Project Management Plan

By: Group 2

Master, Alan

Eng, Jonathan

Anderson, David Michael

Pandit, Bigyan

Akbar, Mohammed Sarim

Table of Contents

[1. Project Overview 1](#_Toc461741576)

[1.1 Project Summary 1](#_Toc461741577)

[1.2 Purpose, Scope, and Objectives 1](#_Toc461741578)

[1.3 Assumptions and Constraints 2](#_Toc461741579)

[1.4 Project Deliverables 2](#_Toc461741580)

[1.5 Schedule Summary 3](#_Toc461741581)

[2. Project Work Plans 4](#_Toc461741582)

[2.1 Work Activities 4](#_Toc461741583)

[2.2 Schedule Allocation 5](#_Toc461741584)

[2.3 Resource Allocation 10](#_Toc461741585)

[3. Project Context 12](#_Toc461741586)

[3.1 Process Model 12](#_Toc461741587)

[3.2 Methods, tools and technique 14](#_Toc461741588)

[3.3 Product Acceptance Plan 14](#_Toc461741589)

[3.4 Quality Assurance Plan 16](#_Toc461741590)

[3.5 Project Organization 16](#_Toc461741591)

[4. References 19](#_Toc461741592)

# 1. Project Overview

## 1.1 Project Summary

This project encompasses creating a new learning management system for grading quizzes. This system will allow users, based on their access level to log in, download/upload quizzes and answers, check grades, and view statistical information. Team members for this project are Sarim Akbar, David Anderson, Jonathan Eng, Alan Master, Bigyan Pandit. The intended customer and key stakeholder for the final product is Dr. Sergiy Vilkomir.

Ref: [1, section 5.1.1]

## 1.2 Purpose, Scope, and Objectives

Purpose: The overarching goal for this project is to create a working learning management system capable of distributing quizzes to students, collecting student responses to quizzes, generate statistical data on those quizzes, and track attendance.

Scope: The scope of this project is to develop the team member’s knowledge and understanding the foundations in software engineering while developing learning management system.

Objectives: There are many requirements/objectives that the final product should meet to successfully accomplish this projects goal of creating a working learning management system. requirements ranging from the administration of the system, logging in, and students uploading their work. The final deliverable should meet the following basic requirements.

1. All users will be able to access the project’s homepage and log on screen

2. All user will be able to login to the system using valid credentials

3. Instructor users will be able to upload quizzes into the system

4. Instructor users will be able to download student’s answers

5. Instructor users will be able to upload grades and comments into the system

6. Instructor users will have full access to all information in the system

7. Student users will be able to download quizzes

8. Student users will be able to upload their answers

9. Student users will be able to check their own grades

10. Teaching Assistant users will be able to upload attendance information into the system

11. Teaching Assistant users will have “read-only” access to all information on the system

12. Department head users will have “read-only” access to all statistical information

13. The system will be able to calculate the mean grade for each quiz

14. The system will be able to calculate the mean grade for all quizzes

15. The system will be able to calculate the percentage of attendance

16. The system shall display a student their attendance

Products to be delivered: The final product to be relieved the customer will be a complete folder with all source code, supporting files, and any necessary documentation required to operate the delivered learning management system.

Ref: [1, section 5.1.1.1]

## 1.3 Assumptions and Constraints

Our environmental constraints and assumptions are that we are assuming that our customer is running Microsoft Windows and is capable of maintaining a server. We also assume that the customer does not have any requirements on configuration and installation of the application.

For the deliverables we assume that each iteration will be an acceptable state despite not having full functionality.

This project also has a time constraint and must be completed by the end this semester.

Ref: [1, section 5.1.1.2]

## 1.4 Project Deliverables

In general, deliverables are work products that are delivered to the customer what they want. But sometimes deliverables are not just the final product; they are also the means by which projects are planned, managed and executed.

Process Deliverables

(internal to team members)

Product Deliverables

(external to client)

Deliverables

Process deliverables: These can be project management plans, requirements, models, design, test cases and so on. Process deliverables within the project can be an input to another team member to conduct their work.

