**Software Testing**

**Instructor Information**

Name: Lehmiakiran

E-mail: lehmia.kiran@nu.edu.pk

**Course Information**

Prerequisite: Software Engineering

Level: BS

Category: Elective

Credit Hours: 3

Semester: Fall 2016

Class Venue: CS-4

**Course Objectives**

Software testingis an investigation conducted to provide stakeholders with information about the quality of the product or service being tested. Software testing also provides an objective view of the system quality to allow businesses to appreciate the risks associated with implementation of the system. This course focuses on testing techniques and tools for software systems. Students will learn importance of testing, testing techniques, testing process and selected open source testing tools.

**Text Book:**

**Naik and Tripathy, Software Testing and Quality Assurance—Theory and Practice, Wiley, 2008.**

**Reference Book:**

* Anne MetteJonassen Hass, *Guide to Advanced Software Testing*, Artech House, 2008.
* Lee Copeland: *A Practitioners Guide to Software Test Case Design,* Artech House Publishers, 2004
* Julian Harty, *A Practical Guide to Testing Wireless Smartphone Applications*, Morgan and Clay Pool, 2010.

**Course Grading**

* Quizzes: 10%
* Project: 10%
* Assignment 10%
* Midterm Exams: 30%
* Final Exam: 40%

**[[1]](#footnote-2)Course Schedule**

|  |  |  |
| --- | --- | --- |
| **S#** | **Topic** | **Week(s)** |
| 1 | **Lecture #1:**   1. Software Quality 2. Role of Testing 3. Veriﬁcation and Validation 4. Failure, Error, Fault, and Defect 5. Notion of Software Reliability 6. Objectives of Testing   **Lecture #2 :**   1. Central Issue in Testing 2. Testing Activities 3. Test Levels | **1** |
|  | **Lecture #3:**   1. White-Box and Black-Box Testing 2. Test Planning and Design 3. Monitoring and Measuring Test Execution Test Tools and Automation 4. Test Team Organization and Management   **Lecture #4:**   1. Writing a test case 2. Equivalence classes 3. Boundary value analysis | **2** |
|  | **Lecture #5**   1. Concept of Unit Testing 2. Static Unit Testing 3. Defect Prevention 4. Dynamic Unit Testing   **Lecture #6**   1. Mutation Testing 2. Debugging 3. Unit Testing in eXtreme Programming | **3** |
|  | **Lecture #7**   1. JUnit: Framework for Unit Testing Tools for Unit Testing | **4** |
|  | **Lecture #8:**   1. Control Flow :Basic Idea 2. Outline of Control Flow Testing 3. Control Flow Graph 4. Paths in a Control Flow Graph 5. Finding cyclomatic Complexity   **Lecture #8**   1. Path Selection Criteria 2. All-Path Coverage Criterion 8. 3. Statement Coverage Criterion 4. Branch Coverage Criterion | **5** |
|  | MIDTERM EXAM 1 | 6 |
| 4 | **Lecture #9**   1. Predicate Coverage Criterion 2. Generating Test Input 3. Examples of Test Data Selection 4. Containing Infeasible Paths   **Lecture #10:**   1. Data Flow Testing General Idea 2. Data Flow Anomaly 3. Overview of Dynamic Data Flow Testing 4. Data Flow Graph 5. Data Flow Terms | 7 |
| 5 | **Lecture #11:**   1. Data Flow Testing Criteria 2. Comparison of Data Flow Test 3. Selection Criteria 4. Feasible Paths and Test Selection Criteria 5. Comparison of Testing Techniques   **Lecture #12**   1. Concept of Integration Testing 2. Different Types of Interfaces and Interface Errors 3. Granularity of System Integration Testing | 8 |
| 6 | **Lecture #13 :**  1. System Integration Techniques 1.1 Incremental  1.2 Top Down  1.3 Bottom Up  **Lecture #14:**  1. Sandwich  2. Big Bang | 9 |
| 7 | **Lecture #15:**   1. Test Plan for System 2. Integration Off-the-Shelf 3. Component Integration 4. Off-the-Shelf Component Testing 5. Built-in Testing   **Lecture #16**   1. **Taxonomy of System Tests** 2. **Basic Tests** 3. **Boot Tests** 4. **Upgrade/Downgrade Tests** 5. **Light Emitting Diode Tests** 6. **Diagnostic Tests** 7. **Command Line Interface Tests** | 10 |
|  | MIDTERM EXAM 2 | 11 |
| 9 | **Lecture #17:**   1. Functionality Tests 2. Communication Systems Tests 3. Module Tests 4. Logging and Tracing Tests 5. Element Management Systems Tests 6. Management Information Base Tests 7. Graphical User Interface Tests   **Lecture #18:**   1. Security Tests 2. Feature Tests 3. Robustness Tests 4. Boundary Value Tests 5. Power Cycling Tests 6. On-Line Insertion and Removal Tests 7. High-Availability Tests 8. Degraded Node Tests 9. Interoperability Tests 10. Performance Tests 11. Scalability Tests 12. Stress Tests 13. Load and Stability Tests 14. Reliability Tests 15. Regression Tests 16. Documentation Tests 17. Regulatory Tests | 12 |
|  | Lecture #19:  **Automated Testing tools**  Lecture #20:  **Automated testing tools II** | 13 |
| 10 | **Lecture #21**  Decision table  **Lecture #22:**  Pairwise combination testing | 14 |
| 11 | **Lecture #23:**  Object-oriented testing  **Lecture #24:**  Web based Testing | 15 |
| 12 | **Lecture #25:**  GUI testing  **Lecture #26:**  **Summarizing Testing Activities:**   * Planning * Test case generation * Testing environment preparation * Execution * Test results reporting and tracking | 16 |
|  | **Lecture #27:**  **Testing Tools Guide** (Test Management Tools—BugZilla)  **Lecture #28:**  Project Discussion | 17 |
|  | FINAL EXAM |  |

**Important Information**

**Attendance**

All students are expected to attend all lectures from beginning to end. Students who arrive more than 10 minutes late will not be allowed to enter the classroom. Partial or full absence from a lecture without a valid reason may hamper chances for securing good grades.

**Exams**

Exams will be closed-book and closed-notes.

**Late-Submission**

Late submission of project deliverables is NOT allowed. Teams that do not meet the submission deadline of a deliverable will not get any credit in that particular deliverable.

**Project**

Project teams comprising 2 students will be formed by the instructor herself using a process of random selection. Projects can be chosen from any domain.

**Quizzes**

Quizzes may be announced or unannounced. A quiz will usually be about 5 – 10 minutes long and it may be given anytime during the lecture. Students missing a quiz will NOT be given a make-up quiz.

**Reading Material**

Students are encouraged to finish the assigned readings BEFORE the lecture. This is likely to improve lecture comprehension and class participation.

**Revision of Grades**

Students can contest their grades on quizzes and project deliverables ONLY within a week of the release of grades. Exams will be available for review according to university policies.

**Unfair Means**

Students are expected to demonstrate the highest degree of moral and ethical conduct. Any student caught cheating, copying, plagiarizing, or using any other unfair means will be strictly dealt-with in accordance with university policies.

1. Course schedule is subject to change. [↑](#footnote-ref-2)