SOFTWARE TESTING METHODOLOGIES

(Professional Elective-IV)

COURSE CODE: 19IT160

Pre-requisites: Software Engineering

COURSE OUTCOMES:

At the end of the Course, the Student will be able to:

CO1: Classify different software testing techniques.

CO2: Distinguish the characteristics of Path Testing and Dataflow Testing.

CO3: Apply the integration testing for any given problem.

CO4: Compare the functional and system testing methods.

CO5: Determine various issues for object oriented testing.

UNIT-I

A PERSPECTIVE ON TESTING:

Basic Definitions, Test Cases, Insights from a Venn Diagram, Identifying Test Cases, Fault Taxonomies, Levels of Testing.

BOUNDARY VALUE TESTING:

Normal Boundary Value Testing, Robust Boundary Value Testing, Worst-Case Boundary Value Testing, Special Value Testing, Examples, Random Testing.

DECISION TABLE-BASED TESTING:

Decision Tables, Decision Table Techniques.

Learning Outcomes: At the end of the module the student will be able to

- 1. Explain Levels of Testing (L2)
- 2. Outline the Boundary Value Testing (L2)
- 3. Explain Decision Table-Based Testing (L2)

UNIT-II

PATH TESTING:

Program Graphs, DD-Paths, Test Coverage Metrics, Basis Path Testing **DATAFLOW TESTING:**

Define/Use Testing, Slice-Based Testing, Program Slicing Tools

Learning Outcomes: At the end of the module the student will be able to

- 1. Classify the path testing and dataflow testing (L2)
- 2. Apply the Program Slicing Tools for the given problem. (L3)
- 3. Illustrate the Test Coverage Metrics (L2)

UNIT-III

INTEGRATION TESTING:

Decomposition-Based Integration, Call Graph-Based Integration, Path-Based Integration, Example: integrationNextDate

Learning Outcomes: At the end of the module the student will be able to

L T P C 3 0 0 3

(10 LECTURES)

(10 LECTURES)

2019

(10 LECTURES)

INFORMATION TECHNOLOGY

- 1. Apply the Decomposition-Based Integration for the given problem(L3)
- 2. Construct the Call Graph-Based Integration for the given problem. (L3)
- 3. Apply the Path-Based Integration for the given problem (L3)

UNIT-IV

SYSTEM TESTING:

Threads, Basic Concepts for Requirements Specification, Model-Based Threads, Use Case–Based Threads, Long versus Short Use Cases, How Many Use Cases? Coverage Metrics for System Testing, Supplemental Approaches to System Testing, Non-functional System Testing, Atomic System Function Testing Example

Learning Outcomes: At the end of the module the student will be able to

- 1. Interpret Atomic System Function Testing with Example.(L5)
- 2. Examine the system testing for the given problem(L4)
- 3. Explain the Model-Based Threads (L2)

UNIT-V

OBJECT-ORIENTED TESTING:

Issues in Testing Object-Oriented Software, Example: ooNextDate, Object-Oriented Unit Testing, Object-Oriented Integration Testing, Object-Oriented System Testing.

Test-Driven Development:

Test-Then-Code Cycles, Automated Test Execution, Java and JUnit Example.

Learning Outcomes: At the end of the module the student will be able to

- 1. Evaluate Object-Oriented System Testing for the given problem. (L5)
- 2. Analyze the Issues in Object-Oriented software testing. (L4)
- 3. Explain the Object-Oriented Integration Testing (L2)

TEXT BOOK:

1. Paul C. Jorgensen, "Software Testing: A Craftsman's Approach", 4th Edition, CRC Press, 2018

REFERENCES:

- 1. Boris Beizer, "Software Testing Techniques", Dreamtech, 2003
- 2. Elfriede Dustin, "*Effective Software testing (50 specific ways to improve Your Testing)*", 1st Edition, Pearson Education, 2006.
- 3. William E.Perry, "*Effective Methods for Software Testing*", 3rd Edition, WILEY, 2011.
- 4. K.V.K.K. Prasad, "Software Testing Tools", 1st Edition, Dream Tech Press, 2005.

WEB REFERENCES:

- 1. https://nptel.ac.in/courses/106/105/106105150/
- 2. https://www.coursera.org/specializations/software-testing-automation
- 3. https://www.coursera.org/learn/introduction-software-testing

(10 LECTURES)

2019

(10 LECTURES)