

Technical Document Evaluation Project Plan
Course: CMPS 5153: Software Engineering

Draft

Format: Scale 1-10 8

- Title page
- Revision History
- Table of Contents
- Font/ Spacing/ Margins consistent
- Use of Headers
- Section names and numbers proper
- Every section has text
- Captions and descriptions for figures and tables
- Necessary citations/references
- Page numbering

Content Scale 1-10 8

- Introduction
- Motivation/purpose/Scope
- Overview of document
- All relevant material/complete
- Organization
- Creative
- Correct spelling/grammar

Process Model - picture.
Can't see whole timeline.
Excellent!
on dependencies
Missing objective
grading sheet.

Name/Group Bozo's

Semester: Fall 2018

Final Grade: 18

Format Scale 1-10 9

- Title page
- Revision History
- Table of Contents
- Font/ Spacing/ Margins consistent
- Use of Headers
- Section names and numbers proper
- Every section has text
- Captions and descriptions for figures and tables
- Necessary citations/references
- Page numbering

Content Scale 1-10 9

- Introduction
- Motivation/purpose/Scope
- Overview of document
- All relevant material/complete
- Organization
- Creative
- Correct spelling/grammar

process Model picture
in table, but not diagram?

Turn in this sheet (and your original draft) in with your final draft.

Project Plan

for

Automated Software Engineering Student Subject Evaluations (ASESSE)

Version 1.1

Prepared by Tanaka Madyara, Deepthi Tummala, Samuel Weems



October 05, 2018

Table of Contents

Table of Contents	1
1. Introduction	2
1.1. Purpose	2
1.2. Scope of Project	2
1.2.1. Main Objective	2
1.2.2. Specific Goals	2
1.2.3. Major Functions	2
1.2.4. Performance Issues.....	3
1.2.5. Management and Technical Constraints	3
2. Project Organization	3
2.1. Team Structure	3
2.2. Management Reporting.....	3
2.3. Process Model	4
3. Risk Management	4
3.1. Likely Risks	4
3.2. Risk Monitoring	4
3.3. Risk Management Plan	4
4. Project Resources	6
4.1. People	6
4.2. Hardware and Software.....	6
5. Schedule	6
5.1. Project Breakdown	6
5.2. Timeline	7
6. Tracking and Control Mechanisms.....	8
6.1. Quality Assurance and Control	8
6.2. Change Management and Control	8
6.3. Communication	9
7. Glossary	9

1. Introduction

1.1. Purpose

The purpose of this document is to provide an outline and overall plan for the Automated Software Engineering Student Subjective Evaluation (ASESSE) project. This project plan describes the scope of the project including the primary objective, specific goals, major functions, performance issues and management and technical constraints. This document outlines the organization of the project team, describes the process model used for the development of the software and describes the methods for management reporting. This document provides an overview of project related risks, as well as how those risks will be monitored and managed. Finally this document also covers the proposed schedule and timeline for the project as well as tracking and control mechanisms that will be implemented. This document includes a glossary in the final section.

1.2. Scope of Project

1.2.1. Main Objective

The “Automated Software Engineering Student Subject Evaluation” (ASESSE) is a web-based application which automates the student peer review system for Software Engineering Students at Midwestern State University located in Wichita Falls, Texas. The main purpose is to allow students to review other students in their course group and to collate that information automatically for the professor.

1.2.2. Specific Goals

Each student in a group should be able to connect to the website, enter a password specific to their course, and then provide the information requested with regards to their specific group members through a graphical user interface. Though the student user submits their username and the review they made of their group members, this collected information will be stored separately and disconnected from one another to maintain anonymity.

The professor, or administrator, should be able to setup new classes with their own specific passwords, change the passwords if needed, delete or clear a class that has been setup, check to see the progress of all students in completing their reviews, request a report for an individual student that requests a group aggregate report of that student, and request an aggregate report which puts all individual student reviews into one printable pdf document.

1.2.3. Major Functions

User 1: Student

Student Logs into system, fills out form, submits form

The major object classes include a main program container (HTML), menu page to choose course (FE system), user login (FE system), then a blank form to enter information. When the information has been entered, the user can push a button to submit the form and a message asking to confirm this action appears (modal). Once the action is confirmed by clicking on a confirmation button, the information is submitted to the database.

