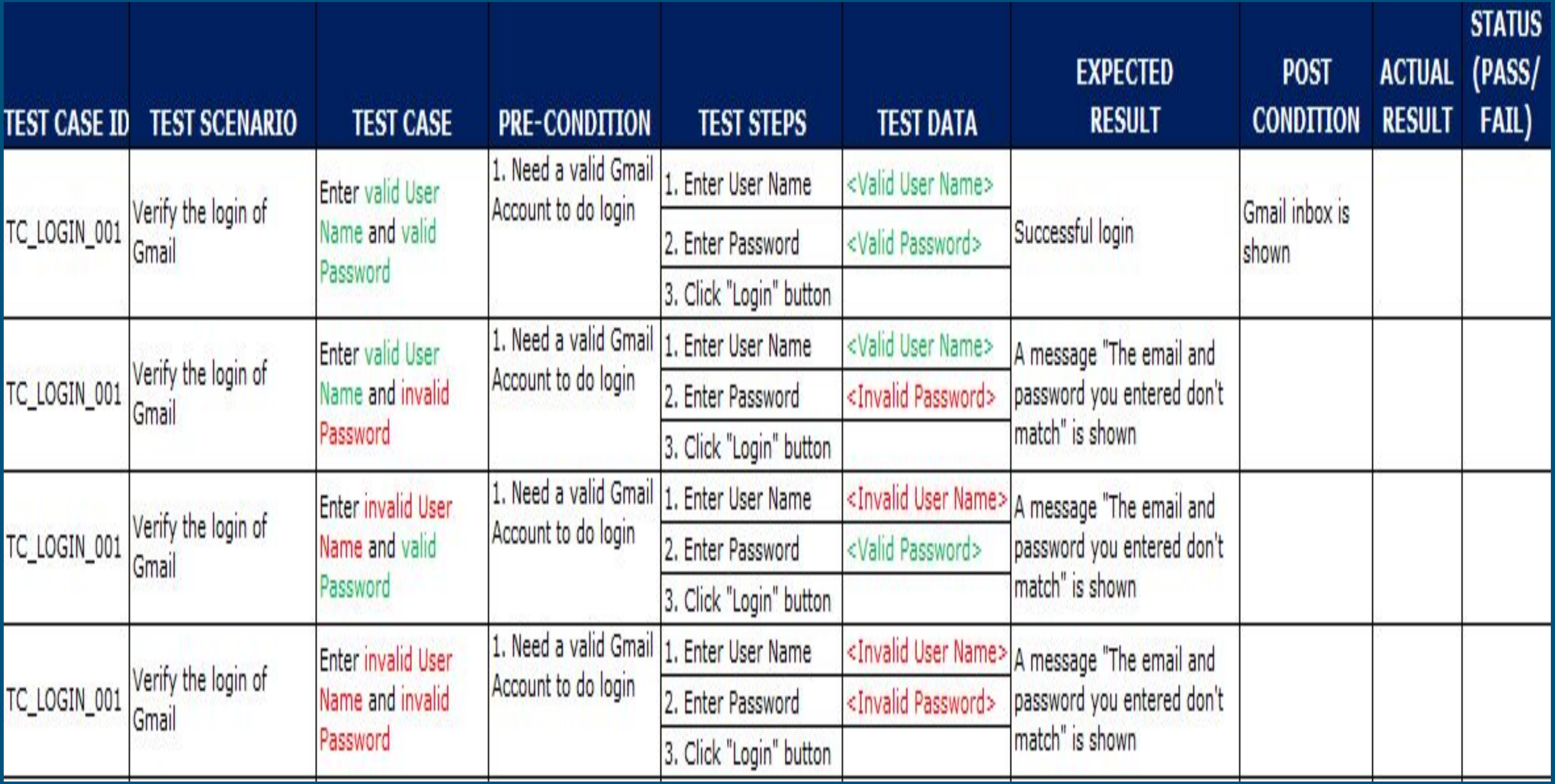
Testing

2022 March/April
SWE
Vinod Pathari

Most Important

1. Need specification to know what to test
2. Need expected output to know whether a particular test has ‘passed’

Gmail Login



TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/ FAIL)
TC_LOGIN_001	Verify the login of Gmail	Enter valid User Name and valid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Valid User Name>	Successful login	Gmail inbox is shown		
				2. Enter Password	<Valid Password>				
				3. Click "Login" button					
TC_LOGIN_001	Verify the login of Gmail	Enter valid User Name and invalid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Valid User Name>	A message "The email and password you entered don't match" is shown			
				2. Enter Password	<Invalid Password>				
				3. Click "Login" button					
TC_LOGIN_001	Verify the login of Gmail	Enter invalid User Name and valid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Invalid User Name>	A message "The email and password you entered don't match" is shown			
				2. Enter Password	<Valid Password>				
				3. Click "Login" button					
TC_LOGIN_001	Verify the login of Gmail	Enter invalid User Name and invalid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Invalid User Name>	A message "The email and password you entered don't match" is shown			
				2. Enter Password	<Invalid Password>				
				3. Click "Login" button					

Fundamentals

- Exhaustive Testing - Not Practical
- Software Testing is an attempt to test a small possible subset

- So, the basic questions

○ What to test?	TEST CASE SELECTION
○ How to test? (with how many test cases)	**STOPPING CRITERION**

Stopping Criterion

- Coverage Criterion
  - k test cases from each subdomain
- Behaviour Criterion
  - error rate is less than the threshold x

How to compare two test coverage criteria?

Idea of SUBSUME

A criterion C1 is said to subsume another criterion C2 iff,
for each use case system and for each set S of test objectives satisfying C1,
