CS 441 Semester Project

Policies and Features

Introduction

Term projects for Software Engineering classes are typically designed to allow students to gain teamwork experience and to be involved in the software development process from inception to deployment. In this class, the instructor will not interfere with team formation as long as the following conditions are met:

- All teams will start with 5 or 6 students. (groups of 5 must get Instructor approval)
- All students are members of exactly one team.

This document covers the essentials for your semester project. Information contained herein covers your roles, your required deliverables and grading criteria. Save this document – read it often.

Team Roles

Each team will be divided into 3 groups/types of members. Each student will be expected to perform all three roles by the end of the semester.

- Programmers
  - There will always be two programmers who will "Pair program". Familiarize yourself with what this means.
  - Each programmer should have time at the keyboard.
  - Document your code
  - Include your names at the beginning of each function/method you write
  - Include a heading section for each .cs file created
  - After the first cycle, all programmers will use existing code. NO REWRITING
  - Fix all bugs as identified by Testers
  - Programmers may talk to Analysts
  - Programmers may communicate to Testers via bug documentation
- Testers (QA)
  - Testers are responsible for writing test cases based on the SRS
    - Positive
    - Negative
    - Edge
  - Testers are responsible for testing code submitted to them and documenting results
    - Maintain documentation of results of test cases
    - Maintain defect or bug documentation
  - Testers may talk to Analysts
  - Testers may communicate to Programmers via bug documentation
- Analysts
  - Analysts have the responsibility of scribes. Familiarize yourself with what this means.
  - Analysts are responsible for writing and updating the SRS.
  - Analysts are responsible for writing the User Guide/Manual
  - Analysts have the responsibility of clarifying the SRS for teammates
  - Analysts ensure all team documents are uploaded to sharePoint in a timely manner
  - Analysts are the lead presenters of the project

Deliverables

The project development will be divided into three cycles. The following table lists the deliverables required for **each** cycle. Please note the media changes for the Final cycle. The bound hard copy shall be submitted on or before the Final class. Soft final copies shall be uploaded to Moodle per Moodle instructions.

Description of Deliverable	File Type	Media	Final May Media
A working application	.exe	CD/DVD	CD/DVD
Readme file	.txt	CD/DVD	CD/DVD
Source code	.cs	CD/DVD	CD/DVD
SRS	.doc or .pdf	Moodle	Hard copy
XMLs	.xml	Moodle	Hard copy
Test Cases	.doc or .pdf	Moodle	Hard copy
Testing Results	.xls or .csv	Moodle	CD/DVD
Defects or Bug document	.xls, .csv, .doc or .pdf	Moodle	CD/DVD
User Manual	.doc or .pdf	Moodle	Hard copy

Application:

- a stand-alone exe created in Visual Studio with C#
- The program must conform exactly to the documented specifications

Readme file:

- Use this file to explain your program deployment
- Document all bug fixes since previous deployment
- Document all new features since previous deployment
- Include the app version and team information
- Readme shall be written by programmers and verified by Analysts

Source code:

- Source should be well documented
- Include a heading section that includes a revision history for each unit created
- Submit all files

Software Requirements Specification:

- The SRS is a living document
- Be sure to keep the revision history section up-to-date

XMLs:

- Submit samples of all XMLs your program uses (pdf format)

Test Cases:

- Submit the test case document
- This is a living document, be sure to keep track of versions (both for test cases and software)

Testing Results:

- Submit the results of all test cases
- Beginning with cycle 2, maintain a spreadsheet for all test runs
- Keep track of tester names, dates, and software versions

Bug document:

- A working document that serves as the communication between the testers and the programmers – so be detailed
- Keep track of names, dates, software versions and bug status

User Manual:

- Instructions for using the app, including screen shots
- Note: An online manual (help) will be accepted with prior instructor permission

Grading (40% of the final grade calculation)

The project will be evaluated based on the following categories:

- Implementation of a well-defined software development **process**
    - A good process will result in continuous deliverable products leading to the on time completion of project requirements.
    - For full credit, evidence of the process must be available to the Instructor
    - Students who fail to demonstrate commitment and meaningful contribution to the success of the project will receive a failing grade for the course. The project will move on without waiting on such an individual's contribution.
- Clearly articulated and carefully reviewed, validated **requirement documents**
    - The software requirement document for each team shall reflect the software being developed by the team. If another skilled team were to take the requirements document to develop the software, it should be able to come up with a very similar and functionally equivalent product.
    - Design documentation should be added to the SRS
    - The software design shall reflect the architectural and component design of the software being developed by the team.
- Records of verifications and validations (**V&V**)
    - The V&V documents shall include records of testing throughout the project. The reviews and critiques of the requirement and design documents are part of such records. The test cases and bugs revealed by the test cases are also part of such records. And, yes, the successful runs of the software are also part of such records.
- The **working product that satisfies the requirements** and conforms to the design