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

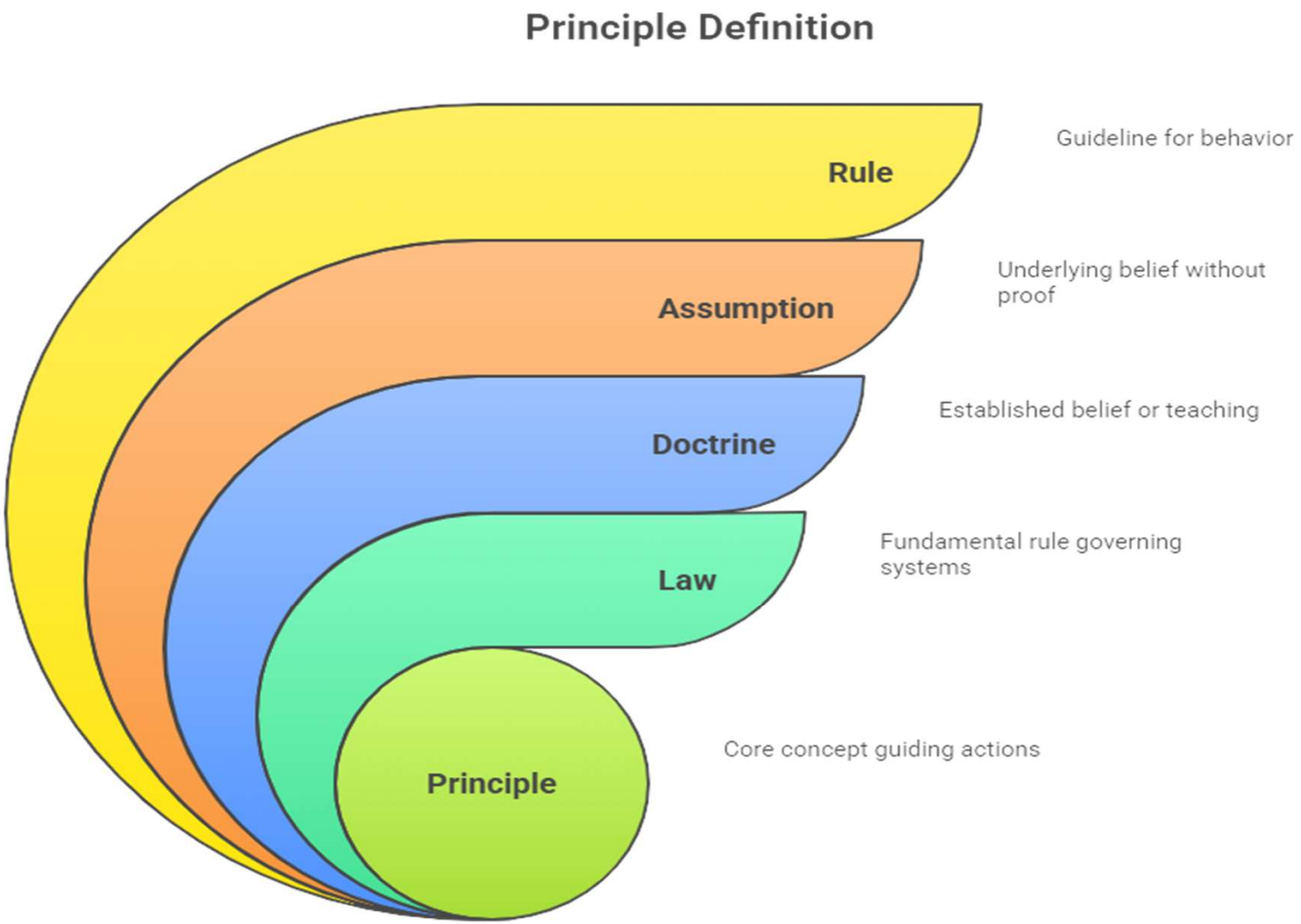
COURSE NAME: 23IT0201 SOFTWARE TESTING (OPEN ELECTIVE)

III YEAR / VI SEMESTER

Unit 1 - INTRODUCTION

Topic 5 : Software Testing Principles

Recall In Principle Definition



Empathy in Software Testing Principles

Context-Dependent Testing

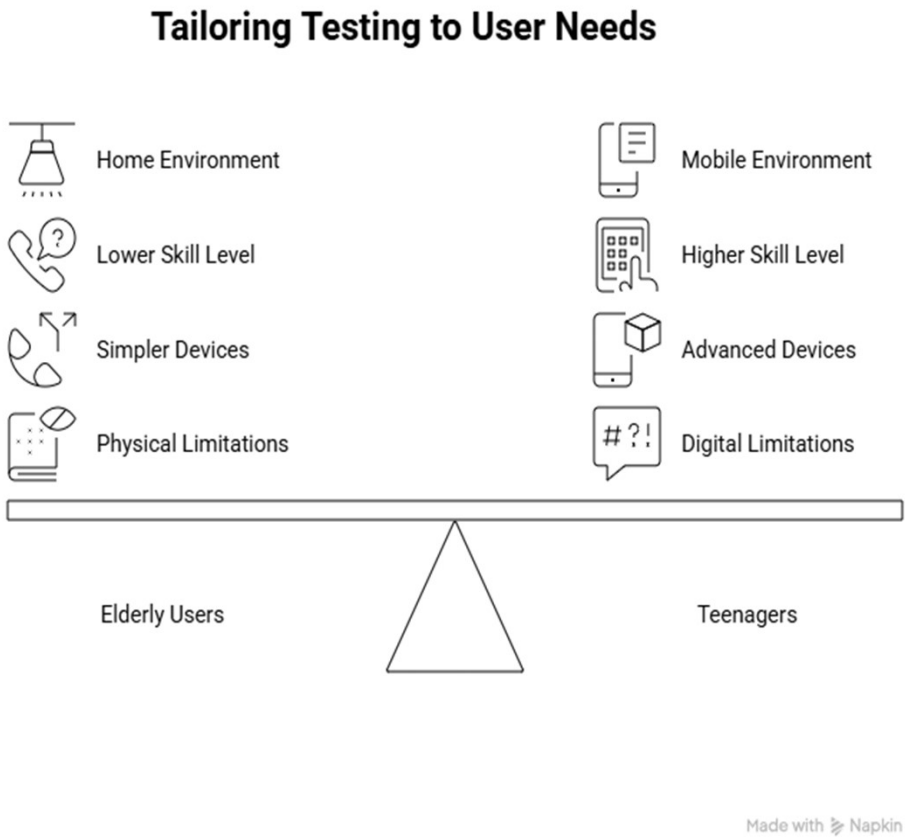
✓ Testing depends on who the users are and where they use the system