S also satisfies C2

Fundamentals

- Exhaustive Testing - Not Practical
- Software Testing is an attempt to test a small possible subset

- So, the basic questions
  - What to test? **TEST CASE SELECTION**
  - How to test? (with how many test cases) STOPPING CRITERION

Test Case Selection

- **Functional (Specification)**
- Structural (Code Structure)
- Data flow
- RANDOM

Functional Testing - Subdomains Method

- **Every distinct type of output**
  - Including every possible error condition
- **Special cases**
  - Tricky situations
- Common mistakes and misconceptions

The Standard Test for Functional Testing

Develop a good set of test cases for a program that accepts **three numbers** a, b, and c, interprets those numbers as the lengths of the sides of a triangle; and outputs **the type** of the triangle.

Types: Scalene, Isosceles, Equilateral

Two per desk - Work for 6 minutes

Left (to me): Writes the C code from specs [syntax not so important]

Right (to me): Writes the test cases parallely from specs (without looking at the code)

-----Three columns - Inputs / Expected Output / Remarks [don't fill this column]

Please don't forget to write your name on top

Next...

- Let us distribute these sheets to others
- As a team complete the remarks column write Pass/Fail/Wrong test case;
- Return the two sheets together

What did we do?

1. A specification
2. Implementation
3. Test case preparation
4. Testing

How effective was the coding?

Test Cases - Did we have more / less than necessary?

Test Case Selection

- Functional (Specification)
- **Structural (Code Structure)**
- **Data flow**
- **RANDOM**

Structural Testing

Every Statement Coverage - After seeing the program

Every Branch Testing - Control Flow Graph

Every Path Testing - Difficult, Exponentially Large, Infeasible combinations

Subdomain Testing - Partitioning the input domain into mutually exclusive subdomains and having equal number of test cases for each subdomain