User 2: Admin

The administrator logs into the system, creates a new user, removes a user, views a course's progress, views individual student progress, views course reports, views student group reports. These functions are described in more detail in 3.4

1.2.4. Performance Issues

The software must be able to perform database queries, store information to the database and create a collated version of database information in a reasonable amount of time. Student input should be saved into the database within a matter of seconds after submitting that information and display a success message to confirm to the student that their data was recorded. Most of the administrator selections should perform similarly with the exception of the printable pdf document which may take more than a few seconds but still should be available on the order of seconds rather than minutes.

1.2.5. Management and Technical Constraints

This project must be completed according to the timeline detailed in section 5.2 of this document. The software must be functional on the Midwestern State University Computer Science Department's student server (CS2 server) and must have an ease of portability built in to the project for transfer to the faculty server (CS server). The system must be programmed in an object oriented programming language so as to implement object oriented programming design elements such as classes and encapsulation. The project requires that the work be equally distributed among all group members.

2. Project Organization

2.1. Team Structure

The team consists of three graduate students: Tanaka Madyara, Deepthi Tummala, and Samuel Weems. The team chose Samuel Weems as team leader whose role is to facilitate the division of project requirements among members and to serve as a centralizing actor for the various materials and components that need to be formed into one. The team name is the Bozos

2.2. Management Reporting

The team will meet regularly and document those meetings by recording and posting the minutes of each meeting. Minutes will be kept and reported by the team leader using the communication methods listed in detail in section 6.3. In addition to reporting this minutes to the group, they will be sent along with progress updates to the client, Professor Stringfellow, as requested.

2.3. Process Model

The team will use the agile process model known as Scrum. This process model is most appropriate for the scale, timeline, and group size of this project. Traditionally, the team formation for the Scrum process model has three major roles: Product Owner, Scrum Master, and Development Team. However due to the Size of the team, the product owner (Samuel) and Scrum Master (Tanaka) will also be a part of the Development Team. The team will implement the traditional Scrum techniques such as the daily scrum meetings and development sprints.

NO Picture .

3. Risk Management

3.1. Likely Risks

Project risk includes illness of team members, change in requirements, lack of experience working in the required programming languages, web-based applications, and with databases. Since the project is web based, resource availability, namely the web servers are a risk as well.

3.2. Risk Monitoring

Given the small size of the group and scope of the project, all members will actively monitor the project and report potential or occurring risks. Communication methods detailed in section 6.3 provide for almost instantaneous notification of risks and the ability to address those risks. The team together will address risks as they arise in a timely manner. Some of the most likely risks have been evaluated and planned for in section 3.3

3.3. Risk Management Plan

The risk management table ~~below~~ (Table 1) shows risks of the project, how likely they are to be incurred, the impact that those risks may have on the project, and ways to mitigate those risks.

Risk Management Table

Table 1

Don't break across pages

4.

Risk	Likelihood	Impact	Strategy
Team Member Unavailable	Low	Low	Member who is missing or unable to complete work will be covered by other members of the group. When the member is able to return work will be redistributed to

Put on next page

			ensure equal workload of the course of the project.
Project Requirements Change	Low	Moderate	Regular documentations and updates should reduce this risk. If there are changes the team will meet to make adjustments to documentation and plans.
Team Coding Experience Lacking	High	High	There is a high likelihood that as team of students with little to no experience in web based applications there will be difficulties in producing reliable and functioning code. If this risk occurs the team will meet to give coding over to the most competent developers while the other requirements are redistributed to other members of the group.
Resource Availability	Moderate	High	There is a chance that the servers will not be functioning or available for testing and implementation. In this case a separate server has been acquired to host the project until the University servers are available.

4. Project Resources

4.1. People

(1) Team Bozos consists of 3 members, who are all listed in Section 2.1 of this project plan. All Team members have committed themselves to this project until its completion. Each and every member of the team will be involved in the development of the project. Developing, Testing, Documentation, Communication with the customer etc., of the project will be shared equally to all the members in the team.

