15-313: Foundations of Software Engineering

Homework 5: Testing

In this homework, you will learn how to perform different types of testing on a software system beyond testing functional correctness with unit testing. The key learning goals of this homework are that you will:

- Get an overview of different testing approaches,
- Gain experience using selected testing tools and techniques,
- Select suitable testing techniques for quality attributes, and
- Learn how to write comprehensive test cases.

In this assignment you will use testing to evaluate selected quality attributes of Etherpad, a web-based collaborative real-time editor, with testing techniques of your choice. You will write several test cases (as team) and reflect on your experience (individually).

Preliminaries

Test Cases. You will write test cases and document their outcomes for Etherpad. In general, there is no single agreed-upon way of how a test case should be documented. For the purposes of this homework, a test case is *a set of reproducible steps that help a tester determine whether a desired software quality is satisfied.* In addition, your test case descriptions will include whether the test passed or failed when you executed it.

Each test case should have the following structure:

- Test case name
- Testing tool used (if any)
- Quality attributes to be evaluated
- Justification
  (Why do you think this test verifies the quality attributes?)
- Test specification
  (What should happen at the end of the test that you say that it passed? Be specific.)
- Steps to take
  (Include at least one screenshot per each step, clearly labeling the elements that we need to focus on and displaying the results when applicable.)
- Test result
  (Did the test pass or fail? Why? Give a brief explanation referring back to the test specification.)
- Any additional remarks you may have

A sample test case description is provided in appendix at the end of this document.

In industrial practice, the information you are required to include when documenting a test case differs based on what type of software you are testing and what testing practices the company is

using, and may not require the same or as much information as we are asking for in this homework.

System to Test. Etherpad (http://etherpad.org) is an open-source web-based collaborative real-time editor. It allows users to simultaneously edit a text document and see all of the participants' edits in real-time. The project source code is available from GitHub: https://github.com/ether/etherpad-lite. For this homework, don't test a public installation of Etherpad, but instead set up a local copy.

Quality Attributes. As discussed in class, there are many quality attributes that can be important to account for when creating a software system, such as performance, reliability, robustness, scalability, availability, usability, learnability, maintainability, and so forth. Although these may be derived from the requirements, we are not specifying them for this assignment. You may select which quality attributes to evaluate/test for Etherpad.

Testing Tools. Many forms of testing can be done either manually or automatically. For example, UI testing can be done manually (a tester interacts with the program's user interface in a principled way) or automatically with scripts or tool support. There are various tools available to automate various forms of testing. For instance:

- LoadUIWeb (http://loaduiweb.org) helps with stress testing and
- Selenium (http://docs.seleniumhq.org) helps with UI testing.

Task

Your task is to test Etherpad to evaluate at least **three (3) quality attributes.** You must write **two (2) test cases per team member (6-10 total)** to evaluate these quality attributes in Etherpad.

Your test cases must include the use of **at least one (1) automated testing tool** of your choice. Using more than one automated testing tool is also acceptable. The tools mentioned above qualify, though you may use others not listed above. You may also write your own tool or set of scripts to automate a certain type of testing; if so, you must also submit the scripts or code that you wrote.

You may construct tests manually for some of your test cases, but those tests must target a non-functional quality attribute. You may also do UI testing manually by actually "clicking-through" the test case steps. Note that classic unit testing doesn't "count" for this homework because it targets software functionality (i.e., functional requirements), whereas you are asked to test quality attributes (i.e., non-functional requirements). If in doubt as to which forms of testing are appropriate, contact the course staff.

We recommend that each group member targets at least two (2) quality attributes in test case construction, since it will be easier to prepare the individual component of this homework.

Deliverables and Deadlines

There are two (2) deliverables and two (2) deadlines for this homework: a team component and an individual component.

1. Team Component – 70 points – due Tuesday, October 21, 23:59

Create a single PDF document containing the following:

- a paragraph explaining which quality attributes you evaluated, how you evaluated them (including which automated tool(s) you used), and why you evaluated them in that way (why did you choose those tools/approaches?)
- 6-10 test cases (two per team member) adhering to the provided test case guidelines.

