

## CS 5387 – Software Integration and V&V Course Information Sheet - Spring 2022

**Instructor:** Dr. Bhanukiran Gurijala

Phone: 915-747-5827

**office:** Prospect Hall 106 C

**email:** [bgurijala@utep.edu](mailto:bgurijala@utep.edu)

**Office Hours:** Monday and Wednesday 10:00-11:30

**Course Description:** This course will provide students with an in-depth study of software validation and verification. Topics include the limits of testing, professional responsibility and liability, testing, management of the testing process, automated testing tools, coverage metrics, software quality, non-testing quality assurance, static and dynamic validation techniques, inspections, and audits.

### **Prerequisites:**

- Knowledge of software development life cycles
- experience in the development of software-reliant systems
- some familiarity with modern software engineering concept

### **Supporting Text Material:**

- Aditya Mathur, *Foundations of Software Testing*, Pearson Publishing, 2008.
- Paul Jorgensen, *Software Testing, A Craftsman's approach*, Auerbach Publications, 2007
- Lee Copeland, *A practitioner's Guide to Software Test Design*, STQE Publishing, 2004

### **Course Purpose**

The objective of this course is to prepare students to understand the role of ensuring quality in software development, recognizing its role from the elicitation, analysis, and specification of requirements through to the delivery and operation of a software system. The focus will be achieving quality through sound software engineering practices. Students will learn techniques for analyzing software artifacts throughout the life cycle and how to incorporate these techniques in software development activities to help ensure quality.

### **Learning Outcomes**

#### **Level 1: Knowledge and Comprehension**

Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. The material has been presented only at a superficial level.

Upon successful completion of this course, students will be able to:

- 1a. Know the purpose of and differentiate among audits, inspections, and walk-throughs.
- 1b. Define quality attributes such as availability, correctness, efficiency, interoperability, maintainability, portability, reliability, reusability, security, simplicity, testability, and usability.
- 1c. Describe different categories of test coverage of code.
- 1d. Describe different test levels (including unit, integration, system, acceptance, regression, installation, alpha and beta, performance, stress, usability, and reliability).
- 1e. Describe the test development techniques including those based on ad-hoc, exploratory, equivalence class, boundary value, data flow, and control flow.
- 1f. Discuss the main ethical considerations related to software assurance.
- 1g. Describe the differences between traditional and object-oriented testing.

## Level 2: Application and Analysis

Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details.

Upon successful completion of this course, students will be able to:

- 2a. Conduct a technical review including inspections, walkthroughs and audits.
- 2b. Use software testing tools and frameworks for verification and validation tools.

## Level 3: Synthesis and Evaluation

Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery.

Upon successful completion of this course, students will be able to:

- 3a. Develop white-box and black-box test cases.
- 3b. Conduct unit and integration testing.
- 3c. Develop a test plan for a large system.
- 3d. Develop test cases that meet white box coverage criteria such as branch, condition, and def-use coverage.
- 3e. Develop test cases based on decision tables.
- 3f. Develop test cases that meet functional coverage criteria.

## Assignments:

Assignments consist of assigned reading, exercises, special reports, and working as part of a team developing a software product. Graduate Students will also be required to deliver class presentations

## Grading Procedures:

1. Each student will work on a team project to gain hands-on experience in software testing, V&V, and quality assurance practices. Each team will be required to deliver a set of artifacts associated with the different level of software testing. The individual student grade on the project will be a combination of an evaluation of the overall completeness and quality of the product deliverables, project presentations and reviews, and the contribution of the individual to the team effort.
2. Individual and team in-class exercises, covering recent assignments and classroom work, will be given on regular basis.
3. Two full period exams will be given during the term.
4. Individual and team homework will be assigned as necessary.
5. Participation in class discussion and attending classes is a must in this course
6. Points will be distributed as follows:

Team Project	25%
Exercises, Homework, etc.	15%
Exam 1	20%
Exam 2	20%
Participation	10%
Research Work Deliverable and Presentations	10%

7. Anyone found cheating on an exam will receive an automatic F in the course.
8. There are no make-ups on in-class exercises. Absence from an exam is excused only in a medical emergency.

## CS 5387 – Spring 22 - CLASS SCHEDULE

Week #	Topic
1	Introduction to Software Engineering Introduction to Software V&V
2	Software formal reviews and inspections
3	Software Inspections: Requirements, Design, and Code Inspections
4	Introduction to Software Testing and System Test Planning
5	Unit Testing: Black-box Testing
6	Continue: Unit Testing: Black-box Testing
7	<b>EXAM 1</b>
8	<b>Spring Break: No Class</b>
9	Unit Testing: White-box testing
10	Unit Testing: White-Box Testing
11	Integration Testing
12	System Testing
13	Usability Testing
14	V&V for parallel and distributed systems: Formal Verification Techniques
15	V&V: for non-functional quality attributes (eg. Security, performance,
16	reliability...) System Security Assurance Tools (IBM QRadar)
17	<b>EXAM 2</b>