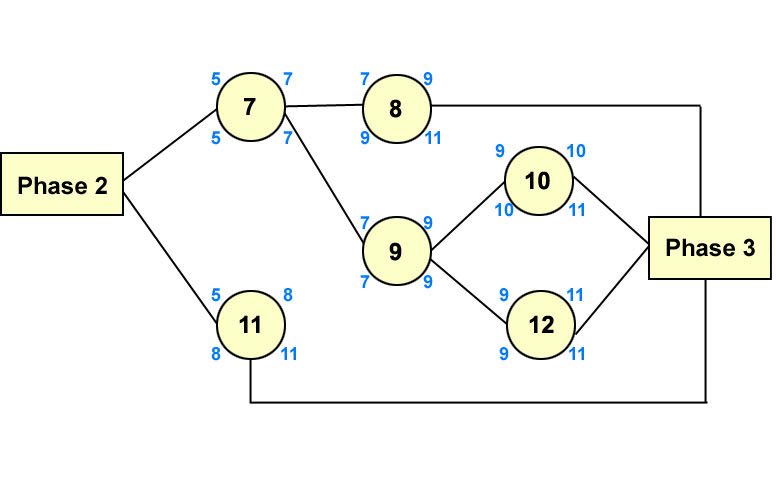


Figure - Phase 2 to Phase 3

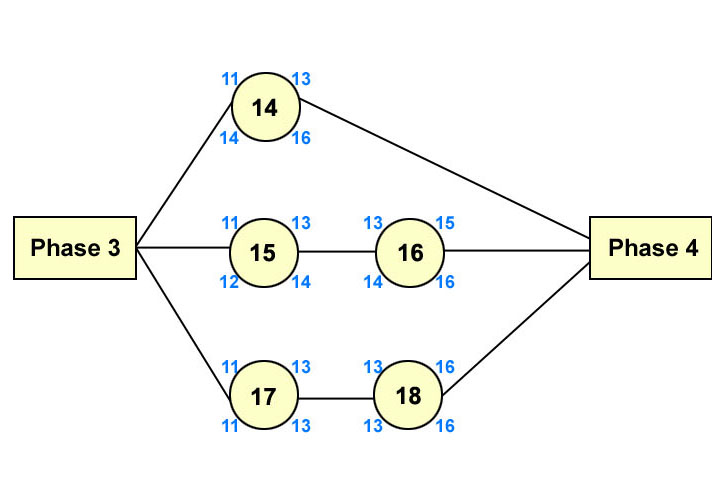


Figure - Phase 3 to Phase 4

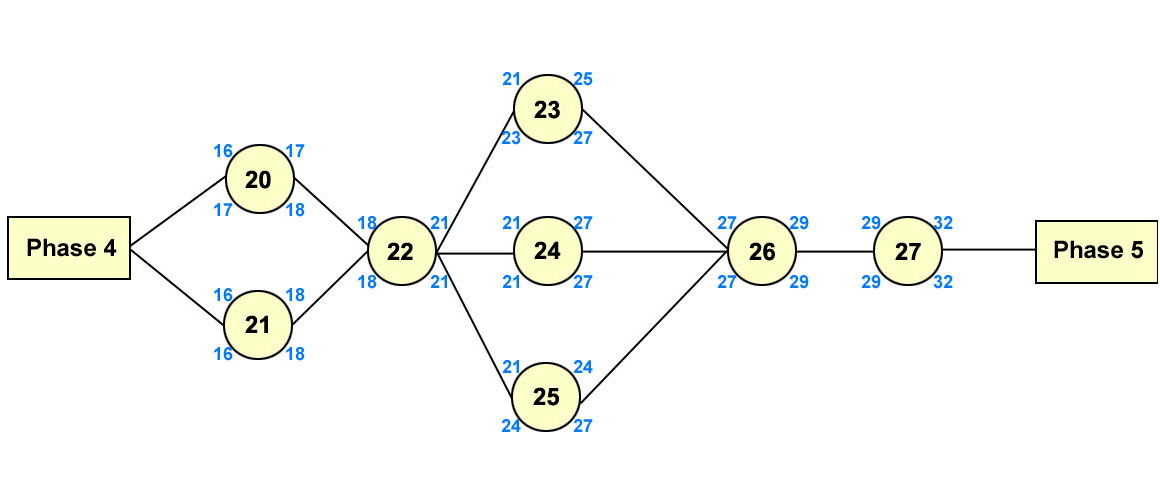


Figure - Phase 4 to Phase 5

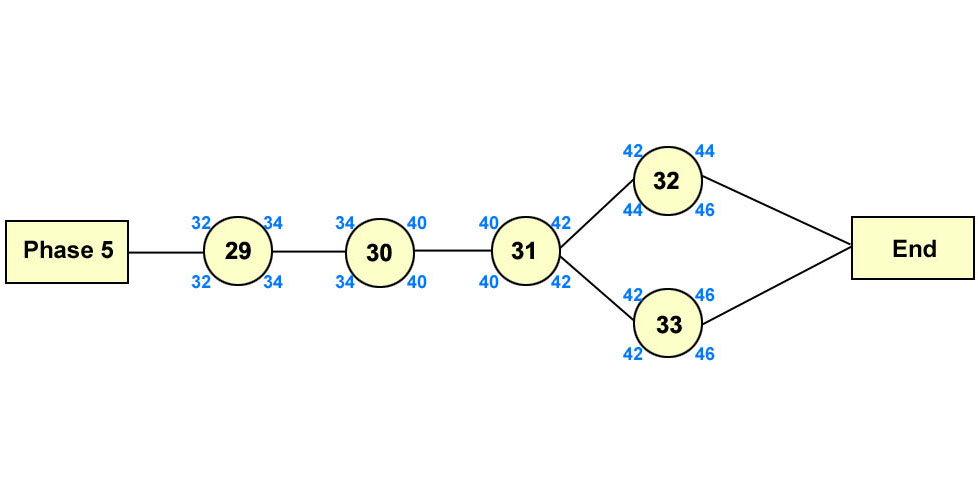


Figure - Phase 5 to Phase 6