If you write your own automated testing scripts, you must also submit the scripts to receive credit for using that particular tool. Submit them together with the PDF in a single ZIP file.

Submit via Blackboard. The name of the document should be of the form "Team X - HW5.pdf" (or Team X - HW5.tar.gz or Team X - HW5.zip if you are also submitting scripts) where X is the letter corresponding to your team name.

2. Individual Component – 30 points – due Thursday, October 23, 23:59

Reflect on your experience of testing the web application from this homework answering the following question: *Has this homework changed your opinion regarding testing?*
You may use the following questions to facilitate your reflection process:

- What did you learn about testing?
- Have you tried testing tools that didn't work out?
- Was there something that you didn't expect?
- How difficult and suitable was the testing you performed?
- Was the performed testing worth the effort?
- How much testing should have been invested?
- How well or poorly did the testing process go?
- How does your experience with this homework relate to your previous testing experience?
- Would you do something differently, if you were to do the testing again?

As in previous homeworks, we will grade the depth of your reflection as well how well you support your arguments with concrete examples from your testing experience for this homework or any previous testing experience you may have had. *You do not have to answer all questions above*, but should instead focus on significant insights. The questions serve as a starting point only.

Your reflection document should not exceed one page (soft limit). Submit it as PDF document via Blackboard. Name document "FirstName LastName - HW5.pdf".

Grading

This homework is worth 100 points. We will grade you based on the learning goals listed above. The test cases constitute 70 points (70%) and the reflection document constitutes 30 points (30%).

To receive full credit for the group component, we expect

- A short description (one paragraph) of which attributes you evaluated and why you chose the tools or techniques you chose,
- 6-10 test cases (depending on team size) each covering all the requested information listed above (name, tool used, quality attributes, specification, justification, steps, test results),
- A clear justification for why the test cases and the selected testing approach is suitable for addressing the selected quality attributes,
- At least three quality attributes addressed, each addressed in at least one of your test cases,
- At least one automated testing tool used in at least one of those test cases
- Clear and reproducible test cases, where each step is documented with a screenshot, and where any scripts or programs you wrote yourself are included in the submission.

To receive full credit for the reflection document (in line with prior reflection assignments), we expect

- A detailed and well-written structured reflection document that answers the question “Has this homework changed your opinion regarding testing?”
- A reflection grounded in your experiences from this homework or any other experience with software testing.
- An analysis beyond superficial statements, mere descriptions, and truisms, which ties specifically to the context of this assignment.
- Substantive arguments behind your opinion. We do not penalize any opinion as long as there is a reasonable argument behind it.

APPENDIX

Sample Test Case

Suppose we are to test Blackboard's usability based on how easy it is to download this homework's description onto the user's computer (for a user who have previously visited the 15-313 course page on Blackboard). In this example, only textual description is given for each step; however, remember that you are to include at least one screenshot for each step.

Test case name:	Downloading this homework's description
Testing tool used:	Manual UI testing
Quality attribute(s):	Usability
Test specification:	The tool is considered usable if it is possible to download this homework's description document from Blackboard's website within 3 mouse clicks.
Justification:	The number of clicks is a usability metric.
Steps:	1. Log in to Blackboard website (https://blackboard.andrew.cmu.edu). 2. Click on the course link ("Foundation of Software Engineering") on the dashboard. 3. Click on the "Homework assignments" link on the left panel. 4. Scroll down the page to find this homework's description. 5. Click on the "hw5.pdf" link under the name of the homework. Result: A new browser tab opened displaying the PDF document. 6. Bring the mouse to the bottom of the browser tab and click on the Download button. Result: The PDF file is save to the Downloads folder.
Test result:	FAIL. It required 4 mouse clicks, instead of 3.
Additional remarks:	1. Tested in Safari 7.0.6. 2. It is assumed that the user have already visited the course page on Blackboard previously and left the left-hand-side panel on the 15-313 course page open.