Project deliverables: The final product of our project is a software program for grading quiz, and is similar in functions with ECU blackboard. As the software is developed based on incremental approach, each iteration will result a stable program capable of performing some specific task, even though it may not have full functionality. The system should calculate some statistical information for each quiz and all quizzes together; like mean grade, % of attendance and so on. As this system is for educational institution, it has different category of users and also different functionality. The final product should be able to do the following tasks for each category;

|  |  |
| --- | --- |
| **Category of users** | **Should be able to** |
| Students | 1. Login to system 2. Download quizzes and upload answers 3. View grades |
| Instructor | 1. Login to system 2. Upload quiz and download students answer 3. Upload grades and comment into the system 4. Full access to all information in the system |
| Administrator | Maintain the list of all the users and their passwords |
| Teaching Assistant | 1. Login the system 2. Upload information on the attendance into the system 3. Have “read only” access to all information |
| Head of Department | 1. Login to the system 2. Have “read only” access to statistical information only |

Ref: [1, section 5.1.1.3]

## 1.5 Schedule Summary

The timeframe of this project is between **30 August 2016** and roughly **1 December 2016**. The following is a rough breakdown of the tasks, effort in person hours, and duration of the project.

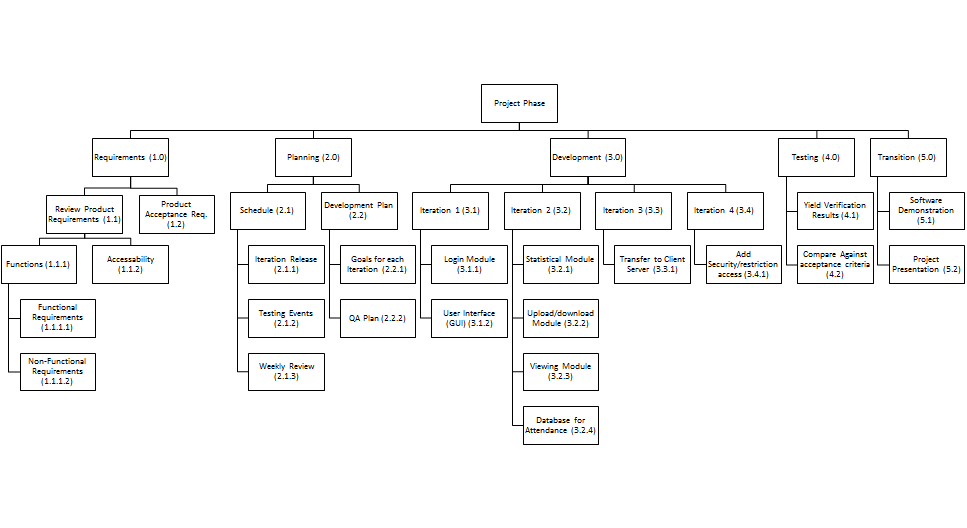
|  |  |  |
| --- | --- | --- |
| **Task** | **Effort** | **Duration** |
| Initial | 6 | 3d |
| Project Planning | 54 | 16d |
| Requirements | 48 | 14d |
| Iteration 1 | 78 | 12d |
| Iteration 2 | 120 | 14d |
| Iteration 3 | 84 | 14d |
| Iteration 4 | 48 | 12d |
| Testing | 72 | 7d |
| Maintenance | 12 | 2d |

Ref: [1, section 5.1.1.4]

# 2. Project Work Plans

## 2.1 Work Activities

The following graphic shows the work activities for each major phase of the project. The phases have been ordered in chronological order from left to right.