Empathy Role:

•Consider user environment, age, skill level, devices, and limitations

🧠 Example:

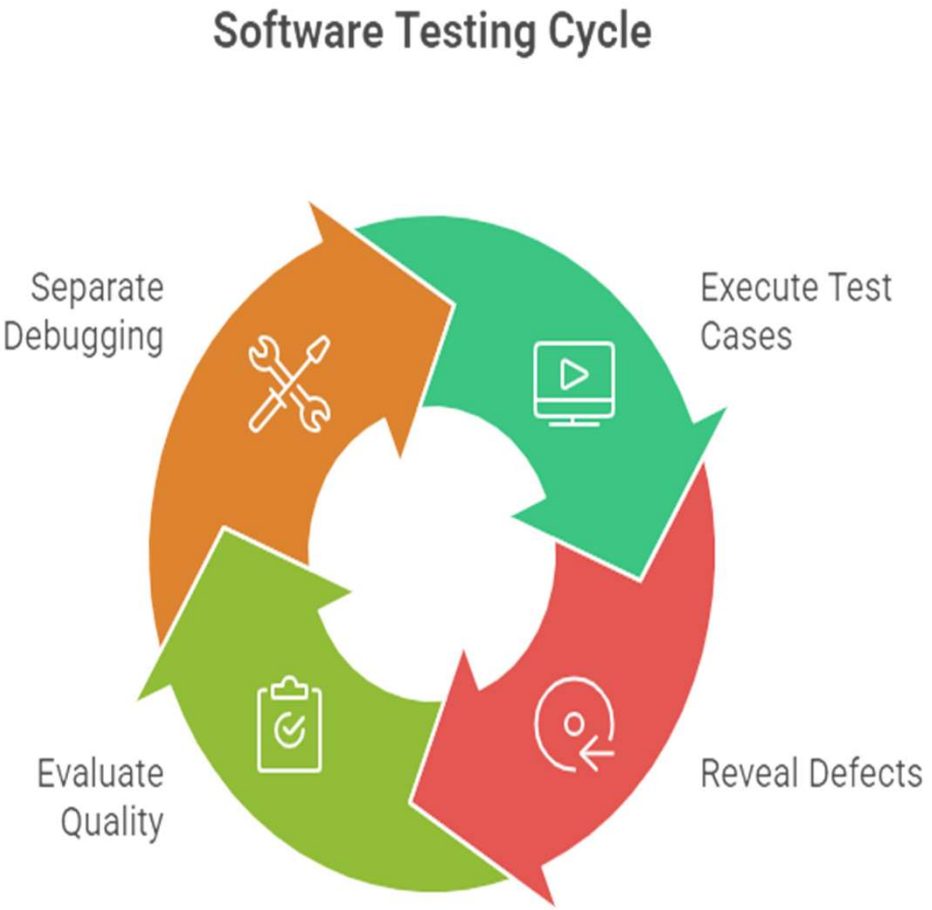
Testing a banking app for elderly users vs. teenagers.