Critical Path

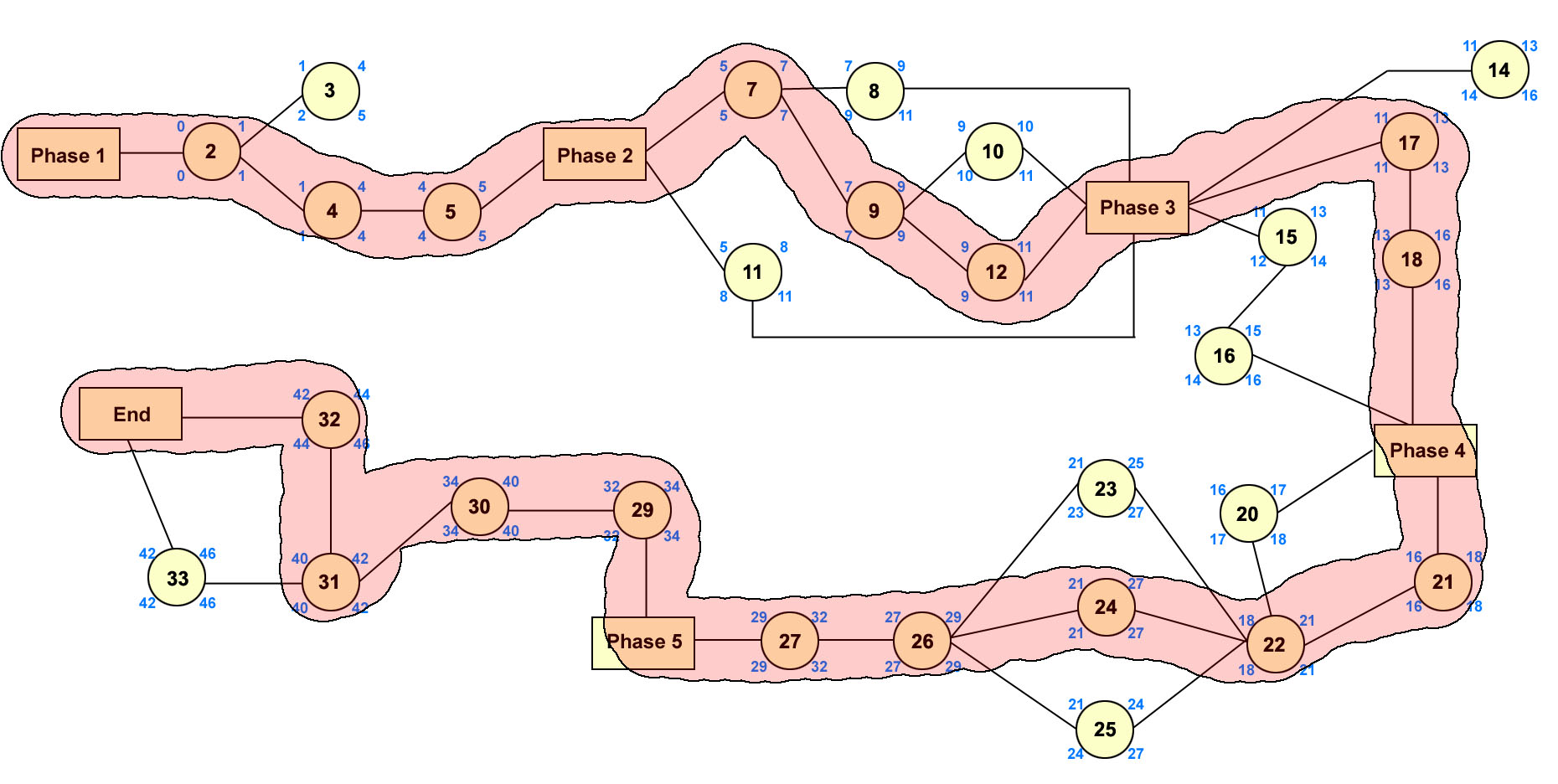


Figure - Critical Path Identified

## 5.5 Gantt Chart

Below is a screenshot taken while we are doing the project schedule. As indicated by the software automatically, the critical path is coloured as red.