Data Flow Testing

Definition of data - def

Use of data - use

- c-use computation
- p-use predicate

Definition free path - def-free path

Random Testing

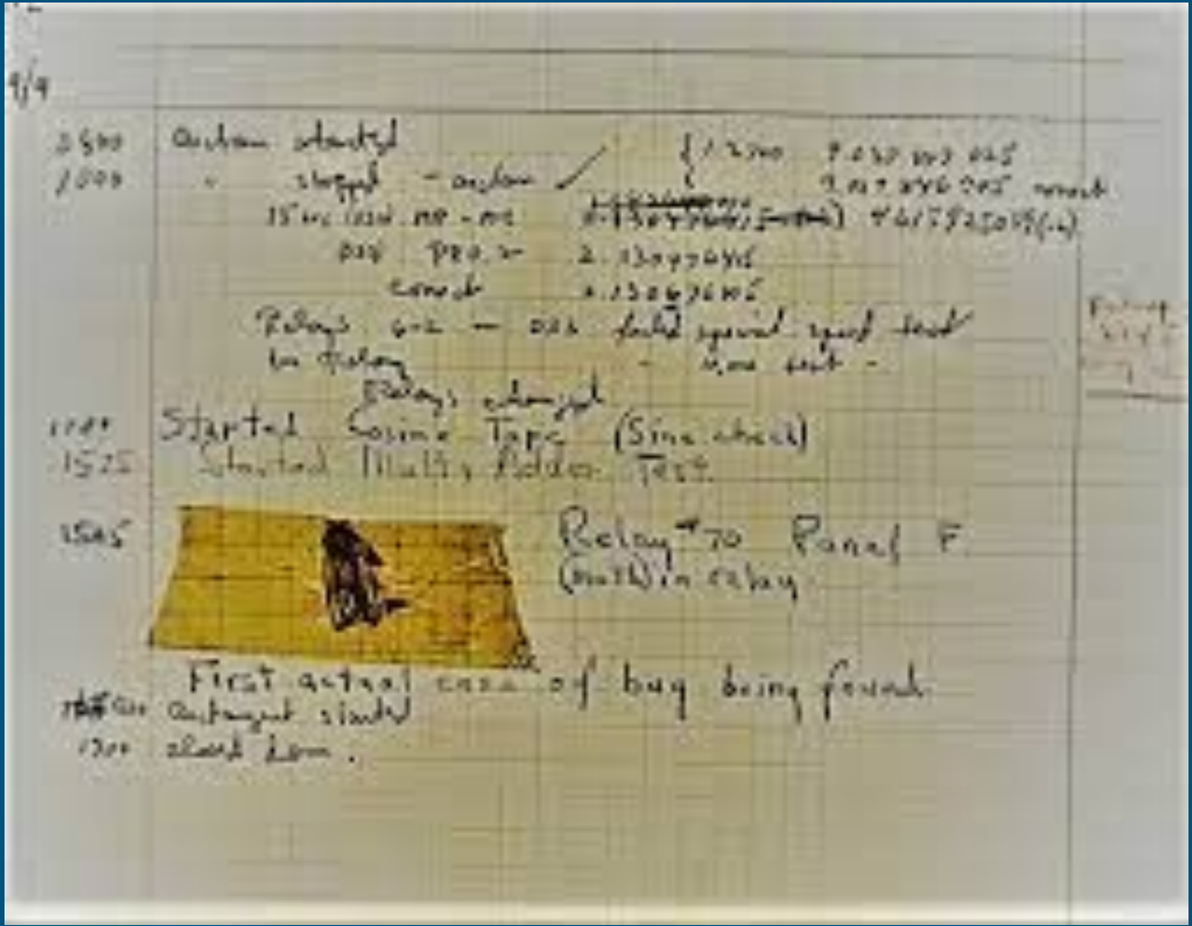
Advantage

Disadvantage

Testing and ?

Testing - Finding inputs that cause the software to fail

Debugging - The process of finding a fault given a failure



9/9

0800 Antan started

1000 " stopped - antan ✓ {1.2700 9.037 847 025

9.037 846 995 conct

13" uc (032) MP - MC ~~1.982147000~~ 4.615925059(-2)

(033) PRO 2 2.130476415

conct 2.130676415

Relays 6-2 in 033 failed special speed test

In relay " " 10.000 test .