(2) All the team members should feel free to consult the experts in database management system and CS servers (Dr. Terry Griffin), and general software engineering issues and process models (Dr.Catherine Stringfellow).

4.2. Hardware and Software

(1) Hardware for the development phases of the project consists of the personal desktop and laptop computers, as well as the computing facilities available to team members in the Midwestern State University computer science laboratories. Since the application must run over the internet, all the hardware is required to connect system to the internet.

(2) Software includes,

Microsoft Windows XP, Microsoft Visual Code

Microsoft Visual Studio, MySQL, Linux, Apache, Php

5. Schedule

The schedule for the project will outline the different stages of the project's execution. Due to the process model being Scrum, the client's involvement is necessary for the majority of the stages except for the development stage.

5.1. Project Breakdown

The project is broken down into the following stages:

- 1) Obtain project details from client
- 2) Outline requirements and specifications
- 3) Create project breakdown
- 4) Develop initial prototype
- 5) Test initial prototype
- 6) Enhance prototype
- 7) Test final prototype
- 8) Deliver final product to the client

To ensure that the final product is to the client's desired specification, any stage may be revisited.

5.2. Timeline

The timeline follows the project breakdown. The development process is broken into two-week sprints, with the database development given priority and the user interface and user experience being the last stage of development. Each team member is assigned with different tasks in the development process using the issue tracking tools Youtrack and Monday.com. A tentative timeline is as shown below in Figures 1 and 2.

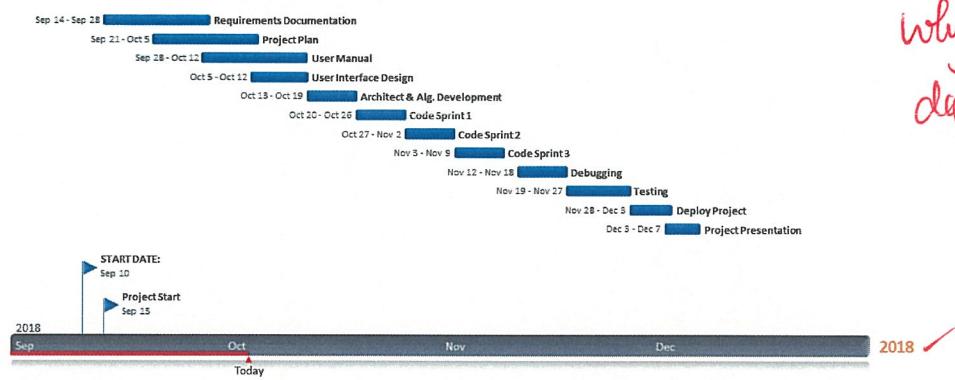


Figure 1

START DATE: 9/10/2018				
	ACTIVITY	START	END	Dependencies
1	Project Start	9/13/2018		
2	Requirements Documentation	9/14/2018	9/28/2018	
3	Project Plan	9/21/2018	10/5/2018	
4	User Manual	9/28/2018	10/12/2018	2
5	User Interface Design	10/5/2018	10/12/2018	4
6	Architect & Alg. Development	10/13/2018	10/19/2018	2
7	Code Sprint 1	10/20/2018	10/26/2018	6
8	Code Sprint 2	10/27/2018	11/2/2018	7
9	Code Sprint 3	11/3/2018	11/9/2018	8
10	Debugging	11/12/2018	11/18/2018	7, 8, 9
11	Testing	11/19/2018	11/27/2018	10 Testing on CS2 Server
12	Deploy Project	11/28/2018	12/3/2018	10, 11, 4; Deployed on the CS server
13	Project Presentation	12/3/2018	12/7/2018	12

Figure 2 

6. Tracking and Control Mechanisms

6.1. Quality Assurance and Control

CS Servers will be implemented first and then connecting the databases and servers using the required process model will be done next. Testing will be performed. Required action will be taken when the project does not behave as expected.

