SNS COLLEGE OF TECHNOLOGY

Kurumbapalayam (Po), Coimbatore – 641 035

An Autonomous Institution

Accredited by NAAC – UGC with 'A++' Grade

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

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

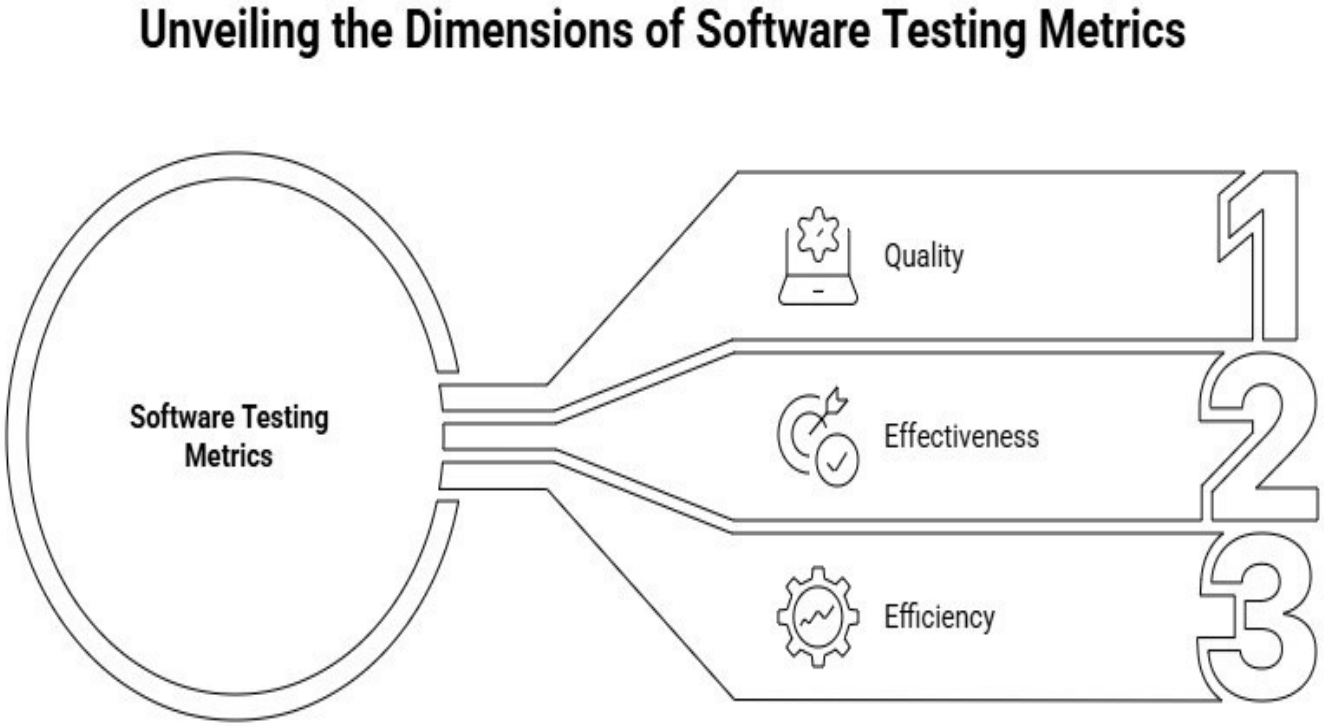
COURSE NAME: 23ITO201- Software Testing (OPEN ELECTIVE)