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Multi Adder Test.

1545 Relay #70 Panel F

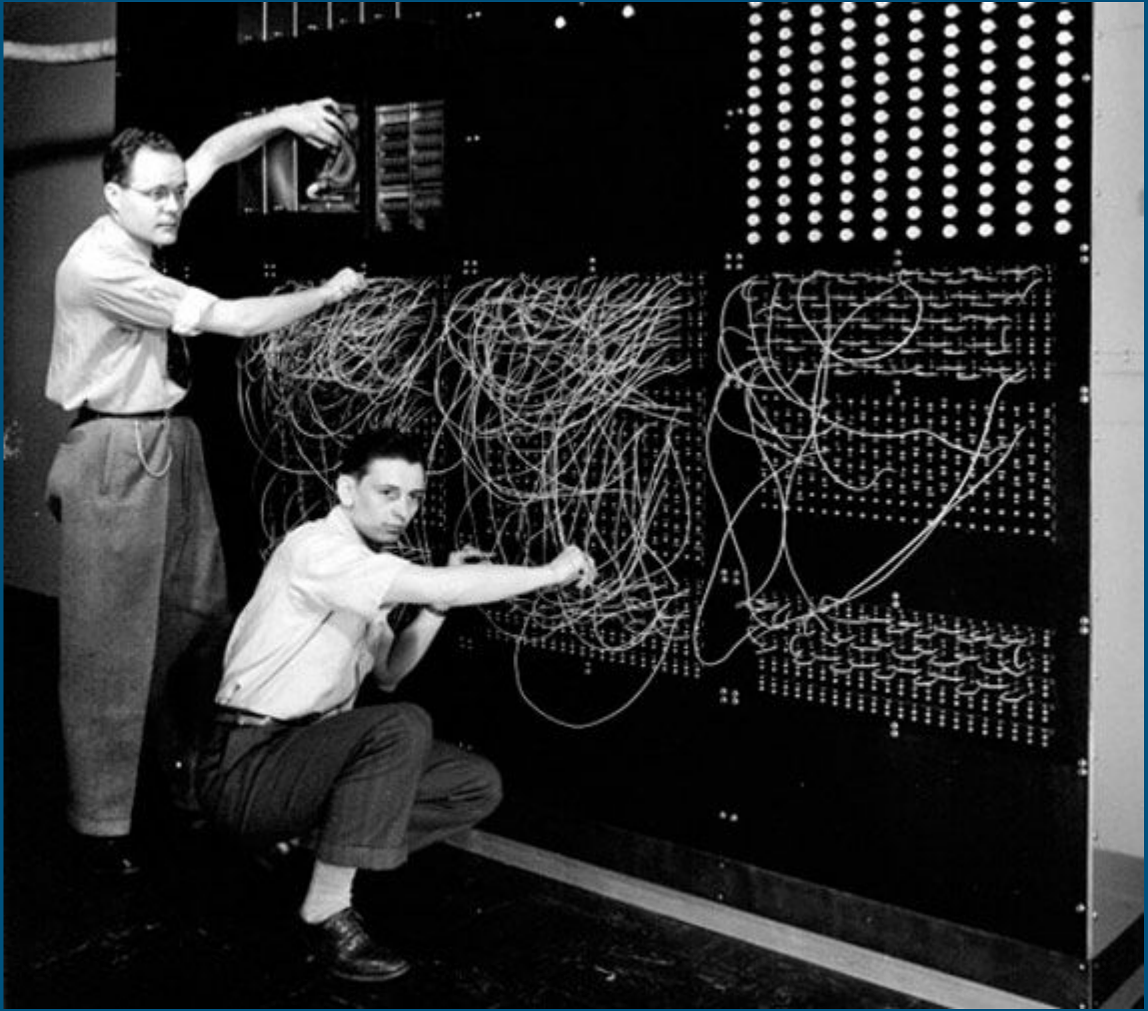
(moth) in relay.