Documentation of all code will be done to insure readability. Programming will be done in object oriented programming using classes and encapsulation. At the point when the quality tests are done, if the quality is satisfactory, the specific procedure will be standardized, if not the procedure will be improved. Informal tests will check for proper integration with the database, proper representation of the user interface, implementation of drag and drop capabilities, and appropriate update of the database. A test plan will be produced and used at the end of the project.

6.2. Change Management and Control

If a change in the requirements is needed, a procedure for control management will guarantee that all the changes are made with reliability and are in a controlled manner. The following procedure will be taken:

- (1) Identifying the problem
- (2) Investigating the problem
- (3) Reviewing the investigation results
- (4) A change will be proposed
- (5) Analyzing the effects that are occurred through the changes made
- (6) A decision will be made whether to proceed with the change or not
- (7) If change is approved, then modifications will be done to the design and implementation and the required documents will be modified.

6.3. Communication

All the team members are communicating through Slack and the required documents for the project development or the changed requirements will be shared through a Github repository. Also group members will attend the team meetings whenever required or needed as well as be available through email and phone communications.

7. Glossary

ASESSE - Automated Software Engineering Student Subject Evaluations.

HTML - Hyper Text Markup Language.

Constraints - A restriction that defines the project limitations or boundaries.

Encapsulation - Used to hide the structured data object inside a class, preventing the unauthorized parties direct access to them

Timeline - Used to help team members to know what milestones need to be achieved and under what time schedule.

Portability - The Server or software which can works on any other environments.

Youtrack - Commercial browser-based bug tracker, issue tracking system and project management software.

Group Aggregate Report - The Aggregate report of all students who are in the same group or team of the project.

Course Aggregate Report - The Aggregate report of all students who has taken the same course.

Number of Members	Duties and Deliverables	5	Weight	Samuel Weems	Tanaka Madyara	Deepthi Tummala	Total Points
Phase							
Requirements	R = .15 * 300 = 45						
	Use Cases	0.20		9.00			4.50
	H/W & SW requirements (and tools)	0.10			2.25		2.25
	Draft Document	0.15			2.25		
	Other Diagrams	0.10			4.50		
	Final Req. Document	0.20		3.00	3.00		3.00
	Reviews	0.10		1.50	1.50		1.50
	Interim Presentation	0.15					
SubTotal		1.00		15.75	11.25	11.25	45
Design	C = .25 * 300 = 75						
	GUI Design	0.35					
	Navigation Diagram	0.20					
	Sequence Diagrams	0.10					
	Class Diagram	0.15					
	Other Diagram	0.10					
	Reviews	0.10					
SubTotal		1.00		0.00	0.00	0.00	75
Implementation	C = .25 * 300 = 75						
	GUI Final	0.15					
	DB	0.10					
	Increment/Iteration 1	0.15					
	Increment/Iteration 2	0.15					
	Increment/Iteration 3	0.15					
	Reviews	0.10					
	Final UJD	0.20					
SubTotal		1.00		0.00	0.00	0.00	75
Testing	T = .15 * 300 = 45						
	Schedule	0.20					
	Test Cases / Criteria	0.40					
	Reviews	0.10					
	Test Plan	0.30					
SubTotal		1.00		0.00	0.00	0.00	45
Manual	M = .05 * 300 = 15						
	Outline	0.30					
	Screen Shots	0.20					
	Preliminary User Manual	0.20					
	Final User Manual	0.10					
	Reviews	0.20					
SubTotal		1.00		0.00	0.00	0.00	15
Plan	P = .05 * 300 = 15						
	Plan Resources	0.10		0.50	0.50	0.50	
	Schedule Draft (MS Project)	0.20		1.00	1.00	1.00	
	Task Network Diagram	0.10					
	Risks	0.10		0.50	0.50	0.50	
	Project Plan Document	0.40		2.00	2.00	2.00	
	Reviews	0.10		0.50	0.50	0.50	
SubTotal		1.00		4.50	4.50	4.50	15
Presentation	C = .10 * 300 = 30						
	Final Report	0.30					
	Deliverables	0.30					
	Presentation	0.40					
SubTotal		1.00		0.00	0.00	0.00	30
Total Points.....		20.25		15.75	15.75	15.75	300.00

V