Software Testing Principles

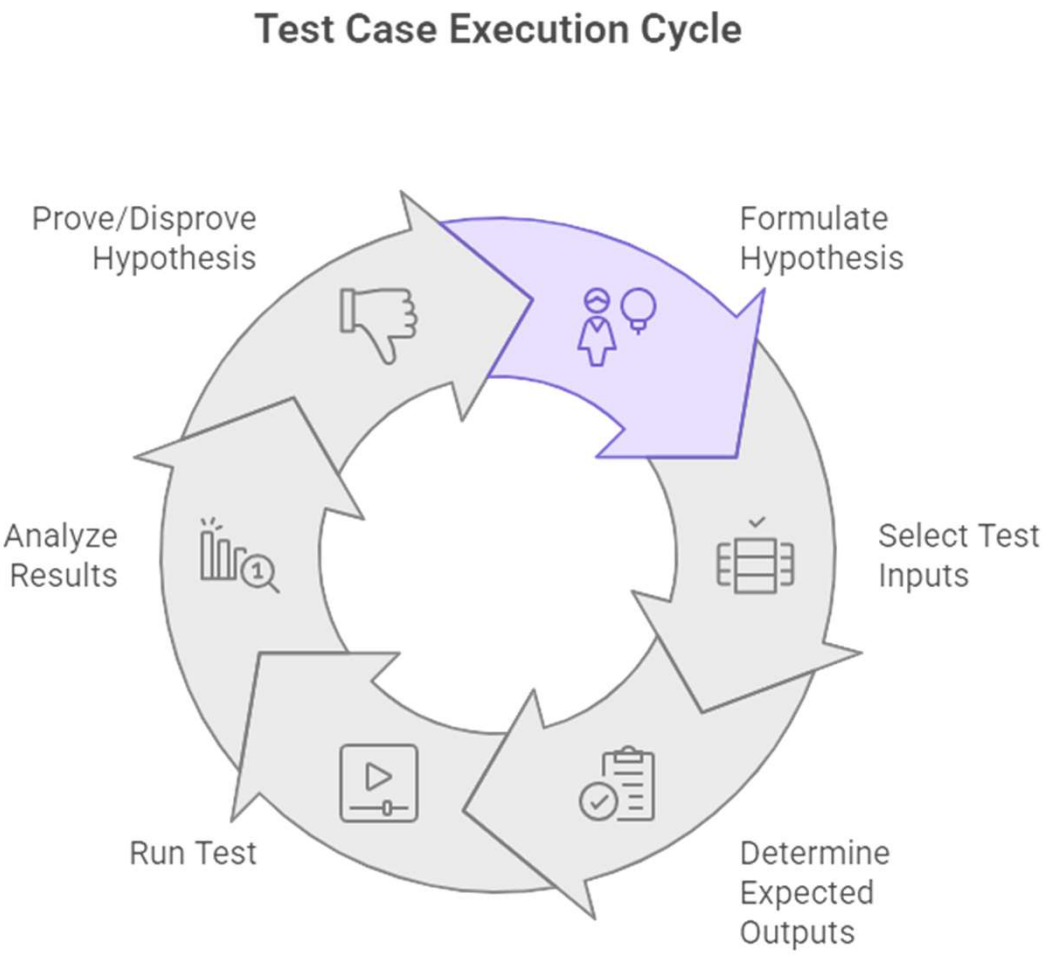
Principle 1/11

Testing is the process of exercising a software component using a selected set of test cases, with the intent of (i) revealing defects, and (ii) evaluating quality



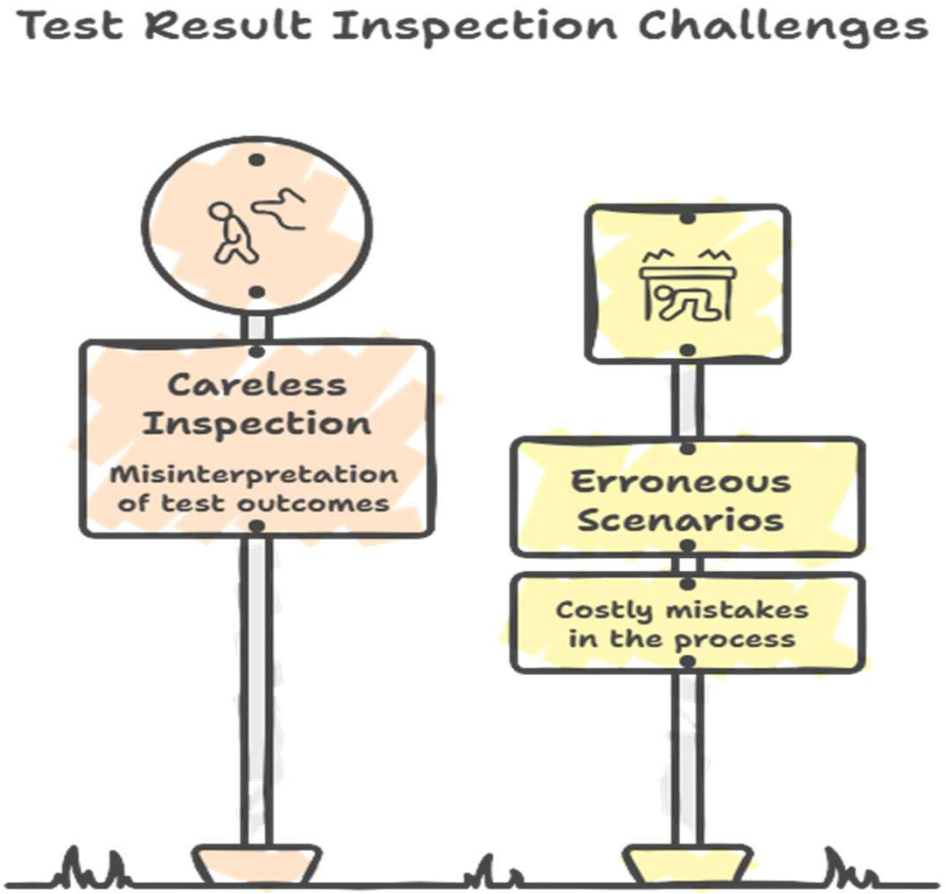
Principle 2/11

When the test objective is to detect defects, then a good test case is one that has a high probability of revealing a yet undetected defect(s).