First actual case of bug being found.

1630 Antangent started.

1700 closed down.

[illegible]

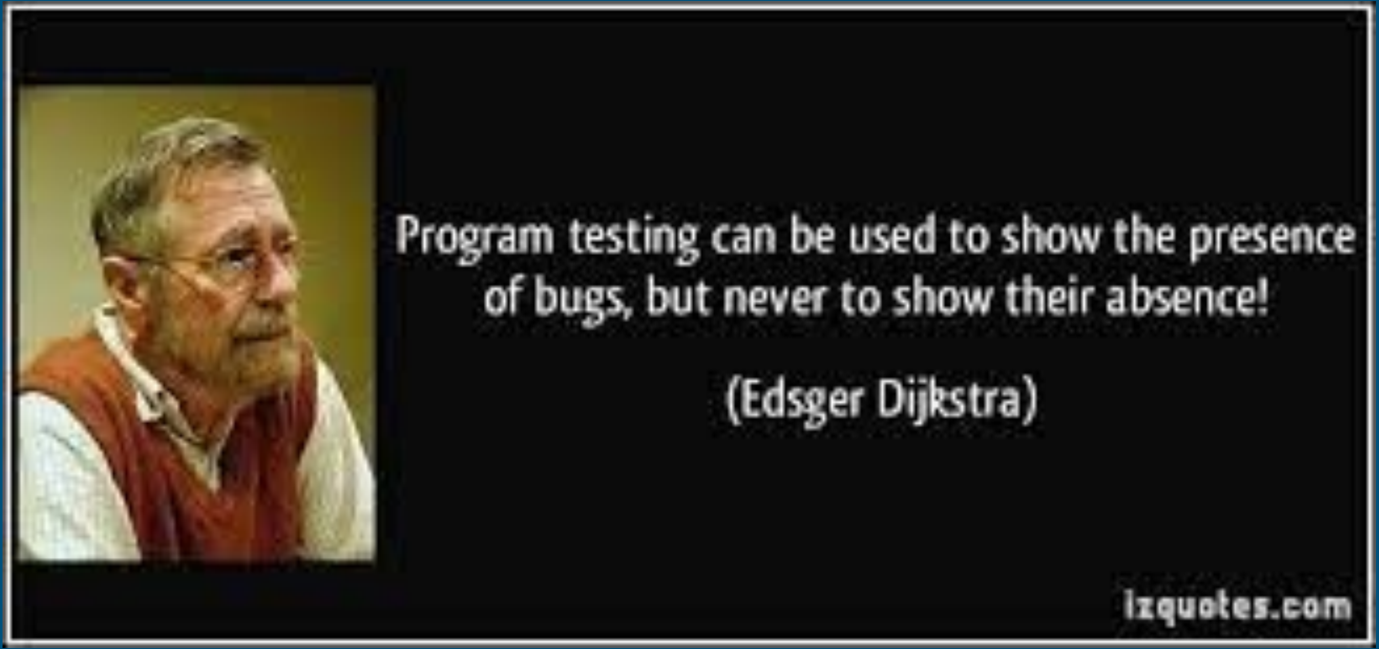


A good test

- *has a high probability of finding an error.*
- *is not redundant*
- *should be “best of breed”*
- *should be neither too simple nor too complex*

Testing Principles

- All tests should be traceable to customer/user requirements
- Tests should be planned long before testing begins
- The Pareto principle applies to software testing - Defects cluster together.
- Testing should begin “in the small” and progress toward testing “in the large” - Early testing saves money.
- Exhaustive testing is impossible - Absence of errors is a fallacy.
- To be most effective, testing should be conducted by an independent third party and beware of Pesticide Paradox