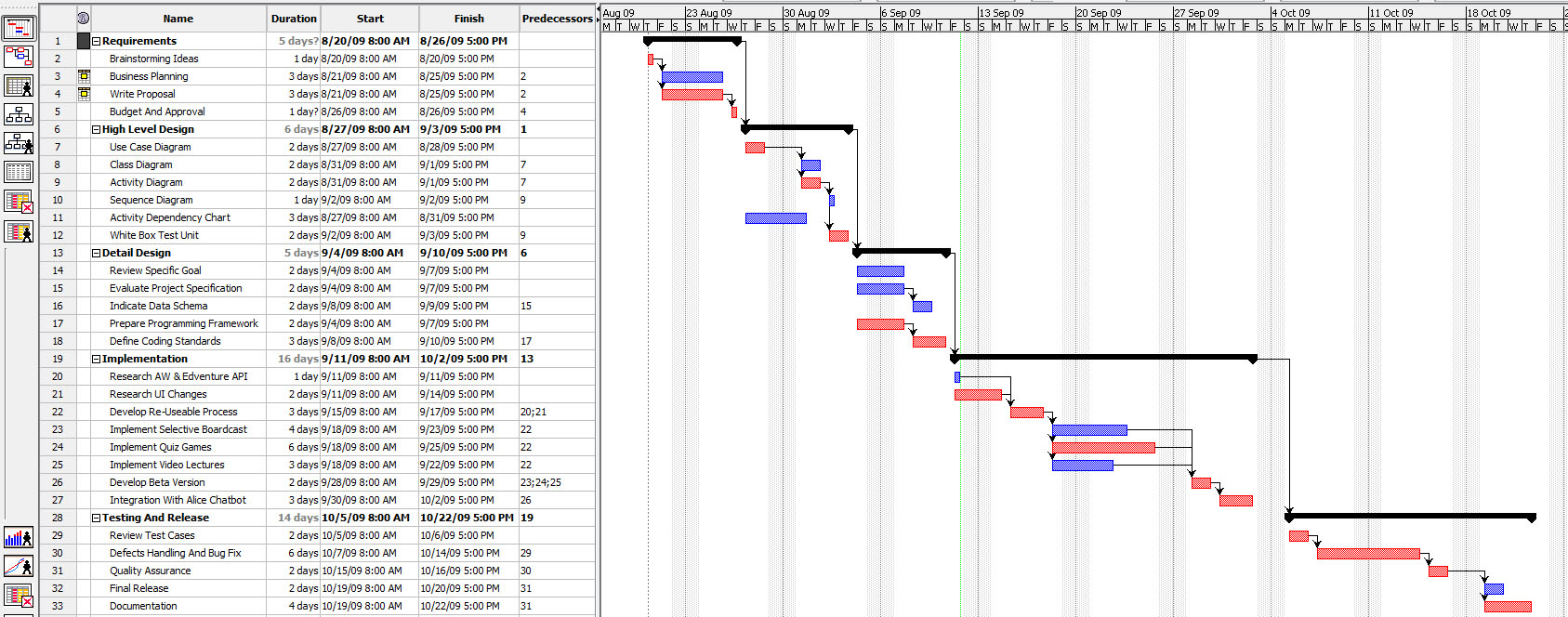


Figure - Gantt Chart

Total Weeks Schedule: 9  
Total Days Schedule: 45  
Total Hours Schedule: 407 hour

## 5.6 Work Package

10 work packages have been identified for this project.