Ref: [1, section 5.5.2.1]

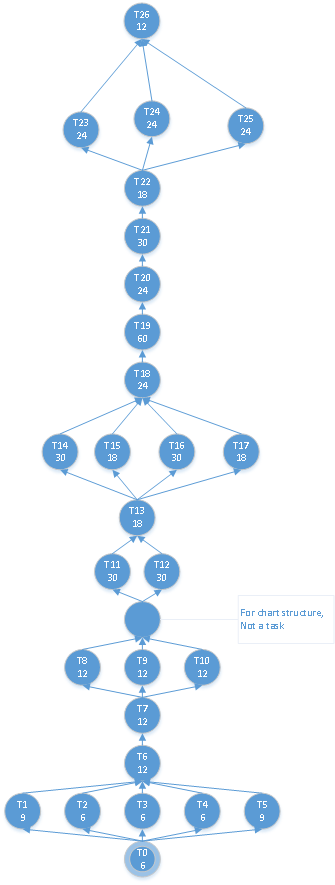
## 2.2 Schedule Allocation

The following is a breakdown of the schedule allocation for this project. This information is included in three forms. First, an effort breakdown table identifies all the tasks of the project and gives each task a code and an estimate of the amount of person hours for that task. These codes and effort hours are then organized into a critical path diagram that identifies concurrent tasks and dependencies and identifies critical paths(s) in the project. Finally, a Gantt Chart

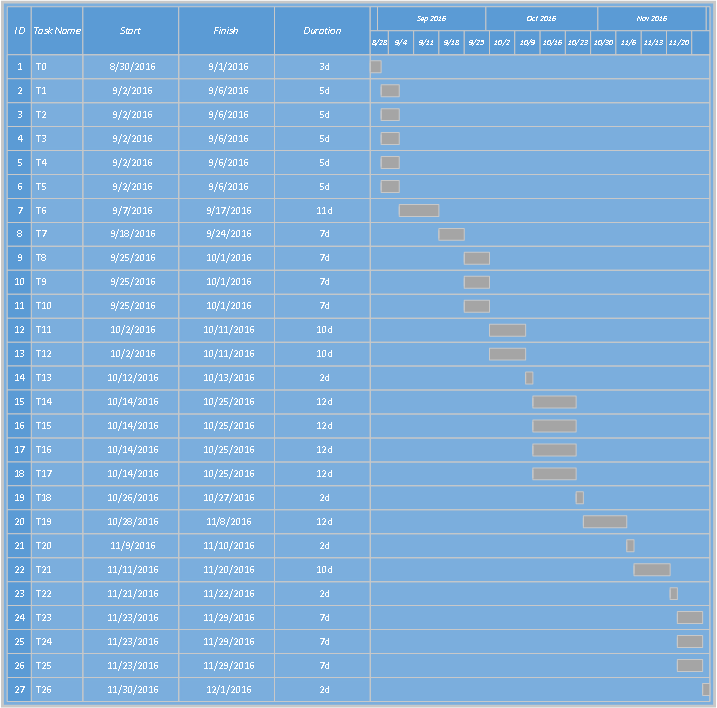
**Effort Breakdown**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Task Code** | **Description** | **Effort** |
| Initial | **T0** | Initial team discussion and introductions. | 6 |
| **Project Planning** | **T1** | project schedule specification | 9 |
| **T2** | Iteration release planning | 6 |
| **T3** | Project milestone specification | 6 |
| **T4** | Project review strategy planning | 6 |
| **T5** | Specify the deliverables of each iteration | 9 |
| **T6** | Creation of development plan | 12 |
| **Requirements** | **T7** | Requirements elicitation | 12 |
| **T8** | Functional requirements specification | 12 |
| **T9** | Nonfunctional requirements specification | 12 |
| **T10** | Creation of verification standards | 12 |
| **Development Iteration 1** | **T11** | Login module development | 30 |
| **T12** | User interface development | 30 |
| **T13** | Iteration 1 component testing | 18 |
| **Development Iteration 2** | **T14** | Statistical module development | 30 |
| **T15** | Upload/Download module development | 18 |
| **T16** | Viewing module development | 30 |
| **T17** | Attendance database development | 18 |
| **T18** | Iteration 2 component testing | 24 |
| **Development Iteration 3** | **T19** | Transfer to client-server application | 60 |
| **T20** | Iteration 3 system testing | 24 |
| **Development Iteration 4** | **T21** | security/access restriction | 30 |
| **T22** | Iteration 4 system testing | 18 |
| **Testing** | **T23** | Perform system verification | 24 |
| **T24** | Acceptance testing | 24 |
| **T25** | Final system testing | 24 |
| **Maintenance** | **T26** | Maintenance planning, future risk analysis | 12 |