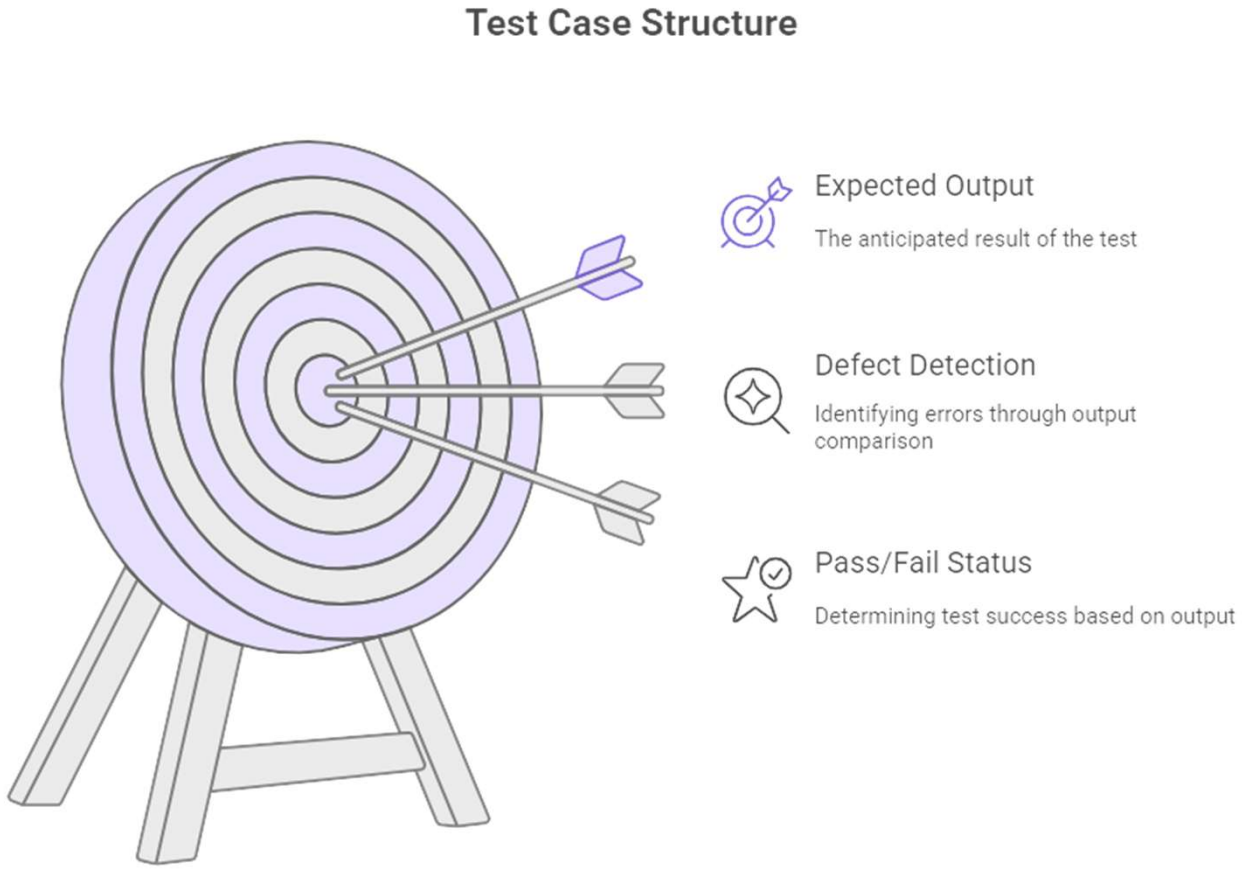
Principle 3/11

Test results should be inspected meticulously



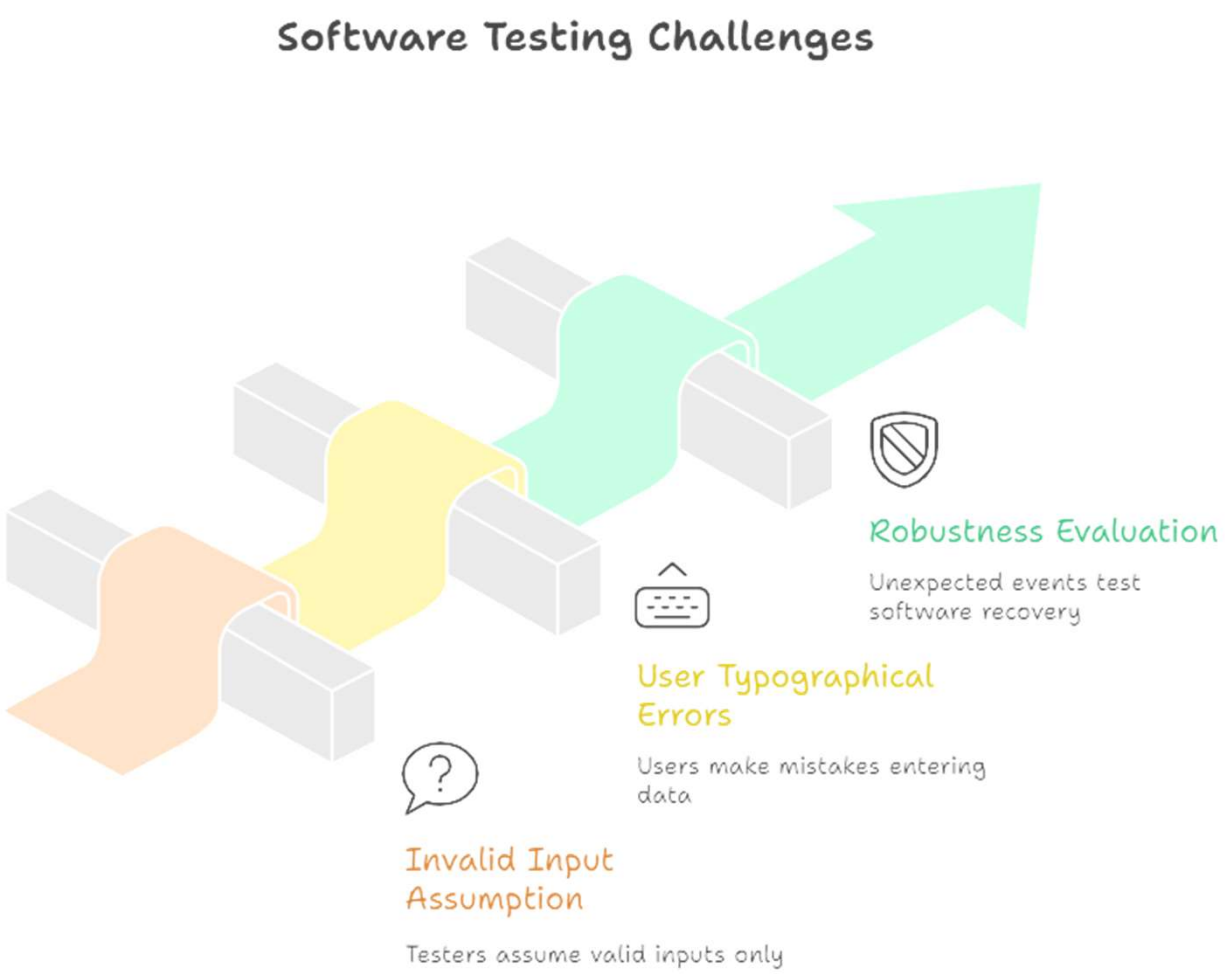
Principle 4/11