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **User Manual** |
| Assigned to | Goh Li-xian |
|  |  |
| Start Date | September 7, 2009 |
|  |  |
| Purpose | Provide detailed instructions on what functions active world is capable of, and how to use the system functions |
|  |  |
| Inputs |        Design specification         System requirement specification |
|  |  |
| Activities |        Capture screen shots         Written detailed instruction on what functions active world is capable of, and how to use the system functions |
|  |  |
| Outputs | User Manual document |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Project Schedule** |
| Assigned to | Lui Quan Fu |
|  |  |
| Start Date | September 8, 2009 |
|  |  |
| Purpose |        Shows how project tasks and milestones are planned         Derive task dependencies         Varying scope, time, resource, and risk constraints to select and document the optimum project time and staff schedule         Monitoring the project life cycle’s progress |
|  |  |
| Inputs |        Project Statement |
|  |  |
| Activities |        Identify project tasks         Establish task duration         Assign staff resources         Derive task dependencies         Create the schedule         Modify the schedule if there is conflict between team members |
|  |  |
| Outputs | Gantt Chart, Work breakdown structure |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Process model** |
| Assigned to | Gillian Ng |
|  |  |
| Start Date | September 8, 2009 |
|  |  |
| Purpose | Encompasses the process, methods, and tools layers, and the generic phases |
|  |  |
| Inputs |        Project Statement         CMM (Capability Maturity Model)         Project Team CMM Level |
|  |  |
| Activities |        Define the lifecycle mode         Define CMM Key Processes         Define Project Processes |
|  |  |
| Outputs | Lifecycle Model, Documented Process Definition |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Lifecycle Model** |
| Assigned to | Sarah Lam |
|  |  |
| Start Date | September 8, 2009 |
|  |  |
| Purpose |        Define the lifecycle model used for this project         Provide guidelines for scheduling and planning |
|  |  |
| Inputs |        Project Statement         Past experiences and best practices |
|  |  |
| Activities |        Research existing lifecycle models and best practices         Define the most suitable lifecycle model for this project |
|  |  |
| Outputs | Lifecycle model |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Estimation** |
| Assigned to | Loh Bao Yuan Ivan |
|  |  |
| Start Date | September 8, 2009 |
|  |  |
| Purpose | Combining the results of post project reviews, metrics, and informed assessment to arrive at time and resource requirements for the system |
|  |  |
| Inputs |        Project constraints         Project assumptions         Project risk analysis |
|  |  |
| Activities |        List and examine the project constraints, assumptions, and risk that the system may encounter   |
|  |  |
| Outputs | Prediction of project size (LOC), effort and durations |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Risk Analysis** |
| Assigned to | Gillian Ng |
|  |  |
| Start Date | September 13, 2009 |
|  |  |
| Purpose | Analysis and quantification of all risks, and an evaluation of the range of possible project outcomes |
|  |  |
| Inputs |        Project statements, past experiences and statistics |
|  |  |
| Activities |        Risk identification         Mitigate a risk event         Deflecting the risk items to another entity         Consider risk event’s probability         Consider the effect’s potential severity |
|  |  |
| Outputs | Risk analysis |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Framework Design** |
| Assigned to | Lui Quan Fu |
|  |  |
| Start Date | September 16, 2009 |
|  |  |
| Purpose | To design the basic framework for the development |
|  |  |
| Inputs |        Data Structures (file allocations, table format, data)         Algorithms and common objects |
|  |  |
| Activities |        Develop commonly use modules         Evaluate each algorithms for the system |
|  |  |
| Outputs | Project Framework |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **User Interface Design** |
| Assigned to | Goh Li-xian |
|  |  |
| Start Date | September 16, 2009 |
|  |  |
| Purpose | Create an interactive UI for the system |
|  |  |
| Inputs |        Requirement Specification |
|  |  |
| Activities |        Create a set of interface design principles         Identifies interface objects and actions         Create a screen layout that forms the basis for an user interface prototype |
|  |  |
| Outputs | User interface design |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Black Box Test** |
| Assigned to | Sarah Lam |
|  |  |
| Start Date | October 2, 2009 |
|  |  |
| Purpose | Utilizes the black-box method and concentrates on testing overall functionality of the system |
|  |  |
| Inputs |        Requirement Specification         Design documents |
|  |  |
| Activities |        Prepare test cases         Examine the many different variables         Prove all the output is correct         Prove the system will not cause abnormal termination of the program or other undesirable results |
|  |  |
| Outputs | Test case report |

|  |  |
| --- | --- |
| Project | Virtual Classroom and Edventure |
| **Work Package** | **Quality Assurance** |
| Assigned to | Loh Bao Yuan Ivan |
|  |  |
| Start Date | October 5, 2009 |
|  |  |
| Purpose |        Demonstrate the system design and development process is complete         Demonstrate that the design risks have been minimized         Demonstrate that the system will meet specifications         Determining whether the system design is supportable (practical, maintainable, safe, etc.), for operational use         Providing test data with which to examine and evaluate tradeoffs against specification requirements, and schedule |
|  |  |
| Inputs |        System design         Risk analysis         Framework design         Interface design         System requirements specification         Test case report |
|  |  |
| Activities |        Create Test Plan         Determine whether the system design and development process is complete or not         Ensure whether the design risks have been minimized         Proof whether the system meet all the specifications         Test whether the system design is supportable         Evaluate tradeoffs against specification requirements, and schedule         Produce report |
|  |  |
| Outputs | Quality Assurance |

# 6 Project Estimates

## 6.1 Code Size Estimation using Function Points

We calculated unadjusted function point based on the complexity of functions provided by this system. Code size is then estimated by adjusted function point.