**Critical Path Diagram.**



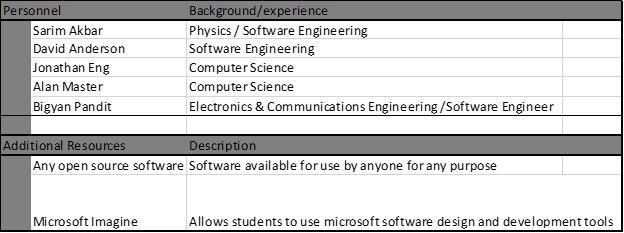
**Gantt Chart**

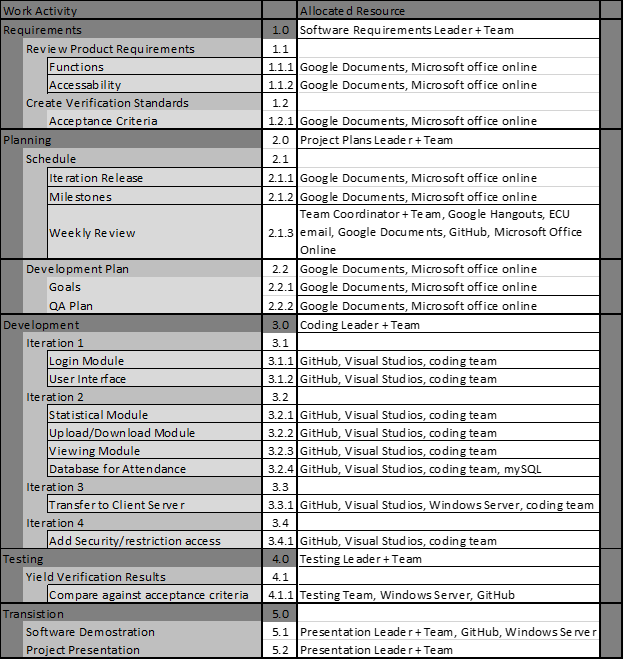


Ref: [1, section 5.5.2.2]

## 2.3 Resource Allocation

The resources allocated to this project are the five team members, any open source software, and any software design, development tools, and any other applications provided through East Carolina University’s Microsoft Imagine Premium catalog (formerly known as DreamSpark, formerly known as Microsoft Academic Alliance). The chart below showcases the resources available for use in this project.



The next chart provides a detailed itemization of our resources allocated to each of our major work activities. 

Ref: [1, section 5.5.2.3]

# 3. Project Context

## 3.1 Process Model

The process model we are using for our development is incremental. We are breaking our development into four sections which will give us several milestones to gauge our progress. Another key attribute of this process model that makes it appealing to us is that by sectioning our development into four iterations we can group similar subsystems together in iterations. Primarily what this does is group similar risks. Consolidating our risks allows us to properly plan each iteration more accurately. The four iterations we have identified for this development cycle is

Iteration 1: Creating single instance application with GUI pages and users

* The only functionality implemented is the user logins and conditional pages displayed based upon it.
* The main implementation is the GUI of the program  
  Consisting of:
  + Buttons,
  + Pages,
  + Text
  + etc.

Iteration 2: application with full functionality of all users and GUI interactions

* This is adding controller code to our view  
  Mainly consisting of:
  + Event handlers
  + Page navigation
  + File submission and data entering
* This will also include the functionality of our database and file system interaction

