Software Enterprise II - Course Project
Software Maintenance, Testing & Quality

Instructions: Through this project you will learn about Software Testing (various tools, creating software test plan, performing software testing) and Software Maintenance activities of the software development lifecycle. You will work in groups of 3. Your project submissions should be made via Blackboard by the specified deadline. You will work through this project in multiple iterations. There are deliverables at the end of each iteration. Each team will be provided one existing software product (an open-source product, a research software tool, a software game, etc.) that you will work on for Software Testing and Maintenance.

---

1. Setup and Execute the project

- Download the source code provided on Blackboard, create an Eclipse project and build it
- Understand how the software works
- Use the software as an end-user

Deliverables: none

Due Date: end of Week 2 (by 11:59pm)

---

2. Software Testing tools survey

A consolidated list of open source software testing tools is available at http://www.opensourcetesting.org/. Each team is required to select TWO testing tools from this list provided, each from a different category of tools (such as Unit Testing, Functional Testing, Defect tracking, Performance, etc.). A tool not listed here may also be selected upon approval from the instructor. Each team will provide a written report with description of the **2** chosen tools via Blackboard. For each tool, you must download the tool and learn to use it. Your papers must answer the questions below in their report.

1. Who (company or individual) developed the tool? What is the cost to the tool user? How do you acquire it?
2. What testing purpose does the tool serve? (i.e, what problem does it attempt to solve? How does it improve productivity?)
3. What programming language(s) does the tool support, if any?
4. In what phase of software testing is the tool useful?
5. What do you need to do in order to use the tool?

a. How do you install it?
b. How do you configure it?
c. How do you use it?

6. What are the strengths and limitations of the tool?

Deliverables: Written Reports (2) on the selected tools (submit via Blackboard)

Due Date: end of Week 3 (by 11:59pm)

---

3. Software Design: Understand the code

- Analyze the code
- Create use case diagram and class diagram for the existing source code. Capture all classes in the code base. Create an Activity diagram and State chart where needed to model dynamic behavior of the system. Use a UML tool of your choice to create these diagrams.

Deliverables: UML Diagrams as picture files (via Blackboard)

Due Date: end of Week 4 (by11:59pm)

---

4. Create a Software Test Plan

Create a Software test plan that includes a detailed description of the testing activities (e.g., individual test cases), a description of the coverage of the testing effort, etc. A template will be provided in class.

Deliverables: Submit Test Plan document (via Blackboard)

Due Date: end of Week 5 (by 11:59pm)

---

5. Perform Testing as per your plan

Perform testing and create a list of bugs and possible enhancements. From this list, identify a few bugs and enhancements that you can fix.

Deliverables: List of Bugs and Enhancements (via Blackboard)

Due Date: end of Week 5 (by 11:59pm)

---

6. *Implementation of Bug fixes and enhancements*

Implement bug fixes and enhancements that you have identified in the previous step and test the software thoroughly.

Deliverables:

- Compressed folder that contains the complete source code
- Executable jar file
- Summary Report that addresses the following:
    - Description of the goals of your testing efforts
    - What are the results of your testing? What coverage did you achieve?
    - What did you discover?
    - Summary of bug fixes and enhancements (status of bug fixes, if an error is not fixed then explain why)
    - How would you improve your process in the future?
- Individual Contribution report
- CATME Peer-review survey

Due Date: end of Week 6 (by 11:59pm)

Grading Rubric:

Deliverable - UML Diagrams: **40 points**
Deliverable - Testing tool reports (2): **80 points**
Deliverable - Software Test plan: **70 points**
Deliverable - List of Bugs/Enhancements: **40 points**
Deliverable - Implementation of fixes: **40 points**
Deliverable - Summary Report: **40 points**
Individual Contribution report and Peer review: **40 points**
Total: 350 points
