school of computing, informatics, & decision systems engineering

Design and Teaching of a Senior Year Software Quality Assurance and Testing Course (CSE 464/598)

WTST 2014

Janaka Balasooriya
janakab@asu.edu

Acknowledgements

- Course design has been done with the help of Dr. James Collofello (Professor of Software Eng. and Associate Dean, Fulton Schools of Eng. at ASU)

- Many thanks to Dr. Cem Kaner (Professor of Software Eng. At Florida Tech) for giving his thoughts during the course design process and letting us use BBST material

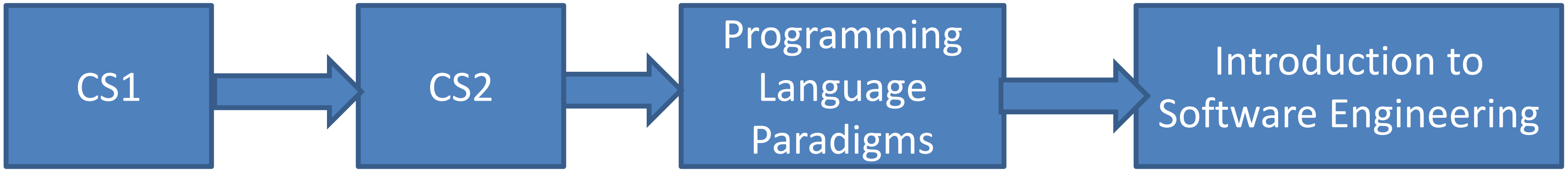
- Several Software Eng Faculty at ASU and some of our industry partners reviewed the course syllabus and provided valuable feedback during the course design process.

Some Background

- Our Rationale
  - Software quality is seen as the foundation towards engineering the “software product” by many.
  - Computer science and Computer Engineering students at undergraduate level should acquire sufficient pedagogical background in software engineering including software quality.
  - Testing software to uncover defects, fixing defective code, review code/design and other software engineering artifacts should be taught across CS/CSE curriculum.
  - CS1 and CS2 classes covers some code-based and functional testing techniques. Most CS students take an introductory level software engineering class in the second year of their study. Most of these introductory level software engineering classes have a medium size semester long group project. This is a place where software design level quality factors in addition to code-based testing can be incorporated.
  - Such a background (even if students do not have any industry experience) will help students to take a course that discusses software testing theory and practice at bit more detailed level. Such a course will enable students to learn some important (commonly used) software testing techniques in-depth and also breadth of software testing.

Intended Audience

- Our CS program has several concentrations including **software engineering concentration**. CSE 464 is a required course for software engineering concentration students while general CS /CSE students take the class as a **technical elective**.
- Intended audience for our software quality assurance and testing course have taken following sequence of programming language/software engineering classes (among other classes).