#### 6.1.1 Unadjusted Function Points

Pookas ETventure Virtual Classroom & Edventure supports the following proposed functions:

**Student:**

* Login to Edventure
* Selects lesson to attend and view the lesson slides retrieved from Edventure
* Views announcement retrieved from Edventure
* Query the Lecturer with questions
* Selects answer for Game Quiz with quiz questions from Edventure
* Broadcasting message to students taking similar modules
* Teleport to other coordinates

**Lecturer:**

* Posts announcement in Edventure
* Broadcast message to students taking his module

The proposed functions will provide the basis of measure of our unadjusted function points. They are characterized into five primary component elements: Inputs, Outputs, Inquiries, Logical Files and Interfaces. Each element has three types of complexity (Low, Medium and High) and will result in a different unadjusted function points.

**Inputs:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Complexity | | |
| Low | Medium | High |
| Lesson selected to attend | √ |  |  |
| Answer selected for Game Quiz | √ |  |  |
| Announcement Posted to Edventure | √ |  |  |

Outputs:

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Complexity | | |
| Low | Medium | High |
| Announcements on screen | √ |  |  |
| Quiz questions on screen | √ |  |  |
| Lecture slides on screen | √ |  |  |

**Internal Files:**

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Complexity | | |
| Low | Medium | High |
| Student’s game quiz results | √ |  |  |

External Interfaces:

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Complexity | | |
| Low | Medium | High |
| lesson slides from Edventure | √ |  |  |
| game quiz questions from Edventure | √ |  |  |
| announcements from Edventure | √ |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Characteristic | Low Complexity | Medium Complexity | High Complexity |
| # Inputs | 3 x 3 = 9 | 0 x 4 = 0 | 0 x 6 = 0 |
| # Outputs | 3 x 4 = 12 | 0 x 5 = 0 | 0 x 7 = 0 |
| # Inquiries | 0 x 3 = 0 | 0 x 4 = 0 | 0 x 6 = 0 |
| # Internal Files | 1 x 7 = 7 | 0 x 10 = 0 | 0 x 15 = 0 |
| # External Interfaces | 2 x 5 = 10 | 0 x 7 = 0 | 0 x 10 = 0 |
| Total Unadjusted FP | 43 | 0 | 0 |
|  |  |  |  |
| Total=L+M+H | 43 | | |

Influence Factors

|  |  |  |
| --- | --- | --- |
| Influence Factors | Score | Remarks |
| Data Communications | 4 | System has to access Edventure for its features |
| Distributed Functions | 0 | Each user can run Activeworld individually on his computer |
| Performance | 0 | No special performance required |
| Heavily Used | 2 | Anticipate an average usage of the system |
| Transaction Rate | 2 | Average transaction rate with most contributing from the game quiz |
| On-line Data Entry | 3 | Fairly average online data entry expected |
| End-user Efficiency | 0 | No special requirements for end-user |
| On-line Update | 2 | Average on-line update only |
| Complex Processing | 0 | No complex processing in system |
| Reusability | 2 | Deliverables can be used by other applications or projects |
| Installation Ease | 0 | Only require installation of Activeworld |
| Operational Ease | 2 | Used mainly by students and lecturers |
| Multiple Sites | 0 | No multiple sites specified |
| Facilitate Change | 3 | Changes can occur as the lesson and quiz can be changed in Edventure |
| Total Score | 20 | |
| Influence Multiplier  = Total score × 0.01 + 0.65 = 26 × 0.01 + 0.65 = 0.85 | | |
| Adjusted FP  = Unadjusted FP × Influence Multiplier = 45 × 0.85 = 38.25 | | |

LOC

According to Function Point Languages Tables by QSM Inc[[1]](#footnote-1)., each Function Point requires 29 lines of code if the application is implemented using C++

Therefore, our total **Lines of Code** = 38.25 FP × 29 LOC/FP = **1109 LOC**

## 6.2 Efforts, Duration and Team Size Estimation

In estimating our team’s efforts, duration and team size required for the project, we use the Top down and Bottom up estimation method, which provides validation to each other. We also used the COCOMO intermediate model to derive the estimation. This combination of different schemes allows us to have a reasonable and practical schedule. We ensure that the team’s estimation is as accurate as possible taking into account the total hours we work per week, reflecting a calculation that will enable us to pace our work processes on the total duration required.

#### 6.2.1 Top down estimation

In this method, we use the function points as a basis to calculate Effort, Duration, Team size, Compression rate. Our team intends to work 5 days a week and based on a team effort of 60 LOC produced per day, our calculation is as follow:

* Effort = Size / Production Rate = (1109 LOC) / (1200[[2]](#footnote-2) LOC/PM) = 0.924 PM
* Duration = 3 × (Effort)1/3 = 3 × (0.924) 1/3 = 2.92 PM
* Team size = 0.924 PM / 2.92 PM = 0.31 person

Since we have 5 in a team and we need to complete the project in 9 weeks (2.2months), we need to compress our schedule.

* Compression ratio = 2.2 / 2.92 = 0.753

As the compression ratio is greater than 0.75, it is actually possible to compress it into 9 weeks.