III YEAR / VI SEMESTER

Unit 5 - TEST AUTOMATION

Topic : Test Metrics and Measurements

Empathy

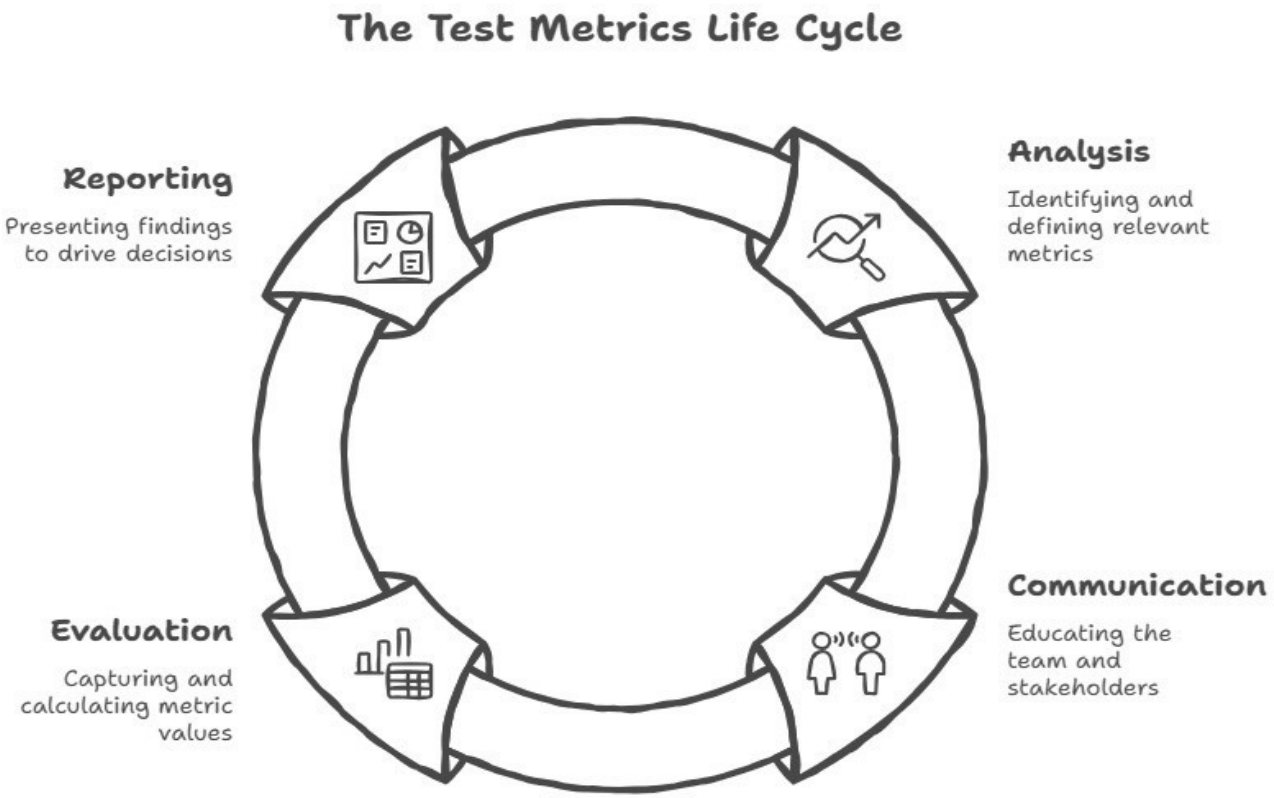

Unveiling the Dimensions of Software Testing Metrics
Software Testing Metrics
Quality
1
Effectiveness
2
Efficiency
3

Software Testing Metrics

Critical Metrics and Formulas

The following table highlights key calculated metrics used to gauge testing success:

Metric	Formula	Purpose
Test Case Effectiveness	$\frac{\text{Total Defects Detected}}{\text{Total Test Cases Run}} \times 100$	Measures the ability of test cases to find bugs.
Defect Density	$\frac{\text{Total Defects Found}}{\text{Size of Module (e.g., KLOC)}}$	Shows the "concentration" of bugs in specific code areas.
Defect Leakage	$\frac{\text{Defects Found in Production}}{\text{Total Defects Found}} \times 100$	Measures testing gaps that allowed bugs to escape to users.
Test Coverage	$\frac{\text{Requirements Covered by Tests}}{\text{Total Requirements}} \times 100$	Ensures all critical functionalities are being tested.
Pass/Fail Percentage	$\frac{\text{Passed Test Cases}}{\text{Total Executed Test Cases}} \times 100$	Provides a high-level view of product stability.


The Test Metrics Life Cycle
Reporting
Presenting findings to drive decisions
Analysis
Identifying and defining relevant metrics
Evaluation
Capturing and calculating metric values
Communication
Educating the team and stakeholders

Measurement Best Practices

Activity

Activity: Identify Requirements for a Test Tool

🧩 **Activity Title**

"Build Your Own Test Tool"

👥 **Group Formation**

•Divide students into **groups of 3–5 members**

📝 **Task Description**

Each group will **design a simple test tool** and identify its requirements.

