Assessment in the Software Testing Course

Cem Kaner

Workshop on the Teaching of Software Testing
Melbourne, FL
February 2003

Overview

□ Course overview

□ Approach to assessment

□ Exams

□ Assignments

□ Bonus projects

Course overview

- First course on testing
- Black box (not “behavioral”)
  - Looks at software in context
- Does not assume any particular development method
  - For example, specs optional
- No textbook (this is a weakness)
  - Uses lecture notes and assigned readings
- Goals:
  - Develop a style of critical thinking
  - Develop communication skills
  - Develop testing skills
  - Give students a broad view of practical black box approaches
  - Prepare students to learn more about testing
  - Give students a leg up in their job search
  - Build enthusiasm about the technical work of testing

Course overview

Five core topic areas, which I prioritize as follows:

- ***Paradigms of software testing:*** 9 dominant styles of black box testing.
- ***Bug advocacy***: effective replication, analysis, and reporting of bugs.
- ***Test documentation***:
  - examples of test documentation components;
  - doing requirements analysis to determine what is needed.
- ***Additional test design issues:*** Examples:
  - Design of GUI-level regression tests for maintainability;
  - all pairs combination testing.
- ***Process and organizational issues:*** Examples:
  - structure and missions of typical software testing groups;
  - implications for testing of different software lifecycle models on the testing process.

Assessment-focused approach to teaching

- Student activity more valuable than lecture
- Lecture provides context for activity
  - Story-based, example-based, hypothetical-based
- Students need to practice what they learn
  - Work with sample application
  - Problem of developing practice exercises
- Exams should require answers that show a higher level of competence

Assignments and homework

- Group work is encouraged
- Dealing with cheating:
    - Group work is encouraged
    - Exam-score threshold before I count non-exam work
- Five significant assignments
    - It takes students a long time to learn domain testing
- Not enough homework

Exam structure

- Definitions
- Short answer: essay or worked problem
- Long answer: essay or worked problem

Exam challenges

- Student incompetence in answering essay questions
  - CS students may be worse than average
  - But new law students have problems too
- Student disbelief in study strategy statements
  - Many students seem to have to have a bad experience before they pay attention
- Language skill differential can make for unfairness

Exam structure

- Study guide handed to students in advance
  - Tips on studying and answering questions (which they ignore)
  - Sample questions
    - All questions are drawn from these
    - Now trying (in metrics class) – any number can be changed, but exact structure of questions will be preserved

Exam structure #2

- The pool of exam questions define coverage (anything not listed will not be studied)
- I organize study sessions with/for the students
  - Important to be available, but not to give them answers

--- BREAK OUT TO
STUDY GUIDE FOR
FALL 2002 MID-TERM ---

Exam answer risks

- Weak structure
- Shotgun answers
- Time management in the exam
- Weak preparation
- Weak group preparation
- Weak answers propagate through the group
- Failure to consult required readings
- Study guides too long or too short

Exam Grading

Consider the following question as an example:

- Define a scenario test and describe the characteristics of a good scenario test. Imagine developing a set of scenario tests for the Outlining feature of the word processing module of *OpenOffice*. What research would you do in order to develop a series of scenario tests for Outlining? Describe two scenario tests that you would use and explain why each is a good test. Explain how these tests would relate to your research.

This has several components:

- Define a scenario test
- Describe the characteristics of a good scenario test
- What research would you do in order to develop a series of scenario tests for Outlining?
- Describe two scenario tests you would use.
- Explain why each of the two scenario tests is a good test
- Explain how these two scenario tests would relate to your research
- A well organized answer will have at least six sections, one for each of the bulleted components. You might have two additional sections, by splitting *Describe two scenario tests you would use* and *Explain why each of the two scenario tests is a good test* into two sections, one for each test.

Without structure, it is easy to miss a section and thereby to lose points.

Assignments

- Students *can* work in groups
- First is simplistic (extends Myers)
- Second / third extend boundary analysis (note the *risk* focus

Assignment example

1 Create between 10 and 20 *domain tests*. You can stop at 10 if you find (and write up) 2 bugs. You can stop at 15 tests if you find (and write up) 1 bug.

2 Work in the *Word Processing* part of Open Office.

3 Pick a function associated with Word Processing. Please run all of your tests on the same function. (If several students are working together, you can pick one function per student.)

4 Pick one (1) input, output, or intermediate result variable

- Identify the variable. Stick with that one variable throughout testing.
- Run a mainstream test (a test that is designed to exercise the function without stressing it). You do tests like this first in order to learn more about the function and the variable's role in that function.

5 Identify risks associated with that variable

Assignment example

6 For each risk, design a test specifically for that risk that is designed to maximize the chance of finding a defect associated with this risk.

7 Explain what makes this a powerful test. It might help you to compare it to a less powerful alternative.

8 What is the expected result for the high-power test?

9 What result did you get?

10 Report your results in a table format that has the following columns:

- □ **Feature or function**
- □ **Variable name or description**
- □ **Risk**
- □ **Test**
- □ **What makes this test powerful**
- □ **Expected result**
- □ **Obtained result**

Assignment example

11 If you find bugs, write up bug reports and enter them into Issuezilla.

12 *I strongly recommend that you pair up with someone and have them replicate your bug and evaluate a draft version of your report before you submit it to Issuezilla. I will evaluate your report against a professional standard of quality (essentially, the same evaluation that you just did in Assignment 2).*

13 Write a summary report that explains what you believe you now know and don't know about the function, based on your testing. (If your group tested several functions, write up a summary report for each.)

Assignment example

Notes (that I’ll use for grading) on Exercise 2

- It’s important to answer every section:
  - The table needs 7 columns
  - There should be 10-20 tests and 0-2 bug reports
  - There should be a summary report that explains what you know about the function under test.
- It’s important to show the domain analysis (or its results)
  - Use boundary values
  - Identify them as bounds and equivalence classes or identify the different sections of the space as you partition it. You might find it useful to start with a boundary analysis (and table).
  - Be specific about risk
  - Be specific about power (compare to others of the same equivalence class)