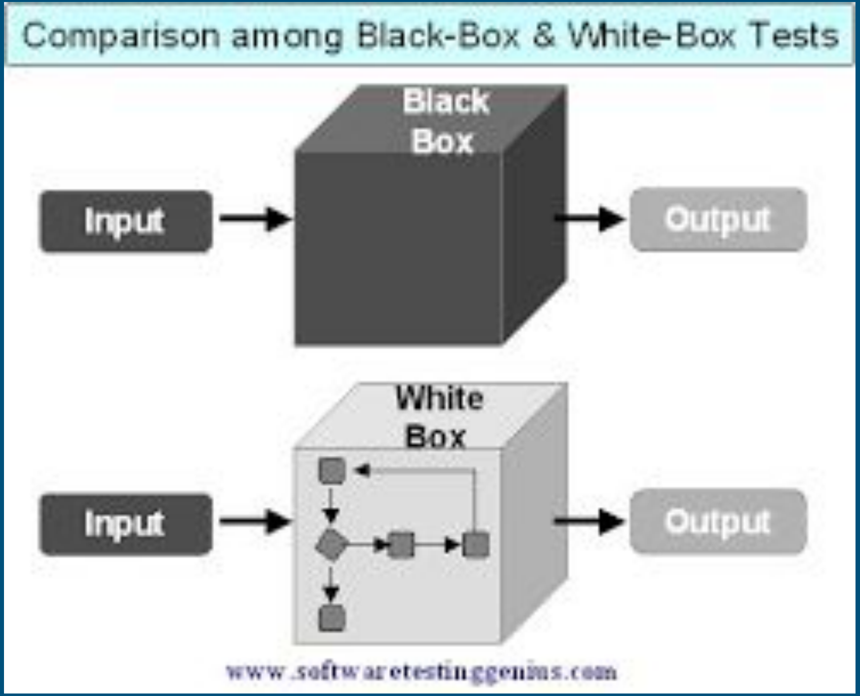


Program testing can be used to show the presence of bugs, but never to show their absence!

(Edsger Dijkstra)

lzquotes.com

Two Types of Testing


Comparison among Black-Box & White-Box Tests
Black Box
Input
Output
White Box
Input
Output
www.softwaretestinggenius.com

Blackbox

- Also called as *functional testing* or *behavioral testing,*
- Tests in the external point of view
- Specifications are used to generate test cases
- Tests for absence of features
- No programming knowledge is required