🔧 **Step-by-Step Activity**

Step 1: Choose a Scenario

Pick one system to test:

•Web application (e.g., login page)

•Mobile app

•API service

Step 2: Identify Requirements

Each group must list:

✅ **Functional Requirements**

(What the tool should do)

•Test case creation

•Test execution

•Result validation

•Bug tracking

✅ **Non-Functional Requirements**

(How the tool should behave)

•**Usability** → Easy interface

•**Performance** → Fast execution

•**Security** → Data protection

•**Compatibility** → Works on multiple platforms

•**Scalability** → Handles more test cases

•**Reliability** → Accurate results

•**Maintainability** → Easy updates

•**Integration** → Works with tools like CI/CD

•**Reporting** → Generates reports

💡 **Example Outcome**

A group designing a **web testing tool** may say:

•Functional → Run test scripts, generate reports

•Non-functional → Fast, secure, easy to use

7

MCQs: Requirements for a Test Tool

1. Which of the following is a primary function of a test tool?

A) Cooking automation

B) Test execution

C) Graphic design

D) Video editing

Answer: B) Test execution

2. Test case creation and management belong to:

A) Non-functional requirement

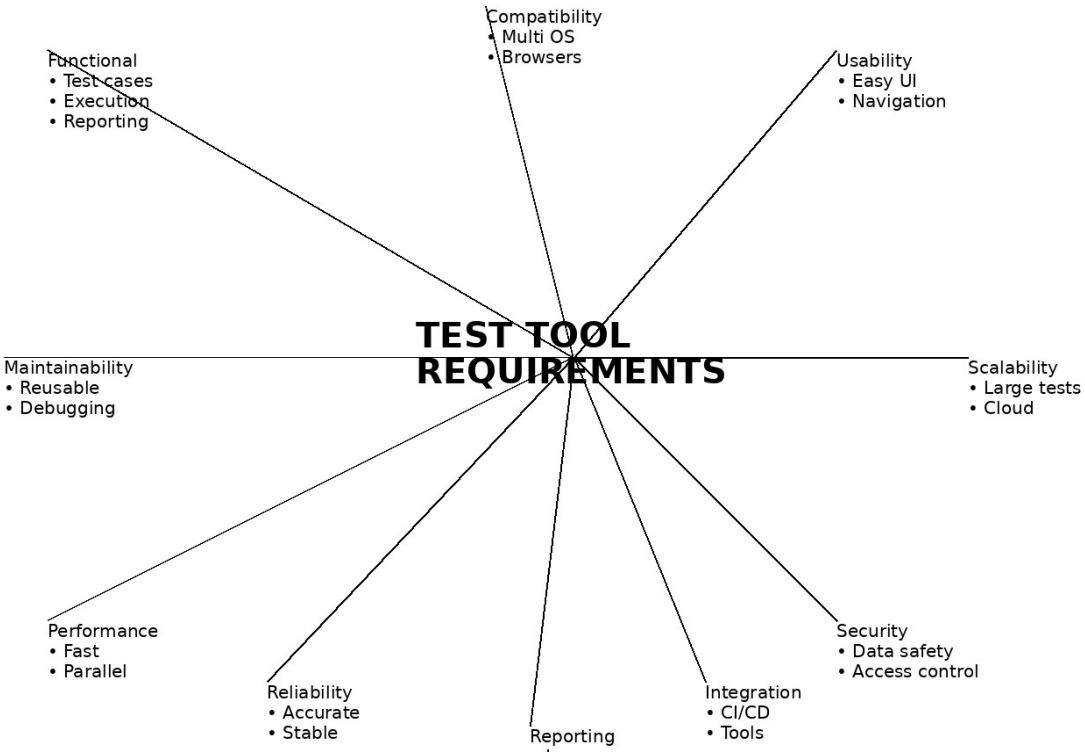
B) Functional requirement

C) Security requirement

D) Performance requirement

Answer: B) Functional requirement

MINDMAP


TEST TOOL REQUIREMENTS
Functional
• Test cases
• Execution
• Reporting
Compatibility
• Multi OS
• Browsers
Usability
• Easy UI
• Navigation
Maintainability
• Reusable
• Debugging
Scalability
• Large tests
• Cloud
Performance
• Fast
• Parallel
Reliability
• Accurate
• Stable
Reporting
Integration
• CI/CD
• Tools
Security
• Data safety
• Access control

TEXT BOOKS	
1.	Srinivasan Desikan and Gopalaswamy Ramesh, ―Software Testing – Principles and Practices, Pearson Education, 2006.
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.
5.	https://www.geeksforgeeks.org/software-testing/automation-testing-software-testing/

THANK YOU!

11