08B51CI101: Software Testing and Debugging

Course Credit: 4
Semester: V

Introduction

This course will examine fundamental software testing and program analysis techniques. In particular, the important phases of testing will be reviewed, emphasizing the significance of each phase when testing different types of software. Students will learn the state of the art in testing technology for object-oriented, component-based, concurrent, distributed, graphical-user interface, and web software. In addition, closely related concepts such as mutation testing and program analysis (e.g., program-flow and data-flow analysis) will also be studied. Emerging concepts such as test-case prioritization and their impact on testing will be examined. By the end of this course, students should be familiar with the state-of-the-art in software testing. Students should also be aware of the major open research problems in testing.

Course Objectives (Post-conditions)

Knowledge objectives:

You will broaden your knowledge of software engineering.
2. You will learn Software testing algorithms and programs.
3. You will increase your proficiency in JAVA Language.
4. You will know how strategies and tactics of effective and efficient testing.
5. You will gain practical experience in design, develop, and document static, white-box, black-box tests.
6. You will acquire the background for understanding Test Management and Software Development.
7. You will acquire the knowledge of higher order and object oriented testing.
8. You will learn a effective and efficient use of debugging techniques.
9. You will learn how to make use of Web testing and Automated software testing.

Application objectives:

1. To develop, implement black box and white box testing cases.
2. To understand use of Flow graphs and computing cyclomatic complexity using various methods.
3. To understand and implement automated software testing techniques for Web testing, Performance testing, and GUI testing.
4. To develop, implement, and demonstrate the learning through a project that meet stated specifications.
5. Debugging of existing program codes and developing test cases.

Expected Student Background (Preconditions)

Students should be familiar with programming concepts.

Topics Outline:

S NO	Topics	Hrs
1	The Psychology and Economics of Program Testing: What is software testing and why it is so hard? , Error, Fault, Failure, Incident, Principles of Testing, Testing Process (Economics of Testing), Limitations of Testing, No absolute proof of correctness,	2
2	Test Management & Software Development – Software Quality Assurance, ETVX Model, Testing Maturity Model, Testing Life – Cycle, Testing V-Model, Test Planning & Control, Test Completion Criteria, Designing Test case templates	2
3	Static Software Testing & Techniques: Program Inspection, code inspection, An error checklist for inspections, Walkthroughs, Desk checking, peer rating Verification, Validation. Test Case Design – I: Using Black Box Testing- Equivalence Class Testing, Boundary Value Analysis, Decision Table Based Testing, Cause Effect Graphing Technique, Experience based techniques, State based or graph based testing.	2+7
4	Test Case Degign – II: Using White Box testing- Path testing, DD-Paths, Cyclomatic Complexity, Graph Metrics, Data Flow Testing, Mutation testing, Fault Injection, Pairwise testing, Predicate based test generation, Module Testing: Integration Testing, Top down versus Bottom up Testing, Test case design and Performing the Test, Integration Complexity	7+ 4
5	Higher Order testing: Function Testing, System Testing, Acceptance Testing, Installation Testing, Performance testing, Database Testing, Regression testing, Ad hoc Testing, Test planning and control, Test completion criteria.	5
6	Object Oriented Testing: Issues in Object Oriented Testing, Class Testing, Object Oriented Integration and System Testing.	5
7	Web Testing/GUI testing and Internet Testing Issues, challenges, Automated test oracles for GUIs, Regression testing of GUIs, Mobile Application testing and Tools, Techniques and Levels, Load	3

	testing	
8	Program Instrumentation: (from e-book of J.C. Huang) Test Coverage Measurement, Test Case Effectiveness Assessment, Instrumenting Program for Assertion Checking, Instrumenting Program for Data Flow-Anomaly Detection.	2
9	Debugging: Debugging and its Techniques, Taxonomy of Bugs, Bug Life – Cycle, Debugging Techniques	3
	Total	42

References

1. Paul C. Jorgensen, Software testing: a Craft's man approach, CRC Press Ilene Burnstein, Practical Software Testing, Springer
2. Srinivasan Desikan and G. Ramesh, Software Testing: Principles and Practices, Pearson Education
3. Glenford Myers, "The Art of Software Testing", John Wiley & Sons Inc., New York, 1979.
4. Aditya P. Mathur, "Foundations of Software Testing" Pearson Education 2008
5. Louise Tamres, "Software Testing", Pearson Education Asia, 2002
6. William Perry, "Effective Methods for Software Testing", John Wiley & Sons, New York, 1995.
7. Cem Kaner, Jack Falk, Nguyen Quoc, "Testing Computer Software", Second Edition, Van Nostrand Reinhold, New York, 1993.
8. Boris Beizer, "Software Testing Techniques", Second Volume, Second Edition, Van Nostrand Reinhold, New York, 1990.
9. Roger S. Pressman, "Software Engineering – A Practitioner's Approach", Fifth Edition, McGraw- Hill International Edition, New Delhi, 2001.
10. Boris Beizer, "Black-Box Testing – Techniques for Functional Testing of Software and Systems", John Wiley & Sons Inc., New York, 1995.
11. K.K. Aggarwal & Yogesh Singh, "Software Engineering", New Age International Publishers, New Delhi, 2003.
12. Marc Roper, "Software Testing", McGraw-Hill Book Co., London, 1994.
13. Gordon Schulmeyer, "Zero Defect Software", McGraw-Hill, New York, 1990.
14. Watts Humphrey, "Managing the Software Process", Addison Wesley Pub. Co. Inc., Massachusetts, 1989.
15. Boris Beizer, "Software System Testing and Quality Assurance", Van Nostrand Reinhold, New York, 1984.

Evaluation Scheme:

S.No	Examination	Marks
1	T-1	15
2	T-2	25
3	T-3	35
4	*Internal Marks	25

*Internal Marks Breakdown:

Assignments 9 marks (3x3)

Quizzes 12 marks (3x4)

Regularity 4 Marks