Whitebox

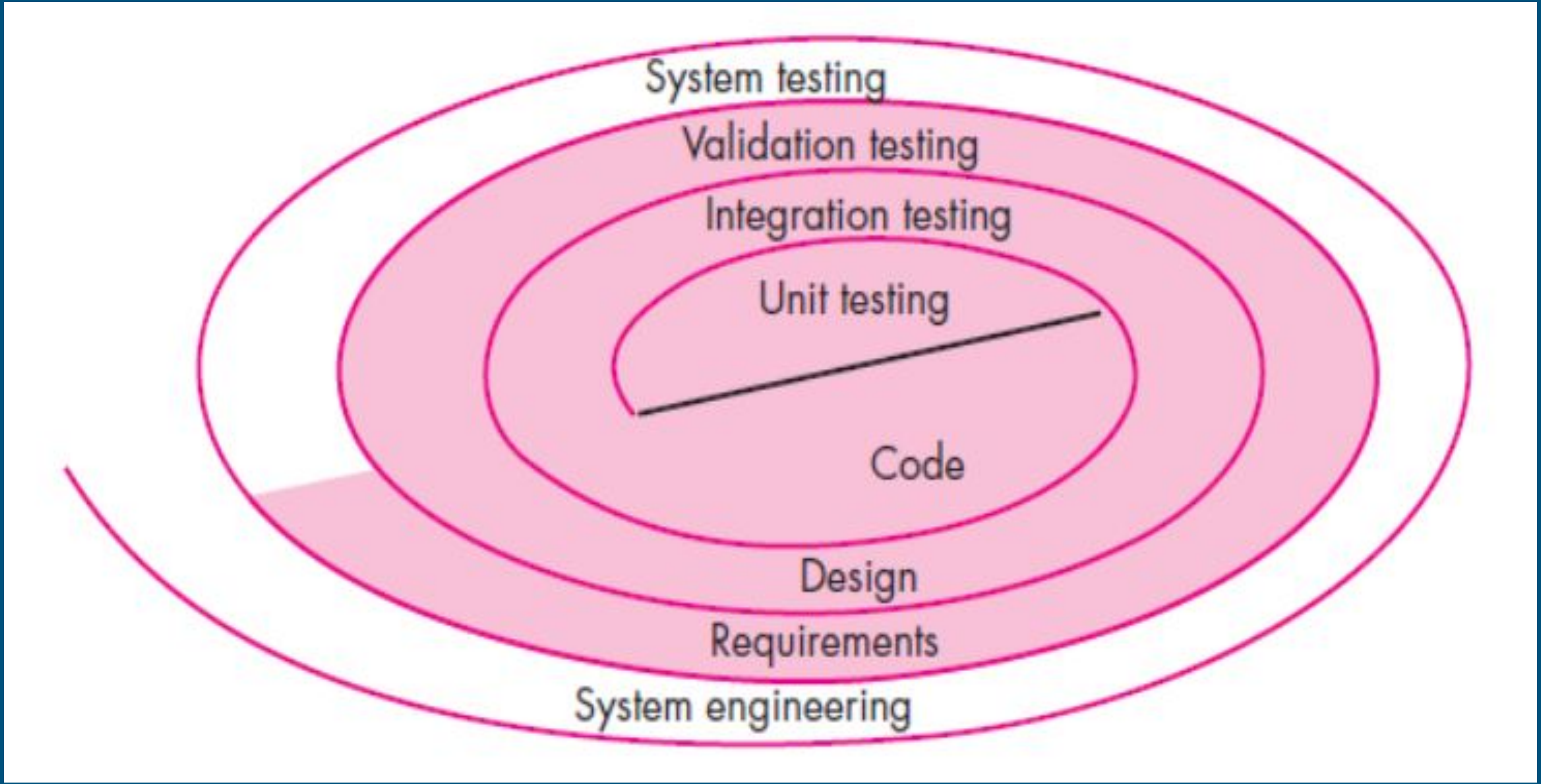
- Also called as *glass box testing* or *structural testing*
- Tests the internal point of view or implementation
- Cannot detect absence of features
- Coverage measures are used
  - ☐ Statement
  - ☐ Branch