Iteration 3: separating into client server

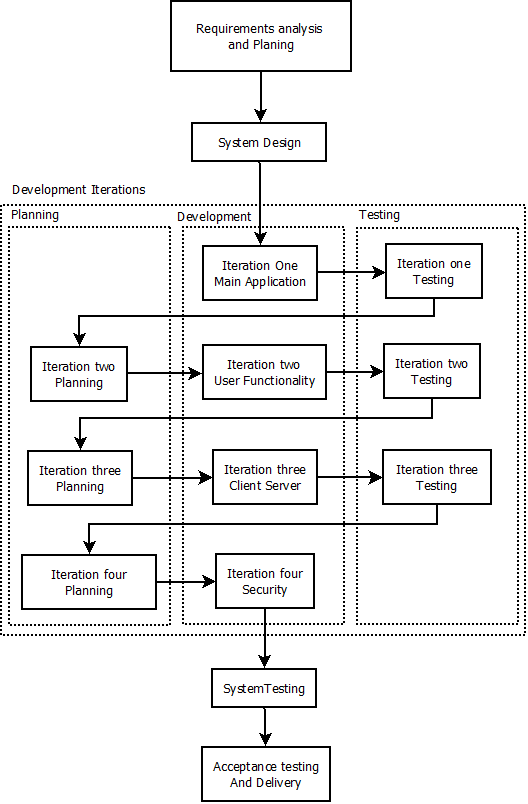
* This will mean creating a service to run on the server
  + We will also migrate the database over to the server service
* We will also implement packet transfers between the client and server

Iteration 4: adding security to full system

* Ensure security between client and server so that externals cannot manipulate database
* Ensuring login security and proper maintenance of passwords

The overarching process will have a planning phase, development phase, and test phase. The planning phase will consist of requirements and specification analysis. The Development phase will have 4 iterations with a revisit to the planning documents and testing in between iterations. The testing phase will consist of a system test and acceptance or delivery testing.

Another key thing identified in this process is that there are several places for modifications to be made throughout the development phase. This will be necessary as we receive updates to customer user stories and requirements. The test sections between iterations is also key to make sure that we can do incremental white box unit tests and gate the next iteration. This will allow us to continually test and ensure that we do not persist bugs in our development until delivery.



Ref: [1, section 5.4.1]

## 3.2 Methods, tools and technique

For tools our development environment will consist of several tools available on the windows OS platform. We will be developing our project in c# using Visual Studio. For a versioned controlled repository, we are using GitHub to centralize and collaborate our work. For our documentation we are using a combination of Microsoft office and Google documents. This documentation will consist of images, pdfs, and Text documents.

Our process model is following an iterative development technique. Our Project will be following a Model View Control method for code organization. Our standards are drawn from IEEE and ISO from our customer deliverables.

|  |  |
| --- | --- |
| **Environment Entity** | **Solution** |
| Operating System | Windows |
| IDE | Visual Studio |
| Repository | GitHub |
| Documentation | Microsoft Office  Google Documents |
| Communication | Outlook (ECU Email)  Google Hangouts |

Ref: [1, section 5.1.4]

## 3.3 Product Acceptance Plan

A product acceptance plan describes how the customer will evaluate the deliverable artifacts from a project to determine if they meet a predefined set of an acceptance criteria. It defines product acceptance tasks, e.g. identification of the test cases that need to be developed, demonstrations etc.

The first step is to explicitly identify which parts of the product acceptance process will be the responsibility of the customer and which will be the responsibility of the project team.

1) Product acceptance plan for the team

To build this software incremental development approach has been implemented by breaking the process into four sections called iterations. We will be testing things as we go down the process.

1. Iteration testing: The functionality of each iteration is mentioned on the Process model (5.4.1), testing is performed on each iteration to make it error free and finally check the acceptance of each iteration. The final product from each iteration should be able to perform some additional task in an error free environment.
2. Unit testing: test program components, such as methods or object classes
3. Component testing: test made for several interactions of objects
4. System testing: integrating components and testing
5. Acceptance/Release testing: Final test before the release. At this moment, the system should be free from all the bugs/errors.

During testing process, try to choose as many inputs that forces system to generate all error messages. After this we can work upon the errors and solve them to make error free.

2) Product acceptance plan for the Customer: This testing is done from the customer/user side to check whether the system is developed as per their requirement or not. As we are building software for grading the quizzes, the customer can check available features such as login in/out from the system, different category logins for students, instructor etc. and their ways of accessing of the system.

The customer also can check the complexity and functionality of the system and may ask ways to use it. They can also use samples values as an input to system and see the actual output. From student side, they can upload and download the file. From instructor side, he/she can upload quiz, grade it and so on. From this sample data, statistical measurements such as mean grades, % attendance can also be calculated to check functionality of the system to decide whether to accept the system or not.

