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-TEST AUTOMATION

Topic 1 : Progress And Productivity Metrics

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

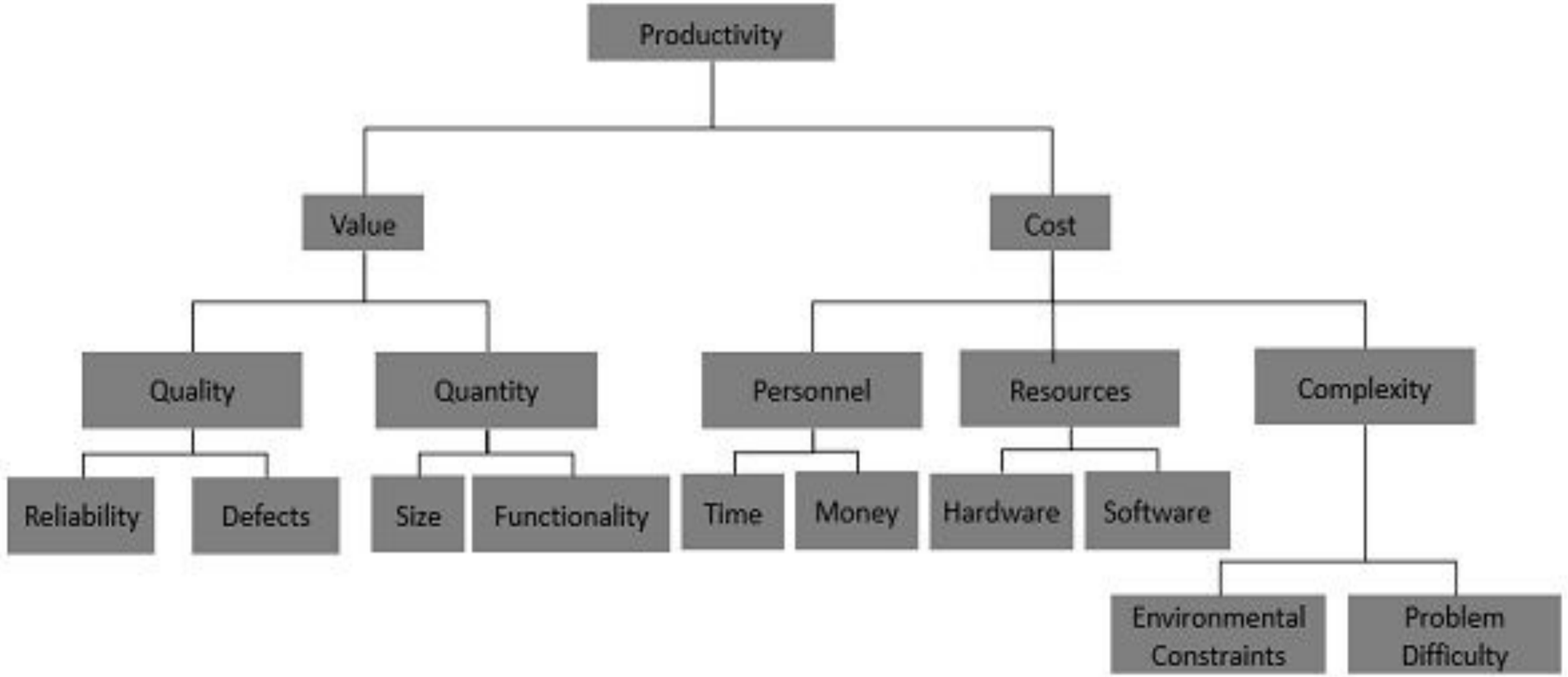
REFERENCES:

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.


Productivity
Value
Cost
Quality
Quantity
Personnel
Resources
Complexity
Reliability
Defects
Size
Functionality
Time
Money
Hardware
Software
Environmental Constraints
Problem Difficulty

Defect Metrics

- **Defect Density**: Measures the number of defects per unit of code (e.g., per thousand lines of code). It helps assess code quality and the effectiveness of testing.

- Formula: Defect Density = (Number of Defects / Size of Code) × 1,000

- **Defect Discovery Rate**: The rate at which defects are identified over time. This helps gauge testing progress.

- Formula: Defect Discovery Rate = Number of Defects Found / Time Period

- **Defect Severity**: Classification of defects by their impact (e.g., critical, major, minor). This prioritizes defect resolution effort.

Test Coverage Metrics

- **Code Coverage**: Measures the percentage of code executed by tests. Higher coverage indicates more thorough testing.
- Types: Line Coverage, Branch Coverage, Path Coverage.
- **Requirement Coverage**: Measures the percentage of requirements covered by test cases
- Formula: Requirement Coverage = (Number of Requirements Covered / Total Number of Requirements) × 100%
- **Line Coverage**: Measures the percentage of code lines executed by tests.
- **Formula**: Line Coverage = (Number of Executed Lines / Total Number of Lines) × 100%
- **Example**: If a test suite executes 80 out of 100 lines of code, the line coverage is 80%

Test Execution Metrics

- **Test Case Execution Rate**: The number of test cases executed over a specific period.

- Formula: Test Case Execution Rate = Number of Test Cases Executed / Time Period

- **Test Pass Rate**: The percentage of executed test cases that pass

.

- Formula: Test Pass Rate = (Number of Test Cases Passed / Number of Test Cases Executed) × 100%

- **Test Failure Rate**: The percentage of executed test cases that fail.

- Formula: Test Failure Rate = (Number of Test Cases Failed / Number of Test Cases Executed) × 100

Productivity Metrics

- **Defects Found per Tester**: Measures the productivity of individual testers.
- Formula: Defects Found per Tester = Number of Defects Found / Number of Testers
- **Test Cases Executed per Tester**: The average number of test cases executed by each tester.
- Formula: Test Cases Executed per Tester = Number of Test Cases Executed / Number of Testers
- **Test Automation Coverage**: Measures the percentage of test cases automated compared to total test cases.

Efficiency Metrics

- **Test Execution Time**: The total time taken to execute the test cases. Shorter execution time with effective results indicates efficiency.
- **Test Cycle Time**: The time taken to complete a full cycle of testing, from planning to reporting.
- Formula: Test Cycle Time = End Time - Start Time of Test Cycle
- **Defect Resolution Time**: The average time taken to resolve defects after they are reported.
- Formula: Defect Resolution Time = (Date Defect Closed - Date Defect Reported) / Number of Defects Closed.

Test Planning and Execution Metrics

- **Test Plan Adherence**: Measures how well the executed tests align with the test plan.

- Formula: Test Plan Adherence = (Number of Test Cases Executed from Plan / Total Number of Test Cases in Plan) × 100%

- **Test Case Preparation Time**: Time taken to develop test cases.

- Formula: Test Case Preparation Time = (End Time of Test Case

- Preparation - Start Time of Test Case Preparation) / Number of Test Cases Prepared

Advantages

- Improved Software Quality
- Better Decision-Making
- Increased Efficiency
- Enhanced Accountability
- Better Communication

Disadvantages

- Misleading Information
- Overhead and Complexity
- Potential for Gaming the System
- Lack of Context
- False Sense of Security

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