A test case must contain the expected output or result



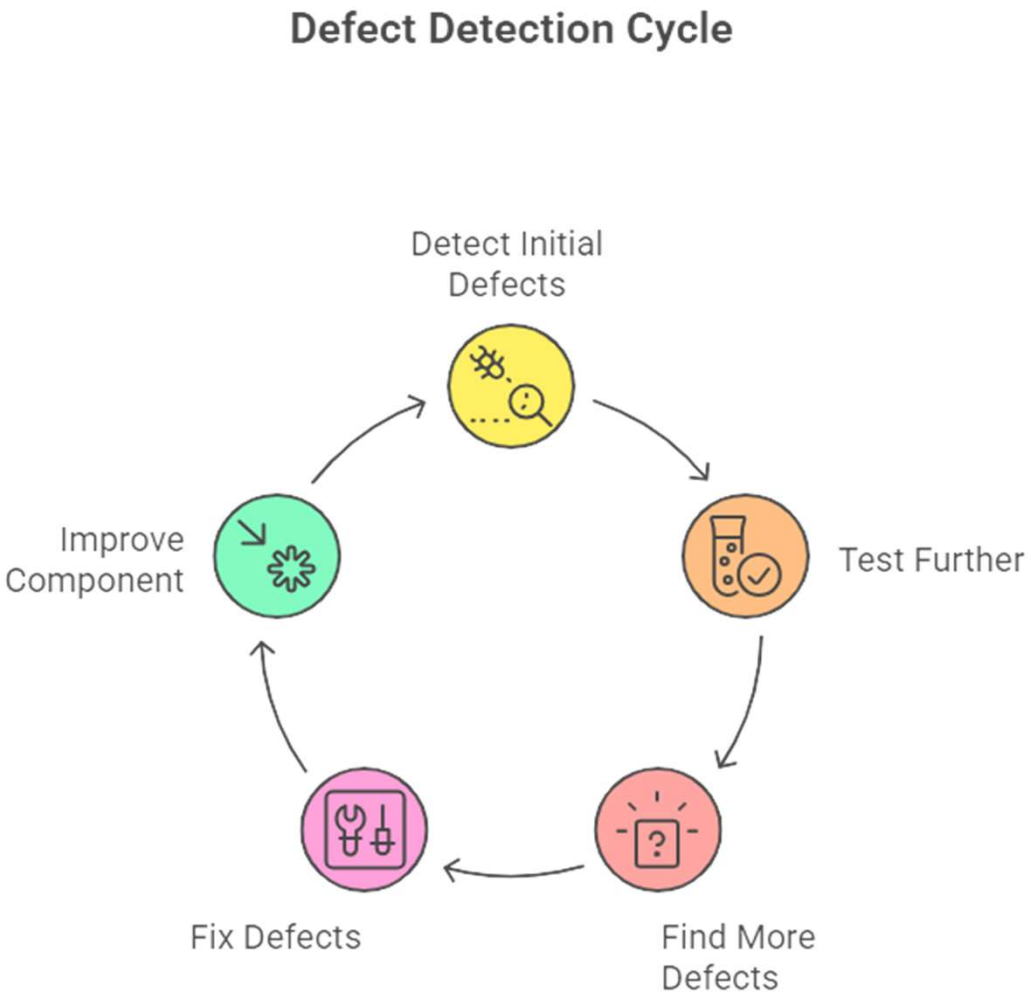
Principle 5/11

Test cases should be developed for both valid and invalid input conditions.



