C S 82A: INTRODUCTION TO SOFTWARE QUALITY ASSURANCE

Foothill College Course Outline of Record

Heading	Value
Effective Term:	Summer 2021
Units:	4.5
Hours:	4 lecture, 2 laboratory per week (72 total per quarter)
Advisory:	Knowledge of an object-oriented programming language.
Degree & Credit Status:	Degree-Applicable Credit Course
Foothill GE:	Non-GE
Transferable:	CSU
Grade Type:	Letter Grade (Request for Pass/No Pass)
Repeatability:	Not Repeatable

Student Learning Outcomes

- Learn different testing techniques to test software using methods and tools.
- Write a QA test plan that reveals defects in source code

Description

Introduction to software quality assurance principles, techniques, processes and tools. A team project takes students through the planning and implementation of the test and release of a software product using a current toolset.

Course Objectives

The student will be able to:
A. List the costs and benefits of following a quality assurance process in the development of a software product.
B. Exhibit professional behavior in working with various stakeholders.
C. Write a QA test plan containing tests of various types and test cases that reveal defects in source code.
D. Take a project through the various stages of software testing.
E. Devise tests that determine if software fulfills different types of requirements.
F. Use a source code control system to track changes and integrate software modules.
G. Use test automation software.
H. Use virtualization software to simulate different configurations and platforms for test execution.
I. Plan and implement a product release.

Course Content

A. Principles of Quality
1. Benefits and costs
2. Legal issues
3. Standards and models
B. Professional Behavior
1. Ethics
2. Communication
3. Teamwork
4. Stakeholders
C. Types of Tests
1. Unit
2. Integration
3. Functional
4. Performance
5. Regression
6. Usability
7. Acceptance
8. Certification
9. Environmental load
10. Stress
11. Black-box
12. White-box
13. Time-box
14. Good-enough
D. Techniques
1. Software development lifecycle models including: waterfall, iterative, test driven development, agile, scrum
2. Test case repository including: worst-case, boundary values, equivalence class partitioning
3. Review of design documentation for completion
4. QA project plan
5. QA test plan
6. Build schedules
7. Defect prevention strategies
8. Fault-insertion, fault-error handling
9. Code freeze
10. Feature freeze
11. Alpha test cycle
12. Beta test cycle
13. Project retrospective
E. Requirements Engineering
1. Categories of requirements: feature, function, system, quality, security, regulatory, performance, acceptance, usability
2. New, acquired and legacy systems
3. Change management
4. Use cases
5. Servers/environments to test
F. Source Code Control Systems
1. Change tracking
2. Source forking
3. Concurrent development
4. Code integration
5. Software builds, versioning schemes, baselines
G. Test Automation
1. Code driven testing
2. User interface testing
3. Code analysis and code coverage
4. Metrics
5. Diagnostics
H. Virtualization
1. Testing on multiple platforms
2. Simulating multiple users
3. Embedded systems, client-server, wireless
4. Configuration management
I. Product Release
1. Planning for product release
2. Defining hardware and software dependencies

3. Archival process for build environments

Lab Content

A. Design, document and carry out a test plan containing various different types of tests
B. Plan a project that cycles through various stages of testing
C. Review a requirements document for completeness
D. Use a source code control system to integrate separate source code modules and test
E. Use a bug tracking system to report on the status of a software development project
F. Use virtualization software to test a new software product on multiple platforms
G. Write code to automate the testing of successive releases of a software product
H. Identify hardware and software dependencies for a new product release

Special Facilities and/or Equipment

A. Access to a computer laboratory with a source code control system.
B. Computer laboratory with virtualization software that is configurable by students.
C. Computer laboratory with test automation software installed.
D. A website or course management system with an assignment posting component (through which all lab assignments are to be submitted) and a forum component (where students can discuss course material and receive help from the instructor).
E. When taught via Foothill Global Access on the Internet, a fully functional and maintained course management system through which the instructor and students can interact.
F. When taught via Foothill Global Access on the Internet, students must have currently existing email accounts and ongoing access to computers with internet capabilities.

Method(s) of Evaluation

Tests and quizzes
Written laboratory assignments which include source code, sample runs and documentation
Team project

Method(s) of Instruction

Lectures
Online labs (for all sections, including those meeting face-to-face/on campus), consisting of:
1. An assignment webpage located on a college-hosted course management system or other department-approved internet environment. Here, the students will review the specification of each assignment and submit their completed lab work
2. A discussion webpage located on a college-hosted course management system or other department-approved internet environment. Here, students can request assistance from the instructor and interact publicly with other class members
Team project
When course is taught fully online:
1. Instructor-authored lecture materials, handouts, syllabus, assignments, tests, and other relevant course material will be delivered through a college-hosted course management system or other department-approved internet environment
2. Additional instructional guidelines for this course are listed in the attached addendum of CS department online practices

Representative Text(s) and Other Materials

Murali, Chemuturi. Mastering Software Quality Assurance. 2010.

Goericke, Stephen. The Future of Software Quality Assurance. 2019.

Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

A. Reading
1. Textbook assigned reading averaging 30 pages per week.
2. Reading the supplied handouts and modules averaging 10 pages per week.
3. Reading online resources as directed by instructor though links pertinent to programming.
4. Reading library and reference material directed by instructor through course handouts.
B. Writing
1. Technical prose documentation.
2. Business correspondence to collaborate with team members on project.

Discipline(s)

Computer Science