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 1- INTRODUCTION

Topic :The defect repository and test design

IT8076 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..

IT8076 SOFTWARE TESTING

UNIT III LEVELS OF TESTING 9

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.

IT8076 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.

The defect repository and test design

Defect Repository

A defect repository is a systematic collection of information about defects identified during the software testing process. It serves as a central location for tracking, managing, and analyzing defects.

Componenets of Defects Repositary

- **Defect ID:** A unique identifier assigned to each defect for easy reference and tracking.
- **Description:** A detailed explanation of the defect, including steps to reproduce, the observed behavior, and the expected behavior.
- **Severity:** The impact of the defect on the software, which helps prioritize its resolution. Common severity levels include Critical, Major, Minor, and Trivial.
- **Priority:** The urgency of fixing the defect, which may depend on business needs, customer impact, or project deadlines. Priority levels often include High, Medium, and Low.
- **Status:** The current state of the defect in the lifecycle, such as New, Assigned, In Progress, Fixed, Verified, or Closed.
- **Reported By:** The individual or team who discovered and reported the defect.

- **Assigned To:** The developer or team responsible for fixing the defect.
- **Date Reported:** The date when the defect was initially reported.
- **Date Fixed:** The date when the defect was resolved or fixed.
- **Comments:** Additional notes or updates related to the defect, including discussions between team members.
- **Attachments:** Any related files, screenshots, logs, or other documentation that can help in understanding or reproducing the defect.

Uses of a Defect Repository

- **Tracking and Management:** Provides a centralized system for tracking the status and history of defects, ensuring that nothing is overlooked.
- **Communication:** Facilitates communication between testers, developers, and other stakeholders by providing a clear record of defects and their resolution status.
- **Reporting and Analysis:** Allows for generating reports and analyzing defect trends, which can be used to improve quality processes and identify recurring issues.
- **Historical Reference:** Serves as a historical record of defects, which can be useful for future reference, understanding long-term issues, or auditing purposes

COMPONENTS OF TEST DESIGNS

1.Test Plan: A high-level document outlining the testing strategy, objectives, resources, schedule, and scope. It includes:

- **Test Objectives:** What the testing aims to achieve.
- **Scope:** What will be tested and what is out of scope.
- **Resources:** Testing tools, environments, and personnel.
- **Schedule:** Timeline for testing activities.
- **Risk Management:** Identified risks and mitigation strategies.

2.Test Case: A detailed description of a specific test scenario, including:

- **Test Case ID:** A unique identifier for the test case.
- **Test Case Description:** A summary of what the test case will validate.
- **Preconditions:** Any setup required before executing the test case.
- **Test Steps:** Sequential actions to execute the test case.
- **Expected Results:** The anticipated outcome of each step or the overall test case.
- **Postconditions:** The state of the system after the test case execution.

3.Test Script: Automated code that performs the actions of a test case. Test scripts are often written in scripting languages or using test automation tools.

4.Test Data: Specific data used during testing to validate functionality, including:

- **Input Data:** Data entered into the system.
- **Expected Output:** Results anticipated from the test.
- **Boundary Conditions:** Data at the edge of acceptable input ranges.

5.Test Environment: The setup in which testing will be performed, including hardware, software, network configurations, and any other required tools.

6.Test Coverage: Ensures that all aspects of the application are tested, including:

- **Functional Coverage:** All functional requirements are tested.
- **Non-Functional Coverage:** Performance, security, and other non-functional requirements are tested.

TYPES OF TESTING

- **Unit Testing:** Testing individual components or units of code to ensure they function correctly.
- **Integration Testing:** Testing the interaction between integrated components or systems to ensure they work together as expected.
- **System Testing:** Testing the complete and integrated software system to verify that it meets the specified requirements.
- **Acceptance Testing:** Validates that the software meets business requirements and is ready for delivery to the end-users. This includes User Acceptance Testing (UAT).
- **Regression Testing:** Ensures that new changes or fixes do not adversely affect existing functionality.
- **Performance Testing:** Assesses the software's performance under various conditions, including load and stress testing.

BEST PRACTICES FOR TEST DESIGNS

- **Define Clear Objectives:** Ensure that test cases are aligned with the test plan objectives and requirements.
- **Use Test Design Techniques:** Apply techniques such as boundary value analysis, equivalence partitioning, and decision table testing to create effective test cases.
- **Maintain Traceability:** Ensure traceability between requirements and test cases to confirm all requirements are covered.
- **Review and Revise:** Regularly review test cases for accuracy and completeness, and revise them as needed based on changes in requirements or defects.
- **Automate Where Possible:** Use test automation to increase efficiency and coverage, especially for repetitive and regression tests.

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