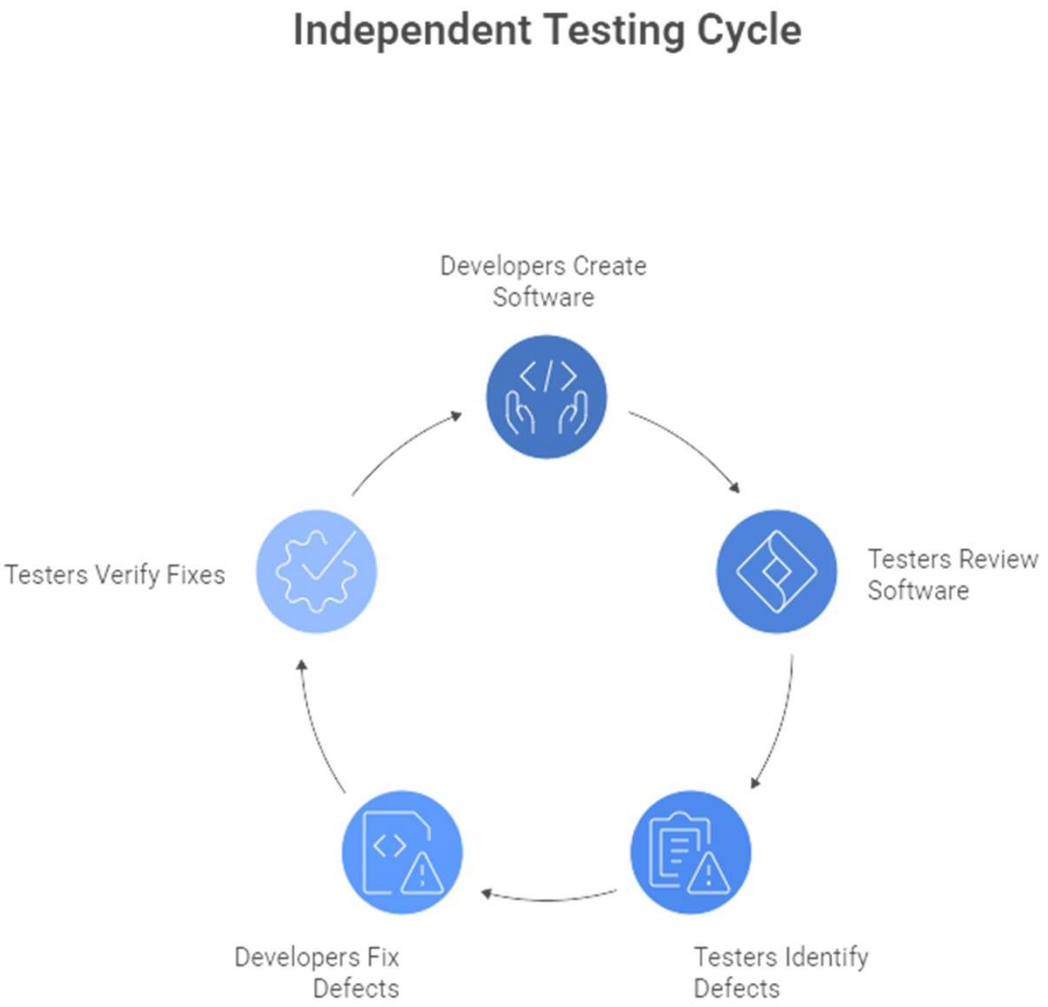
Principle 6/11

The probability of the existence of additional defects in a software component is proportional to the number of defects already detected in that component



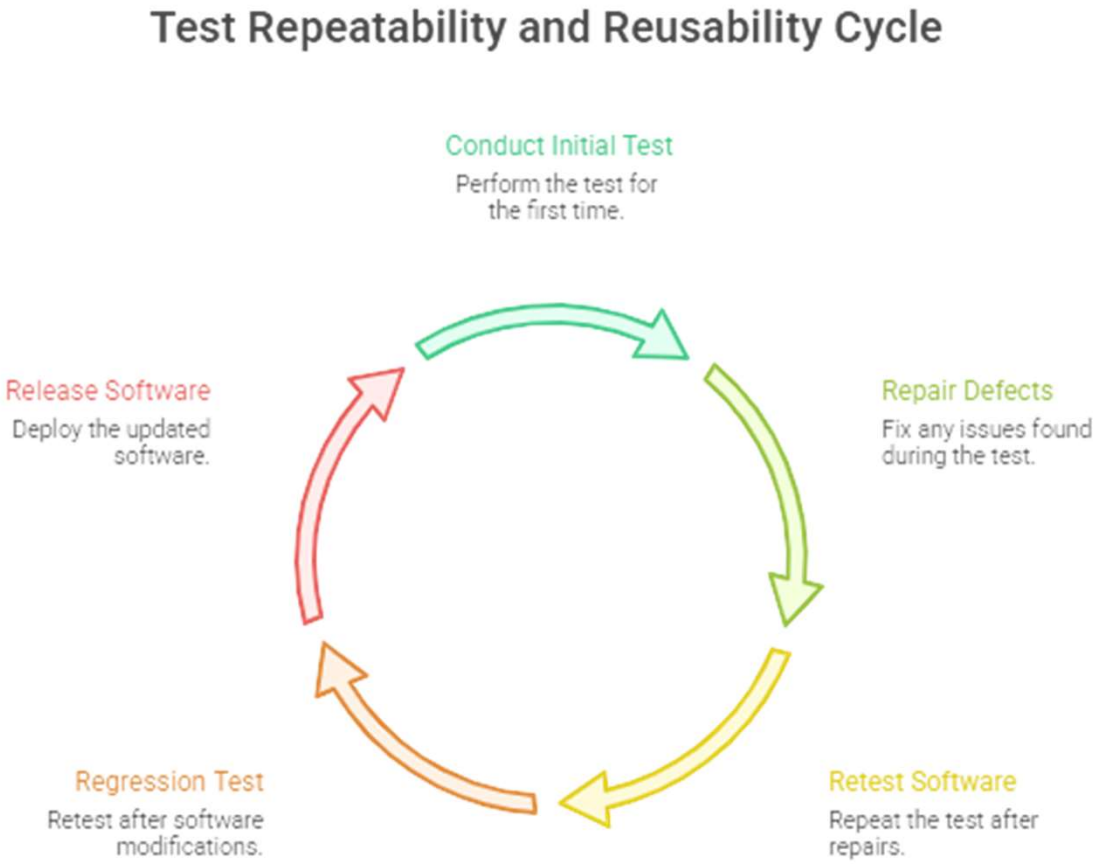
Principle 7/11

Testing should be carried out by a group that is independent of the Development group



Principle 8/11

Tests must be repeatable and reusable.