**Total person-hours** = 2.2 months x 8 hours x 22 days = 387 hours

#### 6.2.2 Bottom up estimation

We also used bottom up estimation to determine the duration and effort needed for our project. Based on the remainder duration of this course and the total hours we intend to spend a week on this project, we have come up with the following calculations:

There are 5 group members; each spends 8 hours/week; total hours per week is 40

Remainder duration of this course is 9 weeks, so the total person-hours is 9 x 40 = 360 hours

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1990’s Industry  Data | Work Package | Distribution | Top-Down  Estimates | Bottom-Up  Estimates |
| Preliminary Design 18% | Project Plan | 9% | 34.83 | 32.4 |
| Requirement Specification | 9% | 34.83 | 32.4 |
| Detailed Design  25% | User Interface | 7% | 27.09 | 25.2 |
| Technical Architecture | 11% | 42.57 | 39.6 |
| Data Modeling | 7% | 27.09 | 25.2 |
| Code & Unit Testing  26% | Code & Unit testing | 21% | 81.27 | 75.6 |
| Online Documentation | 5% | 19.35 | 18 |
| Integration & Test  31% | Integration & Quality Assurance | 31% | 119.97 | 111.6 |
|  | Extrapolated total effort |  | 387 | 360 |
|  | 2% for project management |  | 7.74 | 7.2 |
|  | 3% for contingency |  | 11.61 | 10.8 |
|  | Total effort |  | 406.35 | 378 |

#### 6.2.3 Intermediate COCOMO Model for Estimation

We have identified the project to be of organic mode since it is relatively small scale. The costs drivers are also identified as:

* Software Reliability – Nominal
* Product Complexity – Low
* Analyst Capability – Very High
* Applications Experience – High
* Programmer Capability – Very High

As such, we can compute the Effort Adjustment Factor (EAF)

EAF = Val (cost driver1) × … × Val (cost driver5)

= 0.85 x 0.71 x 0.91 x 0.7

= 0.384

Estimated effort required = 3.2 × (KLOC) 1.05 × EAF

= 3.2 × (1.109) 1.05 × 0.384

= 1.37 person-months

Estimated duration = 2.50 × (Effort) 0.38

= 2.50 × (1.37) 0.38

= 2.81 months

Estimated team size = Effort / duration

= 0.48 person

We intend to compress the project into 2.2 months (9 weeks).

Compression Ratio = 2.2 / 2.81

= 0.78

Since we have a team size of 5 people, and the estimated team site requires only around 0.5 FSP, we believe we are able to achieve completion of this project in 2.2 months. The compression ratio is 0.78 hence it is greater than 0.75 and based on our team’s capability; we will be able to complete the project in the compressed schedule.

Mini-conclusion

All the 3 estimation model shows that it is feasible for us to complete the project in 9 weeks.

## 6.3 Cost Estimates

Hardware:

|  |  |
| --- | --- |
| Laptop | |
| Description | **Quantity** |
| Dell Inspiron 14 Laptop  Intel Core 2 Duo T6600 Processor  2GB Ram  160GB Hard Drive  Windows Vista | **3** |
| HP Pavilion dv4t Laptop  Intel Core 2 Duo T5500 Processor  1GB Ram  80GB Hard Drive  Windows XP SP3 | **2** |
| Total Costs: $0.00 | |

|  |  |
| --- | --- |
| Developer Desktop (Available in Software Engineering Lab) | |
| Description | **Quantity** |
| Compaq dc8500 Core 2 Duo Desktop  Intel Core 2 Duo T8500 Processor  2GB Ram  250GB Hard Drive  Windows XP SP3 | **5** |
| Total Costs: $0.00 | |

Software

|  |  |
| --- | --- |
| Software Required | |
| Description | **Quantity** |
| Media Wiki | **1** |
| WAMP | **1** |
| Microsoft Visual Studio Express Edition | **5** |
| Microsoft Project 2007 | **1** |
| Active World | **5** |
| Microsoft Office 2007 | **5** |
| Subversion | **1** |
| Mailing List – Mail Man | **1** |
| Total Costs: $0.00 | |

Other Resources

|  |  |
| --- | --- |
| Other Resources Expenses | |
| Description | **Costs** |
| 5 Members total 406.35 working Hours @ $15/hr | **$6095.25** |
| Web Hosting of MediaWiki and Mailing List | **$30** |
| SVN Server | **$60** |
| Total: $6185.25 | |

The bulk of the costs are attributed to development costs incurred for the team members’ salary.

# 7 Process Modeling

## 7.1 Lifecycle Model

The lifecycle model used for this project will be the waterfall model with feedback. Development proceeds sequentially through a series of phases, as shown below. Feedback loops between each phase allow for necessary modifications to be made as verification is carried out. This model shows clearly the entry and exit conditions and milestones, which aids in project management.



Figure - Lifecycle Model

## 7.2 Capability Maturity Model

The Capability Maturity Model guides software process improvement by assessing it based on maturity level. Given the time constraints of this project, the level targeted is Level 2.