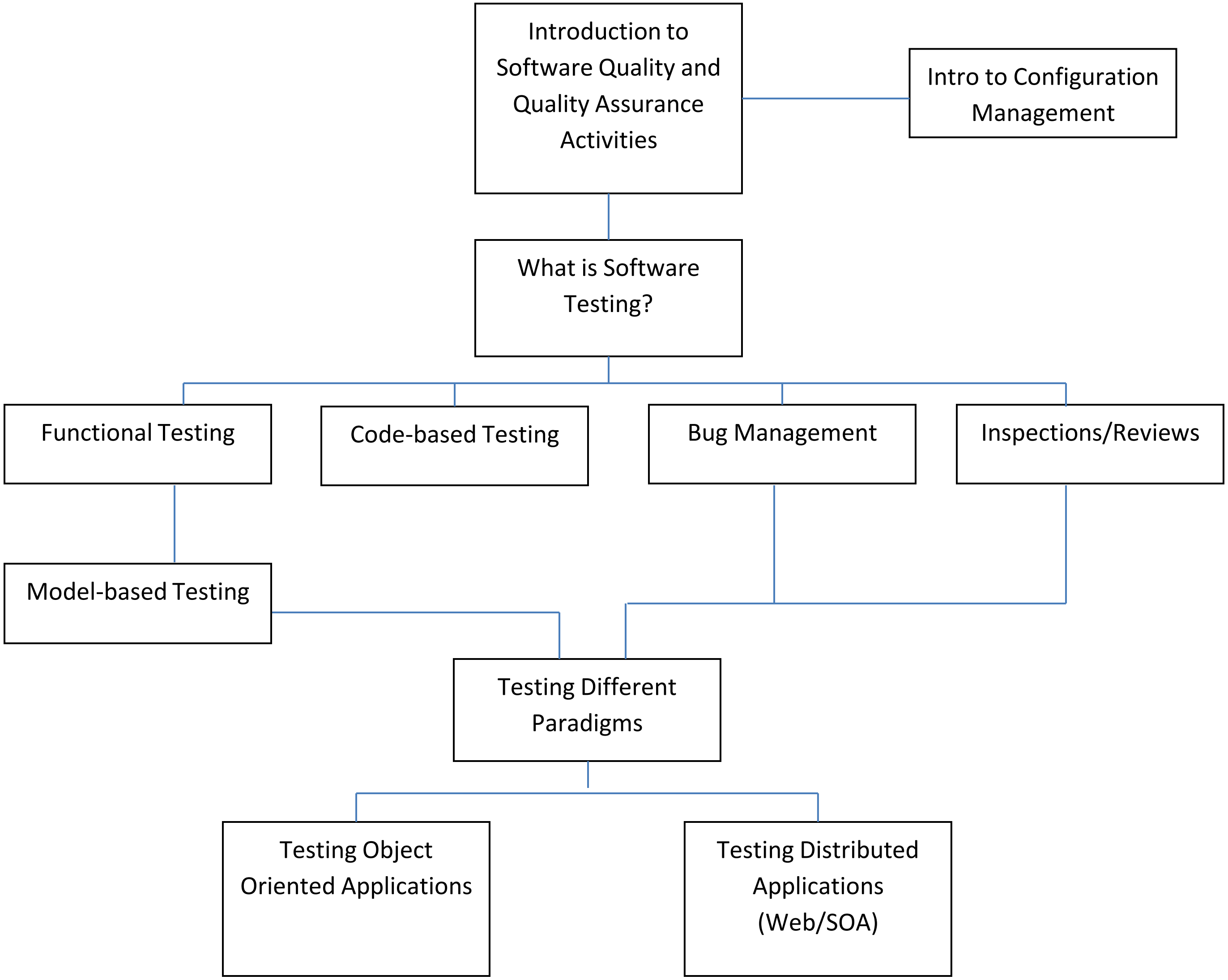

CS1

CS2

Programming Language Paradigms

Introduction to Software Engineering

Subject Coverage



Introduction to Software Quality and Quality Assurance Activities

Intro to Configuration Management

What is Software Testing?

Functional Testing

Code-based Testing

Bug Management

Inspections/Reviews

Model-based Testing

Testing Different Paradigms

Testing Object Oriented Applications

Testing Distributed Applications (Web/SOA)

Course Objectives and Outcomes

We have set course objectives and outcomes for this course to give students a **balance of theory and practice in a standard academic setting**. We specially want to make sure that testing is not mechanical, but it's an intellectual agile process that needs careful consideration of the context.

After taking this course, students should be able to

- explain fundamental SQA considerations and understand the role of testing in SQA
- explain the basis of various approaches in unit, integration, and system testing
- understand how testing is applied throughout the software life cycle
- able to use an automated testing tool to write test cases and perform testing
- able to perform inspections, reviews, and walkthroughs of code and design of a large software system
- carryout software testing for a complex software using testing tools
- explain basic principles of software configuration management
- gain a solid understanding of state-of-the-art in software testing

Lecture Format

Lecture Format:

In addition to lectures, the class consists of **lecture supplemental activities**. Those activities are **completed in-class as part of the instruction**. Some activities for which coding/testing needed are take-home.