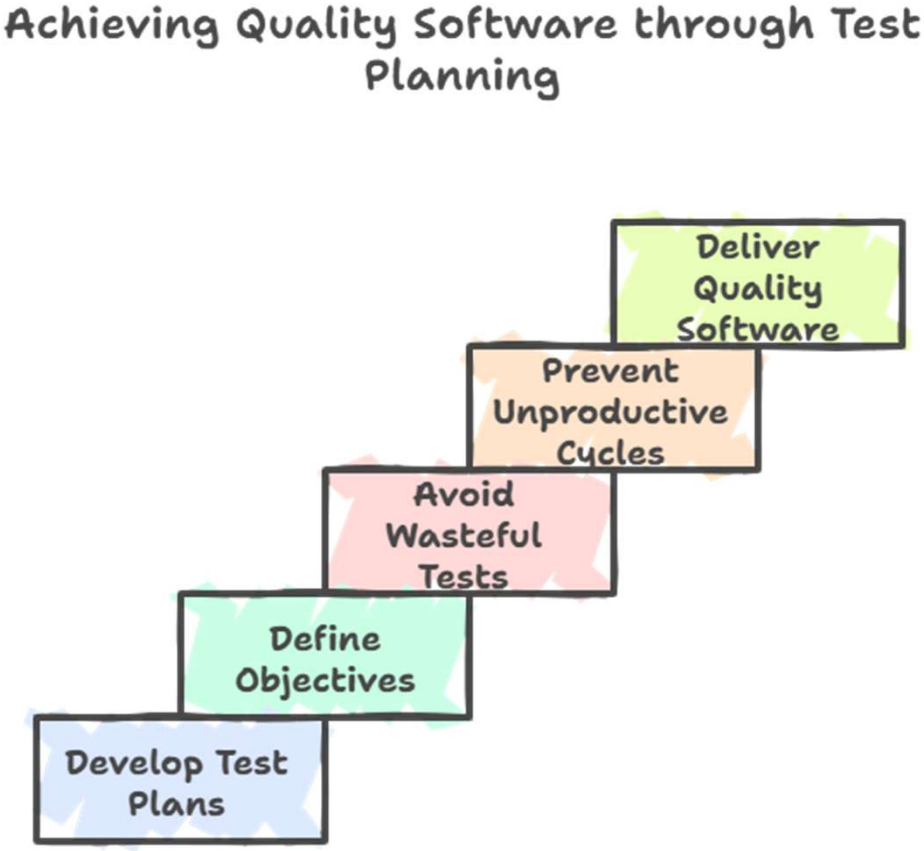


Activity


Software Testing Principles – Activities
Finding Bugs
Testing Shows Presence of Defects
Test a buggy app to find defects.
Combinations
Exhaustive Testing is Impossible
Login
Count all possible inputs for a login page.
Early Testing
Early Testing
Check requirements before coding.
Module Defects
Defect Clustering
High Defects
Find the module with most defects.
Repeticle Paradox
Pesticide Paradox
See why same tests stop finding new bugs.
Context-Dependent Testing
Context-Dependent Testing
VS
Compare tests for a bank vs. game app.
User Feedback
Absence-of-Errors Fallacy
Test a user-friendly but flawed system.
Risk Assessment
Risk-Based Testing
HIGH
MEDIUM
LOW
Focus tests on high-risk areas.
Team Effort
Testing is Everyone's Responsibility
Work together to find defects.

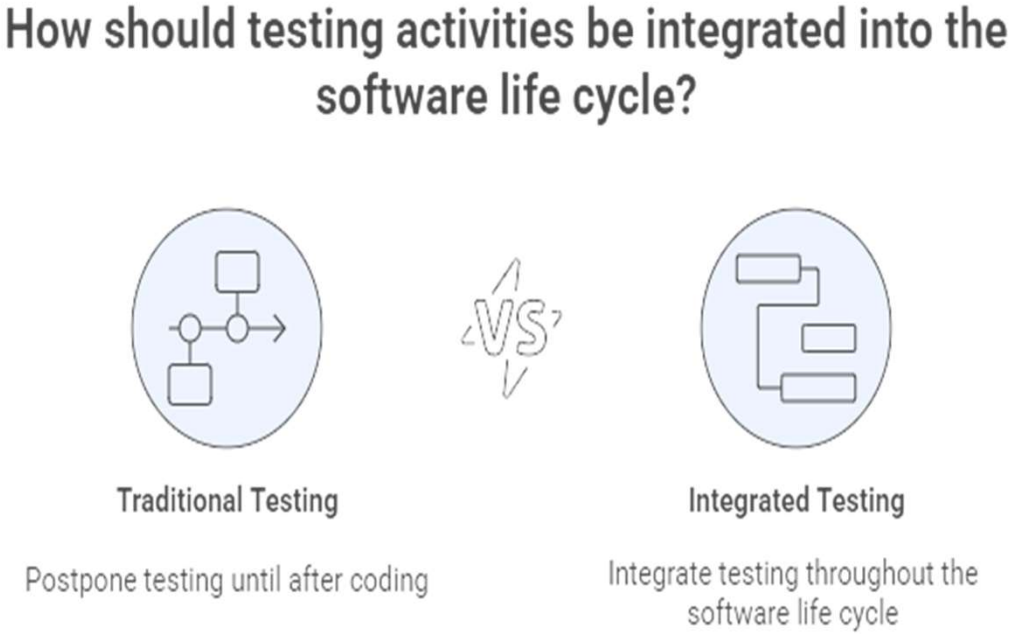
Principle 9/11

Testing should be planned.



