Fall 2013

CEG 3110/5110-01: Introduction to Software Testing

John A. Reisner
Wright State University - Main Campus, john.reisner@wright.edu

CEG 3110/5110: Introduction to Software Testing

Fall Quarter, 2013

Course Description and Goals

This course covers software testing strategies, along with established best practices, to teach students how to test software in a complete and systematic (vice ad-hoc) manner. Particular attention is paid to planning, writing, and executing software tests, along with associated documentation, (i.e., a software test plan), which includes documented results. Various projects are assigned, designed to illustrate various challenges associated with software testing, and to reinforce the strategies and techniques used to overcome these challenges.

Textbook & Reading Assignments

The course text is Paul C. Jorgensen's *Software Testing: A Craftsman's Approach*, Auerbach Publications, 2008, ISBN 0-8493-7475-8. This is a required textbook for this course.

Some lessons may have corresponding reading assignments. The course lectures are designed to *augment* (not simply *rehash*) these readings. Put another way, the book is designed to supplement the overall learning experience for the student.

Course Projects & Lectures

This will be a "learning-by-doing" class. Students will have a series of projects throughout the course, where they will write code, write test plans, execute test plans, and document results. These assignments are rather unique in that grading is based primarily on how well the code is ***tested***, as opposed to how well it is written.

Class time will consist of interactive discussions. Students should attend class ready to contribute through active participation. **No open laptops are allowed in class**, unless they are being used as part of a testing exercise. Class time is devoted to learning, not browsing.

Instructor Contact Info

John Reisner
Office hours after class or by appointment
Daytime Phone: 255-3636 x7422 (this is a WPAFB phone number).
email: **john.reisner@wright.edu** (for a more timely response, consider sending a CC: to **jreisner@afit.edu**).

The instructor is an adjunct faculty member. Most contact will be done via email, phone, or during before- or after-class discussions. Other meetings can be arranged as needed.

Course Objectives

By the end of this course each student should be able to:

1. Write appropriately comprehensive test plans.
2. Effectively document test plans and results.
3. Develop software using a test-driven approach.
4. Employ effective testing strategies for different needs.
5. Write drivers, stubs, and testware as needed to sufficiently test a program.
6. Verify a program's correctness via a test strategy.

Course Components

Being a higher-level college course, my goal is to teach at the higher levels of **Bloom's Taxonomy**. Overall, the goal of this course's homework assignments, projects, and exams are to stimulate and promote ***thought***, particularly at higher levels of the cognitive and affective domains.

35% Course Projects

- There will be a series of programming projects. The emphasis of these projects will be testing software by using a written test plan.
- Most projects will be extended over two or three weeks, to give students sufficient time to (a) generate requirements, (b) write a test plan, (c) write and debug code, and (d) execute the test plan.
- Each project will be graded individually. Although the grade will be based primarily on the thoroughness and quality of the test plan, students are expected to use good programming style and employ accepted programming practices throughout the course.
- The student may select the programming language used for each project.
- Projects are to be submitted in paper form, to include code listings.

20% Midterm Exam

- Mixed-format exam, administered in class. Exams generally consist of 7 multiple-choice questions (28 points), plus some (usually between four and eight) short answer, essay, or analysis questions (72 points).
- Any of the material covered in assigned readings, or in-class discussions is considered "testable." That said, tests are designed to focus on and reiterate important and fundamental concepts, rather than the minutiae.
- Many exam questions are designed to test the ability of the student to analyze, synthesize, and eloquently explain information and concepts, rather than recall simple facts. You may be asked to argue a certain position; your argument is expected to be sound, to demonstrate that you are learning in the course, and to demonstrate that you can state your position logically.

20% Final Exam

- Comprehensive, mixed-format exam, administered during the school's final exam week. This exam will resemble the midterm (although the questions will be different, of course).

20% Homework Assignments

- Homework assignments are designed to facilitate deeper comprehension about a lecture topic (in other words, these are "think and respond" assignments).
- There may be up to two assignments per week, but some weeks may have one or zero assignments. Most weeks will not have more than one.
- Homework assignments are different from the weekly projects.
- All assignments are turned in via hardcopy. Further details about each assignment will be posted on Pilot.
- Normally, these assignments will be due on Monday each week (thus, students have one week to complete a Monday assignment and five days to complete a Wednesday assignment). Any exceptions to this policy will be mentioned when the homework is assigned.

5% Class Participation

- Based on attendance, participation, attitude, readiness to learn, and willingness to share thoughts and ideas.