Also, class consists of **visiting lectures from our industry partners** that give students an idea of how testing is done in the real-world and relate what course material to industry usage.

Required Textbook:

Software Testing: A Craftsman's Approach, Third Edition, Paul C. Jorgensen, *Grand Valley State University, Allendale, Michigan, USA*, 2008, ISBN**:** 9780849374753, CRC press.

This is more of a theoretical book gives in-depth analysis of applicability, limitations, and strengths of well-known testing principles. Specially, the book has good comparisons of related techniques and discussed using several examples such as the triangle problem and next-date problem.

Using BBST material:

While covering the theory, I believe that the idea of context-aware testing is an important concept we need to inculcate in the minds of students. Students in this class, listen to exploratory testing lectures developed Prof. Cem Kaner at Florida Tech and complete a three phased assignment testing open office writer using exploratory testing to find a bug and writing a bug report.

Assessment

Class assessments are done through various activities.

- Simplest form is **in-class activities** that simply assess the knowledge of students on a particular topic. This is somewhat equivalent to the lowest layer of learning from Bloom's taxonomy.

- **Home works** asses the comprehension and application, and analysis of subject matter that is equivalent to the second, third, and fourth layers of learning from Bloom's taxonomy.

- **Exams** are mainly covering knowledge, comprehension and sometimes application.

Bug Management Assignment

Bug Management Assignment (adopted from Dr. Cem Kaner's Assignment on Replicate and Edit Bugs Assignment of Bug Advocacy Course)

The purpose of this assignment is to give you experience reviewing bugs written by other people. This task will give you practice thinking about what a professional report should be, before you start wring your own bug reports. This assignment is a multi-phased assignment with three phases. Main reason for this to be a multi-phased assignment is to give you a best possible exposure in bug management and understanding of the bug workflow.

Phase I: Evaluating a Bug Report in Open Office Bug Management System

Phase II: Writing your own bug report (test writer using exploratory testing and identify a bug and then write a bug report)

Phase III: Confirming a bug reported by your classmates.

Code-based Testing Homework Example

Introduction: This assignment helps students to reinforce the topics discussed in the class including

Basic Concepts of Structural Testing, Control Flow Graph and Basic Path Testing, Data flow and D-U Path Testing Also, another goal of this assignment is to illustrate a situation where D-U path testing is more applicable than control path testing. Finally, students will understand relationship between **testability of code with better coding practices**.

1. [50 Points] Suppose you want to perform basic control path testing and D-U path testing for the Java program ***(modified from homework #2)*** that calculates the change after making a payment for a purchase.

a) [15 Points] Draw the control-flow graph and annotate the graph nodes with def and use terminology for all the variables

Submit: clearly labeled D-U graph.

b) [15 Points] Identify D-U for variables dollars, quarters, dimes, nickels, pennies, and change.

Submit: D-U pairs for each variable in a table as we have discussed in the class, D-U paths for each variable.

c)[20 Points] Now design test cases based on your D-U Paths above (c) and implement them in Junit.

Submit: Test cases in a table format and your Junit tests

How do Students Feel About the Course?

Overall student’s feedback for the course is excellent. A good number of our students at this point work either fulltime or part-time. Many indicated that some techniques that we have discussed in the class are applicable and related to their work. Also, enrollment numbers have grown from sixty students to hundred within last three years. Here are some of the student comments

“Practical and hands-on material. Everything feels relevant to real world applications.”

“I didn't like the amount of work assigned for it was quite much but again it helped grasp the material better.”

“Lectures and assignments can be a little unclear in their requirements sometimes.”

Potential Improvements

At this point, most of our assessment falls within first three layers of the Bloom’s taxonomy of learning.

Developing a good class project and required infrastructure such as a test-bed that challenges students to achieve higher level of learning is a good improvement.

Thank You!

QA