Acceptance criteria for each iteration:

|  |  |
| --- | --- |
| Iterations | Acceptance criteria |
| Iteration 1 | Single instance application with GUI pages and users e.g. login |
| Iteration 2 | Application with full functionality of all users and GUI interactions e.g. file submission |
| Iteration 3 | Application service able to run on the server |
| Iteration 4 | Secure system |

Ref: [1, section 5.4.5]

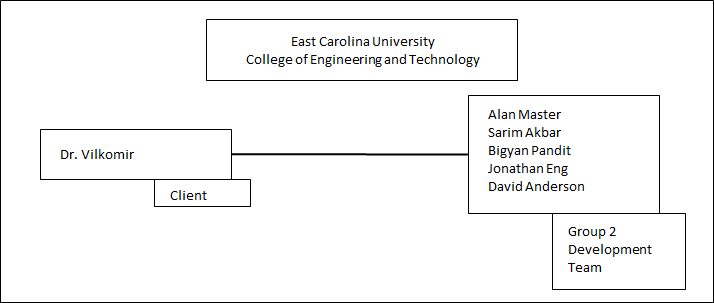
## 3.4 Quality Assurance Plan

As our QA plan, we will be conducting Iteration testing before each release to ensure the release meets our product requirements. The Testing will be conducted by Sarim Akbar, and Bigyan Pandit, and the results will be submitted to Alan Master and Jonathan Eng for verification before release.

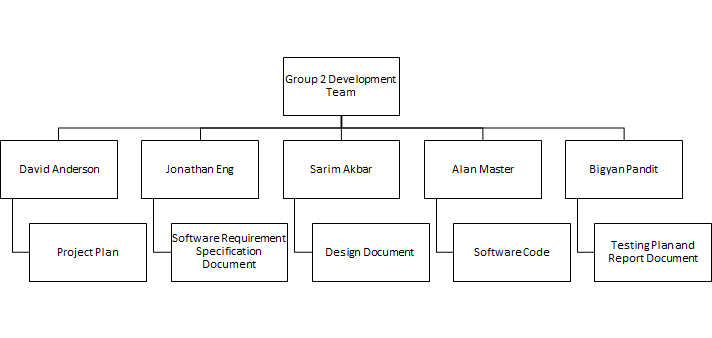
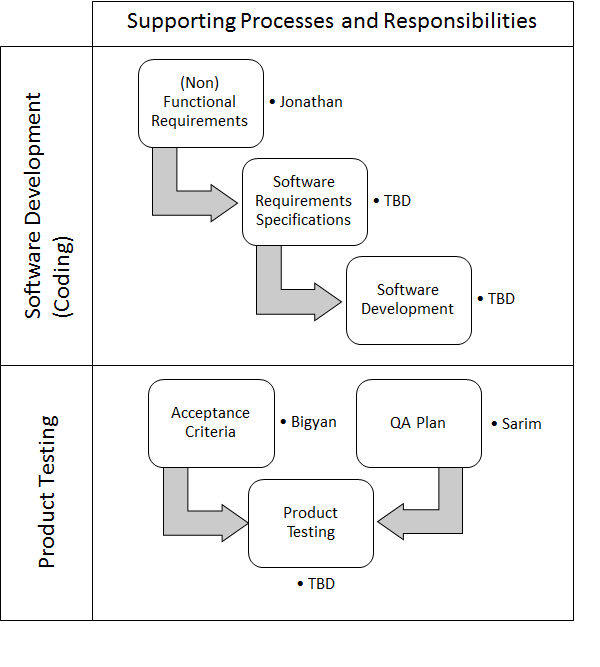
Ref: [1, section 5.6.5]