V & V

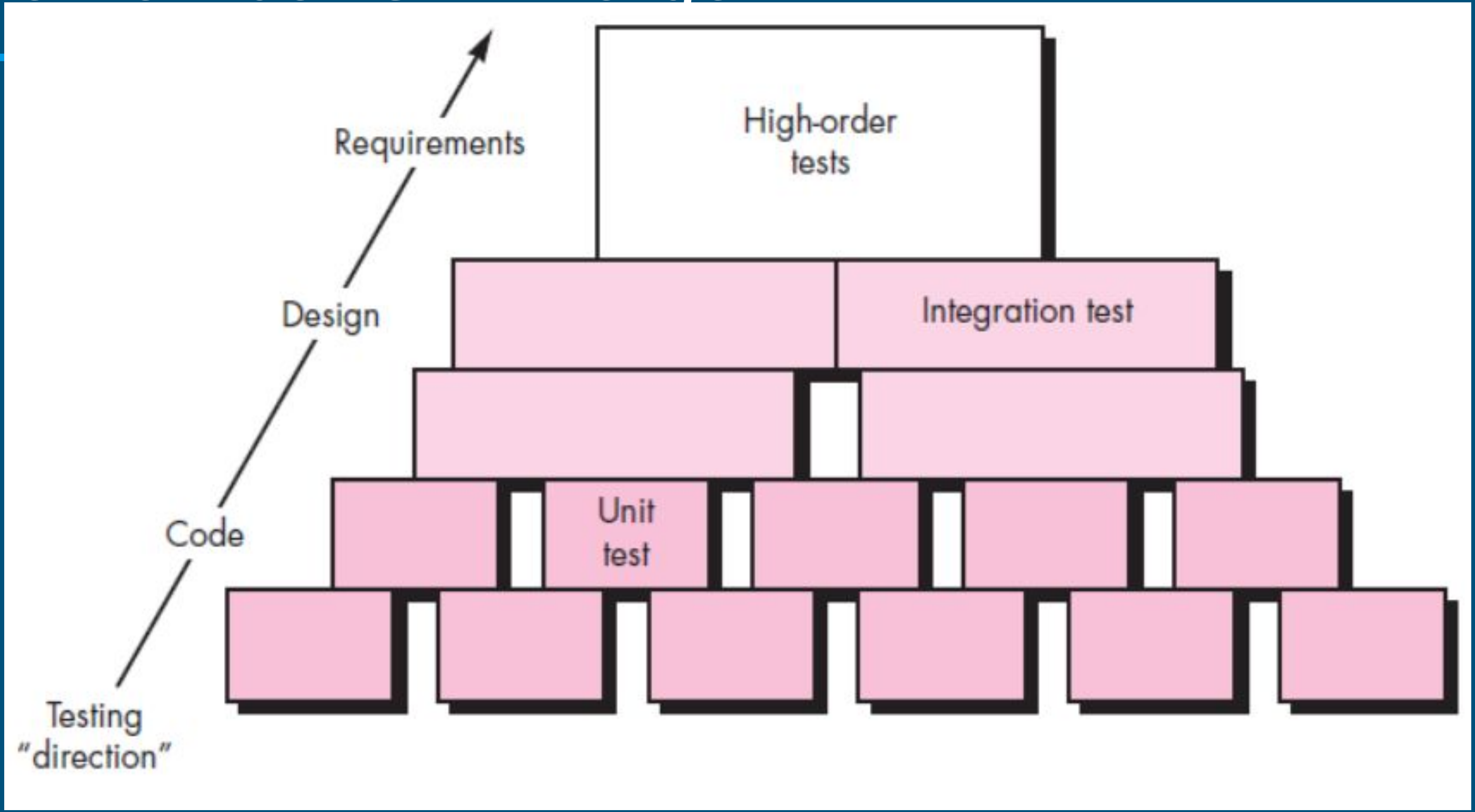
Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

The BIG Picture



Remember GIT Merge



Regression Testing [MOST IMPORTANT]

- *Re-execution of some subset of tests that have already been* conducted to ensure that changes have not propagated unintended side effects.

- Whenever software is corrected, some aspect of the software configuration (the program, its documentation, or the data that support it) is changed.

- Regression testing helps to ensure that changes do not introduce unintended behavior or additional errors.

Validation Testing

- Testing focuses on user-visible actions and user-recognizable output from the system.
- Validation **succeeds when software functions in a manner that can be reasonably expected by the customer**
- If Software Requirements Specification has been developed, it forms the basis for a validation-testing approach

Acceptance Test

- Alpha Testing
  - Conducted at the developer’s site by a representative group of end users in a controlled environment.
  - The software is used in a natural setting with the developer and records errors and usage problems.
- Beta Testing
  - Conducted at one or more end-user sites
  - The developer generally is not present.
  - It is a “live” application of the software in an environment that cannot be controlled by the developer.
  - The customer records all problems.

System Testing

- Recovery Testing
  - Systems must recover from faults and resume processing with little or no downtime.
  - It is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed
- Security Testing
  - Security testing attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper penetration
- Stress Testing
  - Stress testing executes a system in a manner that demands resources in abnormal quantity, frequency, or volume
- Performance Testing
  - It test the run-time performance of software within the context of an integrated system