Principle 10/11

Testing activities should be integrated into the software life cycle.

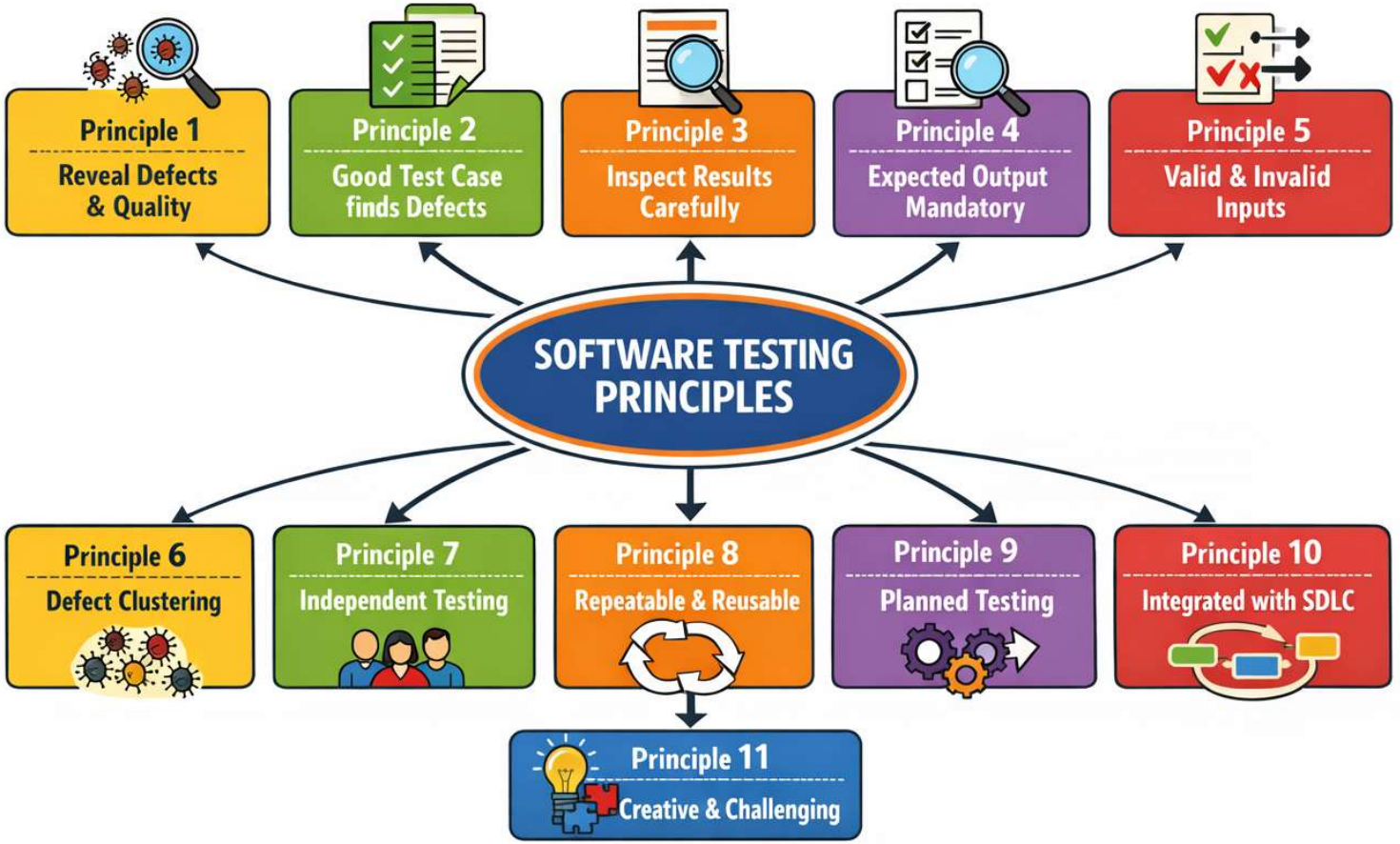


Principle 11/11

Testing is a creative and challenging task.

- Creative
- Face difficulties

Mind Map


SOFTWARE TESTING PRINCIPLES
Principle 1
Reveal Defects & Quality
Principle 2
Good Test Case finds Defects
Principle 3
Inspect Results Carefully
Principle 4
Expected Output Mandatory
Principle 5
Valid & Invalid Inputs
Principle 6
Defect Clustering
Principle 7
Independent Testing
Principle 8
Repeatable & Reusable
Principle 9
Planned Testing
Principle 10
Integrated with SDLC
Principle 11
Creative & Challenging

Software Testing Principles – MCQs

1. Which principle states that testing can show the presence of defects but not their absence?

A. Exhaustive testing
B. Defect clustering
C. Absence-of-errors fallacy
D. Testing shows presence of defects

✅ **Answer:** D

2. Exhaustive testing is:

A. Possible with automation
B. Possible for small systems
C. Not possible
D. Mandatory for safety systems

✅ **Answer:** C

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.

THANK YOU