Some key processes have been dropped due to them not being applicable to this project. The relevant key process areas of this maturity level are as follows:

* Survival
* Requirements management
* Software project planning
* Software project tracking and oversight
* Software quality assurance
* Software configuration management
* Organization process focus
* Organization process definition
* Training program
* Integrated software management
* Software product engineering
* Peer reviews

The following key process areas are irrelevant due to the project being handled internally in one group:

* Software subcontract management
* Intergroup coordination

## 7.3 Project Processes

The following project processes will be carried out over the course of development.

|  |  |
| --- | --- |
| Process 01 | Requirements Management |
| Entry Criteria | Project description |
| Description | Defines the requirements of the software project. |
| Exit Criteria | Approved requirement specification |
| Begin  End | Understand the project description  Brainstorm for new features to be implemented  Document brainstorming sessions  Select new features to implement  Write requirements specification document based on prior training in CSC207 Software Engineering I  Review the requirements specification document  Requirements specification is agreed on and approved by management |

|  |  |
| --- | --- |
| Process 02 | Software Project Planning |
| Entry Criteria | Approved requirements specification document |
| Description | Defines the phases needed to complete the project, and estimates the effort, costs, and resources required. The roles of the team members are also assigned. |
| Exit Criteria | Approved Project Plan |
| Begin  End | Define objectives and scope  Assign roles to various team members  Analyze risks involved  Estimate size, cost, and time required  Decide on software and resources to be used  Produce Gantt chart based on reasonable estimates  Determine sequence of activities to be carried out (Critical Path Measure)  Document the above in a project plan  Review the project plan  Project plan is agreed on and approved |

|  |  |
| --- | --- |
| Process 03 | Architectural Design |
| Entry Criteria | Requirements specification document |
| Description | Based on the requirements specification, a detailed design document is written. A mock-up of the user interface is created |
| Exit Criteria | Detailed design document  Mock-up of user interface |
| Begin  End | Define objects in the system  Define properties and methods of the objects  Document these with a suitable modeling diagram  Create a mock-up of the ActiveWorld |

|  |  |
| --- | --- |
| Process 04 | Software Project Tracking and Oversight |
| Entry Criteria | Approved Project Plan |
| Description | The progress of the project is tracked in accordance with the software development plan and Gantt chart |
| Exit Criteria | Team Software Process documents  Approved project and documents delivered |
| Begin  End | Team Software Process used to log how the team spends their time  Project manager takes action and reassigns tasks based on the status of the project  Repeat until project is finished |

|  |  |
| --- | --- |
| Process 05 | Training Program |
| Entry Criteria | Approved Project Plan |
| Description | Development of the skills and knowledge of individuals so they can perform their roles effectively |
| Exit Criteria | Training session documentation |
| Begin  End | Plan for training  Identify technical and management skills required for the project  Identify organizational standards that need to be followed  Develop training program  Conduct technical and management training  Document training sessions |

|  |  |
| --- | --- |
| Process 06 | Software Configuration Management |
| Entry Criteria | Approved Project Plan |
| Description | Tracking and control of changes in the software product |
| Exit Criteria | Changelog  Approved project and documents delivered |
| Begin  End | Establish configuration management standard  Set up subversion system to keep track of changes in the software  Maintain configuration items repository  Document changes in new version.  Control change requests  Version release control  Repeat until project is finished |

|  |  |
| --- | --- |
| Process 07 | Implementation |
| Entry Criteria | High level design document |
| Description | Integration of software engineering and management activities into a defined software process. |
| Exit Criteria | Unit testing test cases  Documentation of problems and solutions  Deliverables passed to QA personnel |
| Begin  End | Perform unit testing  Document the test cases  Convert detailed design specifications into modules  Document the implemented functionality following the organization-wide standard  Meet to express concerns and difficulties encountered  Brainstorm for solutions  Document the problems and solutions  Pass the stable deliverables to the QA personnel for testing  Deliverables approved by QA  Maintain a history of approved versions |

|  |  |
| --- | --- |
| Process 08 | Peer Reviews |
| Entry Criteria | At any stage |
| Description | Team members to review each deliverable |
| Exit Criteria | Defects log  Peer review meeting log |
| Begin  End | When a deliverable is finished  Distribute deliverable to peers for testing  Peers identify defects and areas for improvement  **Peer review meeting held**  Document suggestions and points brought up at peer review meeting |

|  |  |
| --- | --- |
| Process 09 | Software Quality Assurance |
| Entry Criteria | Deliverables received by QA personnel |
| Description | Verification of the software products to ensure that they comply with requirements and function correctly |
| Exit Criteria | Deliverables approved by QA personnel  Modified and approved deliverables |
| Begin  End | Inputs and outputs of the deliverables are made clear to the QA personnel  Write test cases  Test deliverables  While bugs exist  **Document and report bugs to developers**  Fix bugs  Modified deliverables passed to QA personnel  Deliverables approved by QA personnel |

