



# **SNS COLLEGE OF ENGINEERING**

Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade

Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**COURSE NAME : 19CS511 SOFTWARE TESTING**

**III YEAR / V SEMESTER**

**Unit 4- Test Management**

**Topic : Locate Test Items**





# 19CS511 SOFTWARE TESTING



## Syllabus

### UNIT I INTRODUCTION

9

Testing as an Engineering Activity – Testing as a Process – Testing Maturity Model- Testing axioms – Basic definitions – Software Testing Principles – The Tester's Role in a Software Development Organization – Origins of Defects – Cost of defects – Defect Classes – The Defect Repository and Test Design – Defect Examples- Developer/Tester Support of Developing a Defect Repository.

### UNIT II TEST CASE DESIGN STRATEGIES

9

Test case Design Strategies – Using Black Box Approach to Test Case Design – Boundary Value Analysis – Equivalence Class Partitioning – State based testing – Cause-effect graphing – Compatibility testing – user documentation testing – domain testing - Random Testing – Requirements based testing – Using White Box Approach to Test design – Test Adequacy Criteria – static testing vs. structural testing – code functional testing – Coverage and Control Flow Graphs – Covering Code Logic – Paths – code complexity testing – Additional White box testing approaches- Evaluating Test Adequacy Criteria..



# 19CS511 SOFTWARE TESTING



9

## UNIT III LEVELS OF TESTING

The need for Levels of Testing – Unit Test – Unit Test Planning – Designing the Unit Tests – The Test Harness – Running the Unit tests and Recording results – Integration tests – Designing Integration Tests – Integration Test Planning – Scenario testing – Defect bash elimination System Testing – Acceptance testing – Performance testing – Regression Testing – Internationalization testing – Ad-hoc testing – Alpha, Beta Tests – Testing OO systems – Usability and Accessibility testing – Configuration testing – Compatibility testing – Testing the documentation – Website testing

## UNIT IV TEST MANAGEMENT

9

People and organizational issues in testing – Organization structures for testing teams – testing services – Test Planning – Test Plan Components – Test Plan Attachments – Locating Test Items – test management – test process – Reporting Test Results – Introducing the test specialist – Skills needed by a test specialist – Building a Testing Group- The Structure of Testing Group- .The Technical Training Program.



# 19CS511 SOFTWARE TESTING



## UNIT V TEST AUTOMATION

9

Software test automation – skills needed for automation – scope of automation – design and architecture for automation – requirements for a test tool – challenges in automation – Test metrics and measurements – project, progress and productivity metrics.



# Locating Test Items



Locating test items in software testing involves identifying and defining the specific elements within a software application or system that need to be tested. This process is critical for ensuring comprehensive test coverage and effective test execution. Here's a detailed guide on how to locate and define test items.



# Steps to Locate Test Items



## 1. Understand Requirements and Specifications:

**Requirement Analysis:** Review the functional and non-functional requirements documents. This includes business requirements, user stories, and use cases.

**System Design Documentation:** Examine design documents, architecture diagrams, and technical specifications to understand how the system is structured and how different components interact.



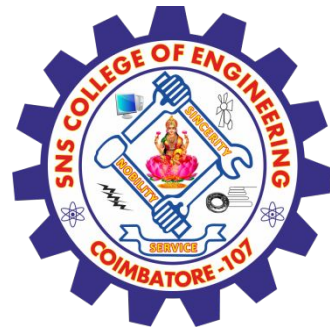
# Steps to Locate Test Items



## 2. Identify Key Features and Functions:

**Feature List:** List all the features and functionalities of the software as described in the requirements or design documents.

**User Stories/Use Cases:** Determine the specific user stories or use cases that describe the interactions users will have with the system.



## Steps to Locate Test Items



### 3. Break Down Features into Testable Components:

**Components and Modules:** Identify individual components or modules within each feature that need testing.

**Sub-Features:** Break down larger features into smaller, testable sub-features or functions.

**Inputs and Outputs:** Define the inputs, processes, and outputs for each component or module





# Steps to Locate Test Items



## 4. Define Test Items:

**Test Cases:** Develop test cases that specify the conditions under which features or components will be tested. Each test case should include a description, steps to execute, expected results, and any necessary preconditions.

**Test Scenarios:** Create test scenarios that outline the conditions and sequences of actions to be tested, often covering end-to-end functionality.

**Test Data:** Identify and prepare test data needed to validate the functionalities of the software. This could include input values, data sets, and expected output results.



## Steps to Locate Test Items



### 5. Prioritize Test Items:

**Risk-Based Prioritization:** Focus on high-risk areas, critical functionalities, and frequently used features.

**Impact Analysis:** Prioritize test items based on recent changes, bug reports, or the impact of new features on existing functionality.

### 6. Traceability:

**Traceability Matrix:** Create a traceability matrix to ensure that all requirements are covered by test cases. This helps in tracking which requirements have been tested and which have not.



# Steps to Locate Test Items



## 7. Incorporate Feedback

**Stakeholder Input:** Gather input from stakeholders, such as developers, product managers, and end-users, to ensure that all relevant test items are identified.

**Continuous Review:** Regularly review and update test items based on changes in requirements, design, or feedback.



# Steps to Locate Test Items



## Examples of Test Items:

1. Functional Test Items
2. Non-Functional Test Items
3. Security Test Items
4. Compatibility Test Items
5. Integration Test Items



# Tools to Assist in Locating Test Items



## 1. Requirements Management Tools:

- i. **JIRA:** Used for tracking requirements, user stories, and their associated test cases.
- ii. **IBM Engineering Requirements Management DOORS:** Helps manage and trace requirements to test cases.

## 2. Test Management Tools:

- i. **TestRail:** For organizing and managing test cases and tracking test coverage.
- ii. **qTest (Tricentis):** Provides features for test case management, execution, and reporting.



# Tools to Assist in Locating Test Items



## 3. Automated Test Tools:

- i. **Selenium:** For automating web application tests based on the identified test cases.
- ii. **Test Complete:** For automating tests across different types of applications.

## 4. Collaboration Tools:

- i. **Confluence:** For documenting requirements, test cases, and test plans in a collaborative environment.



## **TEXT BOOKS:**

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, —Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books, 2011.
2. Ricci, F, Rokach, L. Shapira, B.Kantor, —Recommender Systems Handbook, First Edition, 2011.

## **REFERENCES:**

1. C. Manning, P. Raghavan, and H. Schütze, —Introduction to Information Retrieval, Cambridge University Press, 2008.
2. Stefan Buettcher, Charles L. A. Clarke and Gordon V. Cormack, —Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2010.

# **THANK YOU**