## 3.5 Project Organization

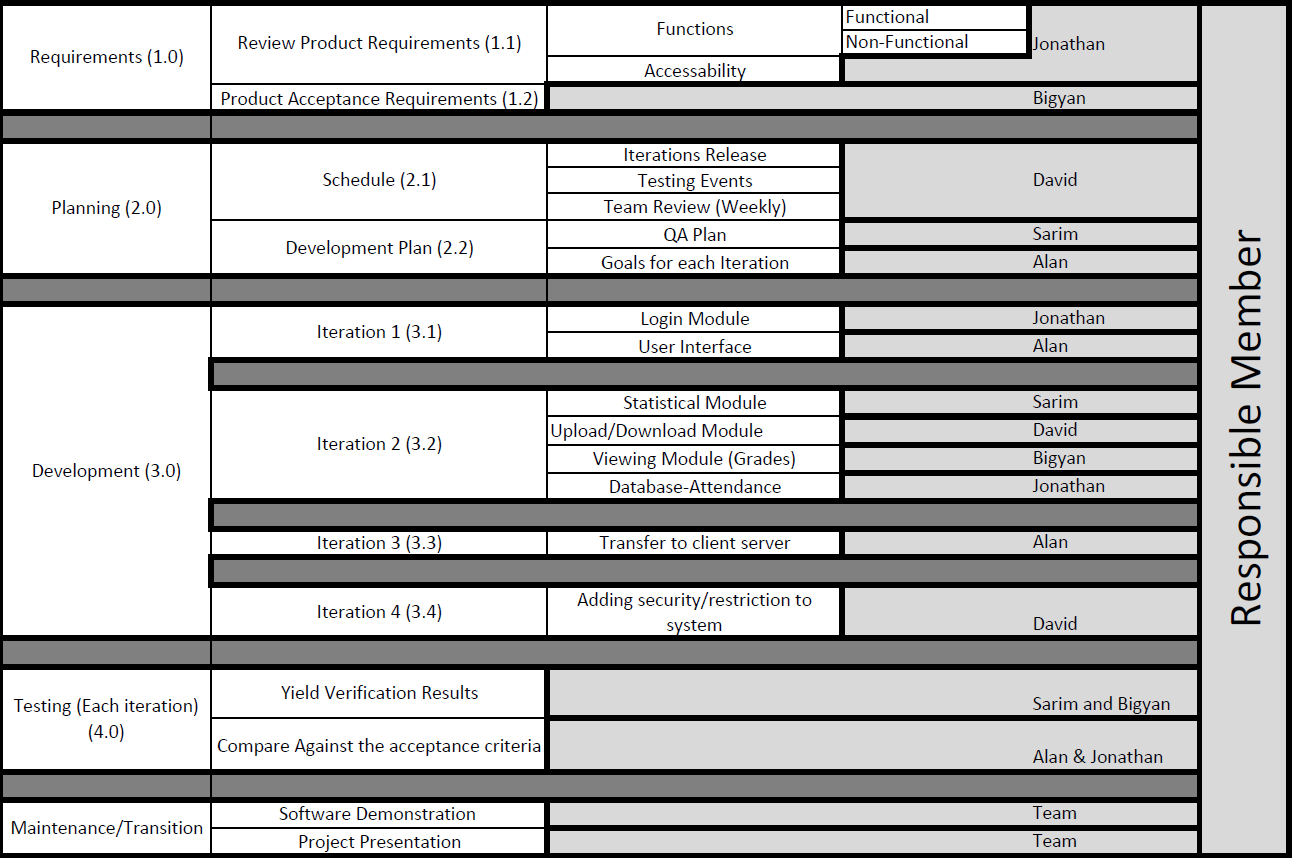
The first graphic depicts the external entities involved in this project. The client is Dr. Vilkomir who has requested development team, Group 2, to produce the quiz application.



The second graphic depicts the internal organization structure for Group 2. Based on the requested outputs for this project, a member from the team has been assigned as the manager for each output. The outputs have been placed in sequential order beginning with the project plan and ending with the testing plan and report document.

The following figure shows the breakdown for all the supporting processes/deliverables that are required to conduct certain dependent work activities.

The following graphic shows the breakdown of the work activities and the associated responsible member for each assignment.



Ref: [1, section 5.4.6]

# 4. References

1. Systems and software engineering — Life cycle processes — Project management

ISO/ IEC/ IEEE 16326, 2009

ISBN 978-0-7381-6116-7 STD95996