|  |  |
| --- | --- |
| Process 10 | Organizational Process Focus |
| Entry Criteria | Milestone reached |
| Description | Verification of produced documents and products to ensure that they comply with organizational standards |
| Exit Criteria | Deliverables and documents approved by management |
| Begin  End | Deliverables and documents passed to management  While non-compliant elements exist  Modify the documents and deliverables to conform to organizational standards  Pass back modified documents and deliverables to management  Documents and deliverables approved by management |

|  |  |
| --- | --- |
| Process 11 | Organizational Process Definition |
| Entry Criteria | Beginning of project |
| Description | Identify new software process assets required |
| Exit Criteria | New software process assets |
| Begin  End | Brainstorm for new processes and documents necessary during the project that might be useful for future projects  Propose such ideas to management  Document ideas  Implement feasible and necessary processes on an organizational level  Management makes sure that these processes are carried out consistently. |

# 7 Product Checklist

The plan is that the items listed below will be delivered on the stated deadlines.

|  |  |
| --- | --- |
| Project Deliverable | Estimated Deadline |
| Project Plan | 26 Aug 2009 |
| High Level Design | 03 Sep 2009 |
| Detail Design | 10 Sep 2009 |
| Implementation | 02 Oct 2009 |
| Testing and Release | 22 Oct 2009 |
| Design Document | 13 Sep 2009 |
| System Release (Demo) | 22 Oct 2009 |

**Project Deliverable Estimated Deadline**

# 8 Best Practices Checklist

This best practices checklist is specially catered to the development of Pooka’s EdveNTUre.

## 8.1 Project Plan Checklist

Project Organization

* Have the relevant roles and responsibilities been defined?
* Is training provided for the relevant skills required for the project?

Project Lifecycle

* Is the selection of a project’s lifecycle described and explained?
* Is any tailoring done to the lifecycle described or explained?

Deliverables

* Are the deliverables described?
* Is the relevant information on each deliverable included?

Risk Management

* Have possible risks been identified?
* Is there an assessment of the likelihood and consequences of these risks?
* Are there contingency plans or strategies to manage the risk?
* Is there close monitoring on the identified risks?

Software Estimation

* Is there an estimate on the size of the system?
* Is there an estimate on the effort needed for the project?
* Is there an estimate on the schedule?
* Is there an estimate on the cost?

Tracking and control

* Is there a method to keep track of the project’s status?

Technical Process

* Are the high level technical processes that will be used on the project described?

Peer review

* Does every team member review the deliverables?
* Is there a log of the errors?
* Is there a log of peer review meetings?

## 8.2 Requirement Checklist

Before you begin

* Did you brainstorm for new ideas?
* Did you narrow down the selection of these features as requirements?

Requirements

* Are all requirements uniquely identifiable?
* Are the requirements feasible?
* Can the requirements be implemented within known constraints?
* Are the requirements sufficient?
* Have functional and non-functional requirements been considered?

## 8.3 Architecture Design and Modeling

Design

* Have the design specifications been written down and finalized?
* Are the objects in the system defined?
* Are properties and methods of the object defined?
* Is there a modeling diagram to model the design?

Modeling

* Are all necessary model elements included in the design specifications?
* Are all unnecessary model elements excluded in the design specifications?
* Do all model elements have descriptive and appropriate names?

Model consistency

* Is there an interaction model, either collaboration or sequence, for every (non-trivial) use case?
* Are the classes in the interaction models consistent with the classes on the class model?
* Are the actors in the use case model consistent with the <<actor>> classes on the class model?
* Is every class and association in the class model accessed by at least one use case?

## 8.4 Code issues

Before you begin

* Do you understand the design you’re about to construct?
* Does the design provide an appropriate level of detail to begin?
* Is the design straightforward and feasible or should it be revisited before construction?
* Do you understand the language before you construct?

General

* Does the code have documented test cases/unit tests as defined by the project?
* Does the code compile with no warnings from the compiler?

Understandability

* Does the code read from top to bottom?
* Are implementation details hidden as much as possible?
* Is the code easy to understand?

Code changes

* Does the change enhance the program’s internal quality instead of degrading it?
* Is the modularity of the program improved by breaking routines into sub routines when possible?
* Is the programming style improved-variable names, routine names, formatting, comments?

Error handling

* Are error conditions handled appropriately as per the requirements and design?

## 8.5 System Testing

Testing

* Does a test case exist for each requirement?
* Do test cases exist to check for incorrect data (null values, negative numbers, incorrect data types, exceeding the maximum value, etc.)?
* Do test cases exist to check normal data values?
* Do the test cases comprise a minimal and complete set?
* Are techniques such as (white box testing, black box testing, stress testing etc.) utilized

1. Function Point Languages Table by QSM available at <http://www.qsm.com/?q=resources/function-point-languages-table/index.html> [↑](#footnote-ref-1)
2. Based on a team effort of 60 LOC per day, we will have 60 LOC per day x 20 days to achieve 1200 LOC/PM [↑](#footnote-ref-2)