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 5- INTRODUCTION

Topic : REQUIREMENTS FOR A TEST TOOL

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.

IT8076 SOFTWARE TESTING

TEXT BOOKS:

1.Srinivasan Desikan and Gopalaswamy Ramesh, ―Software Testing – Principles and Practices‖, Pearson Education, 2006.

2. Ron Patton, ―Software Testing‖, Second Edition, Sams Publishing, Pearson Education, 2007. AU Library.com

REFERENCES:

1.Ilene Burnstein, ―Practical Software Testing‖, Springer International Edition, 2003.

2. Edward Kit,‖ Software Testing in the Real World – Improving the Process‖, Pearson Education, 1995.

3. Boris Beizer,‖ Software Testing Techniques‖ – 2nd Edition, Van Nostrand Reinhold, New York, 1990.

4. Aditya P. Mathur, ―Foundations of Software Testing _ Fundamental Algorithms and Techniques‖, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008.

REQUIREMENT FOR A TEST

FUNCTONALITY

- **Test Case Management:** Ability to create, manage, and execute test cases. Should support different test case types, such as functional, non-functional, regression, and user acceptance testing.
- **Automation Support**: For tools that need to support automated testing, they should integrate with scripting languages or frameworks (e.g., Selenium, JUnit, TestNG).
- **Test Data Management**: Ability to handle and manage test data, including generation, storage, and anonymization, if needed.
- **Test Execution**: Features for running tests, either manually or automatically, and handling different execution environments

USABILITY

- **User Interface**: Intuitive and user-friendly interface that allows testers to efficiently create, modify, and run tests.
- **Configuration**: Easy setup and configuration of the tool to fit different testing environments and requirements.
- **Reporting and Dashboards**: Clear and customizable reports and dashboards to provide insights into test results and metrics.
- **Feedback and Improvement**: Implement mechanisms for users to provide feedback on the tool's usability, report bugs, and suggest improvements. This could include feedback forms, surveys, or in-tool feedback options.

INTEGRATION

- **CI/CD Integration**: Seamless integration with Continuous Integration/Continuous Deployment (CI/CD) pipelines and tools (e.g., Jenkins, GitLab CI).
- **Version Control:** Compatibility with version control systems (e.g., Git, SVN) to manage test scripts and related assets.
- **Issue Tracking:** Integration with issue tracking systems (e.g., JIRA, Bugzilla) to link test results with defect reports.
- **Collaboration and Communication Tools :**Capability to send alerts and notifications to team members based on test execution outcomes or critical issues.

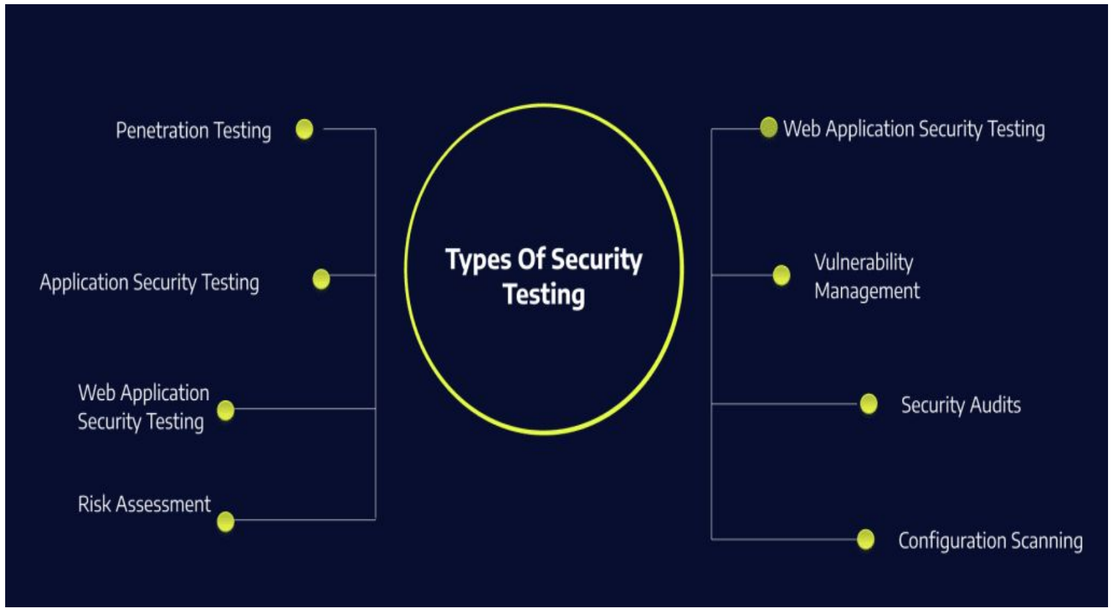
FLEXIBILITY AND EXTENSIBILITY

- **Customizability:** Ability to customize test scenarios, reports, and workflows to fit specific needs.
- **Plugin Support:** Support for plugins or extensions to add new functionalities or integrate with additional tools.
- **Scripting Support:** Support for custom scripts or code to handle specialized testing needs.
- **Adaptability:** Provide flexibility in generating and managing test data, including data-driven testing and support for different data sources (e.g., databases, APIs).

SECURITY

•**Data Protection**: Mechanisms to ensure test data and results are secure, including encryption and access controls.

•**Authentication and Authorization**: Secure login mechanisms and role-based access controls to manage who can access and modify test configurations and results.



COST AND LICENSING

- **Cost-Effectiveness**: Consideration of the cost in relation to the features and benefits provided. This includes initial purchase, subscription fees, and potential costs for scaling or additional features.

- **Licensing Model**: Clear understanding of the licensing terms and conditions, including any restrictions on usage or distribution.

- **Evaluating Cost and Licensing:**Evaluate different tools and their licensing models to find the best fit for your budget and requirements. Consider both the initial costs and the long-term implications.

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.


SNS COLLEGE OF ENGINEERING
COIMBATORE -107


sns
INSTITUTIONS

THANK YOU