

Testing

- * It is the process of checking the software to see if it's correct, complete, and good quality.
- * Testing helps to find errors before the software is released to users.
- * It also helps to find missing requirements.

Types of Testing

- i) Functional Testing
 - Tests what the software should do. (features, functions).
- ii) Non-Functional Testing
 - Tests how well the software performs. (speed, security, load).
- iii) Maintenance Testing
 - Tests software after updates or fixing.

(i) Functional Testing

1) Unit Testing

→ Testing small pieces of code.
like a function or module.

→ Done by developers.

2) Integration Testing

→ Testing when two or more
modules are combined.

→ checks if modules work together
correctly.

3) Smoke Testing

→ A quick basic test to check
if the build is stable.

→ Example: Does the app open?
Do main pages load?

4) Sanity Testing

→ Quick test done after small changes to check if the changes work.

→ Example: Developer fixes login,

Sanity tests check login only.

5) UAT (User Acceptance Testing)

→ Testing done by clients or end users to accept the system.

→ checks if the system meets their need.

6) White box Testing

→ Tester knows the internal code.

→ Tests logic, loops, conditions inside the code.

7) Black Box Testing.

→ Tester does not know the code.

→ Tests input vs output.

8) Regression Testing

→ Done after updates or bug fixes to check that old features still work.

→ Ensures nothing broke.

9) Localization Testing

→ Tests if software works for a specific region/language.

→ Example: Urdu version,

Arabic version.

10) Globalization Testing

→ Tests if software is ready to work worldwide.

→ Example: Supports multiple currencies, languages, time zones.

11) Interoperability Testing

→ Tests if the software works with other systems.

→ Example: Bank software talking to ATM Software.

(ii) Non-Functional Testing

* NFT checks how well the system works, not what it does.

1) Performance Testing:

→ speed of the system

2) Load Testing

→ How many users can use at once

3) Stress Testing

→ How system behaves under extreme load.

4) Volume Testing

→ Large amount of data.

5) Security Testing

→ protection from attacks.

6) Installation Testing

→ checks installation process

7) penetration Testing

→ Ethical hacking

8) compatibility Testing

→ works on different devices /
browsers

9) Migration Testing

→ Moving data from one system
to another.

(iii) Maintenance Testing.

- * It checks that the new updates, bug fixes, or enhancements did not break anything in the existing system.
- * It has more than 150 types of testing.

1) Regression Testing

2) Maintenance

x Difference Between Functional and Non-Functional Testing.

Functional	Non-Functional
i) Tests what the software does.	i) Tests how well the software does the work.
ii) checks the functions and features of the application.	ii) checks performance, speed, security, usability etc.
iii) performed first, before non-functional testing.	iii) performed after functional testing.
iv) Easy to perform manually.	iv) Tools are used.
v) Includes: unit, Integration, smoke, Sanity etc.	v) Includes: performance, load, stress, volume, security etc.
vi) Example: login, sign up, Add to cart.	vi) Example: How fast login works under load, how secure it is.

* Principles of Testing *

1) Exhaustive testing not possible.
→ You cannot test everything.

2) Defect clustering
→ most bugs are in few modules.

3) pesticide paradox
→ same tests find fewer bugs; ...
need new tests.

4) Testing shows presence of Defects
→ Testing finds bugs; not perfect software.

5) Absence of error fallacy.
→ Software can be bug-free but still wrong.

6) Early Testing

→ Start from requirements stage.

7) Testing is context-dependent

→ Different systems need different testing.

Testing Tools

1) Ranorex Studio

→ Automated functional testing.

→ testing desktop, web and mobile apps.

2) Selenium

→ Open-source automation testing tool

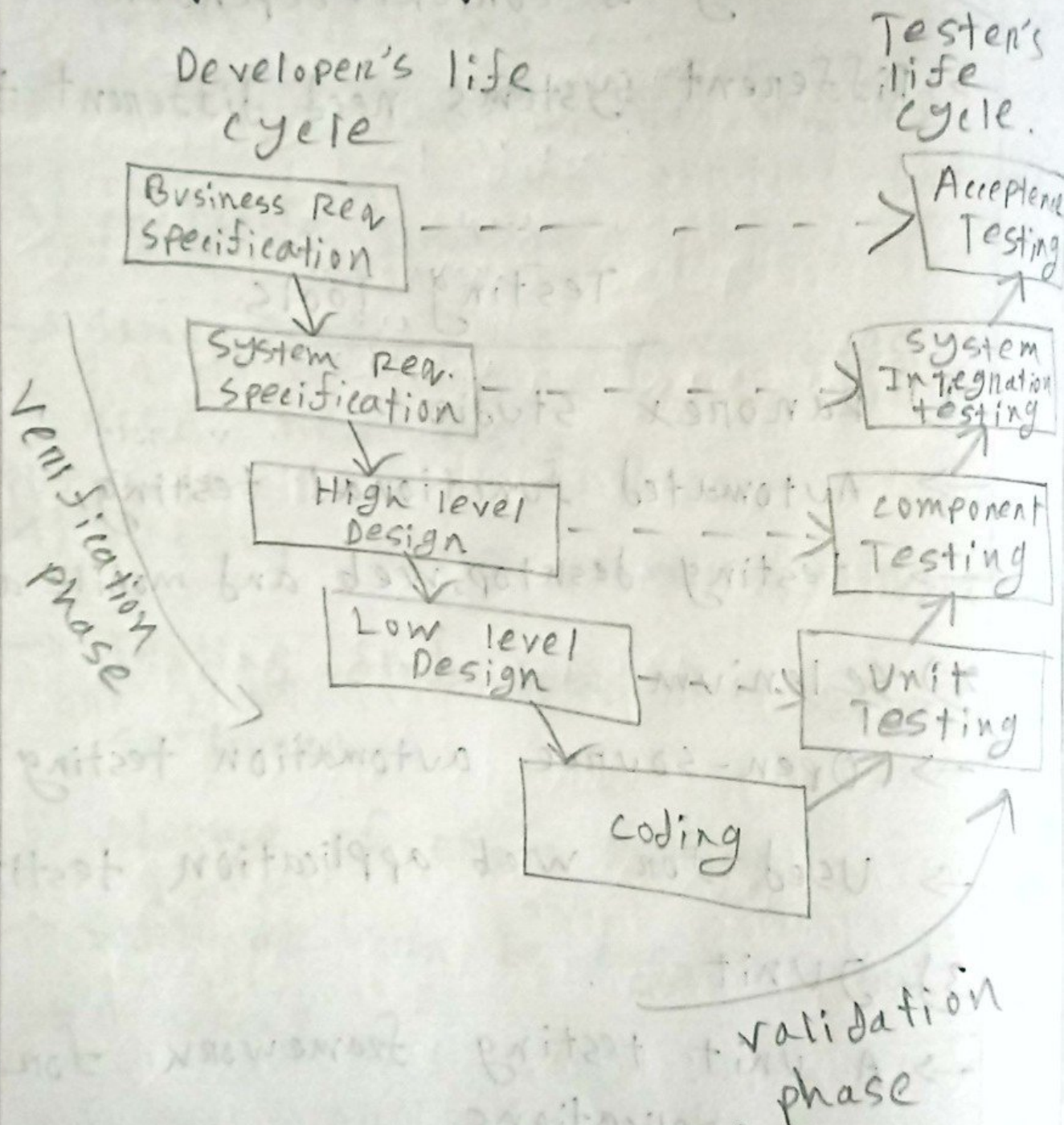
→ Used for web application testing

3) JUnit

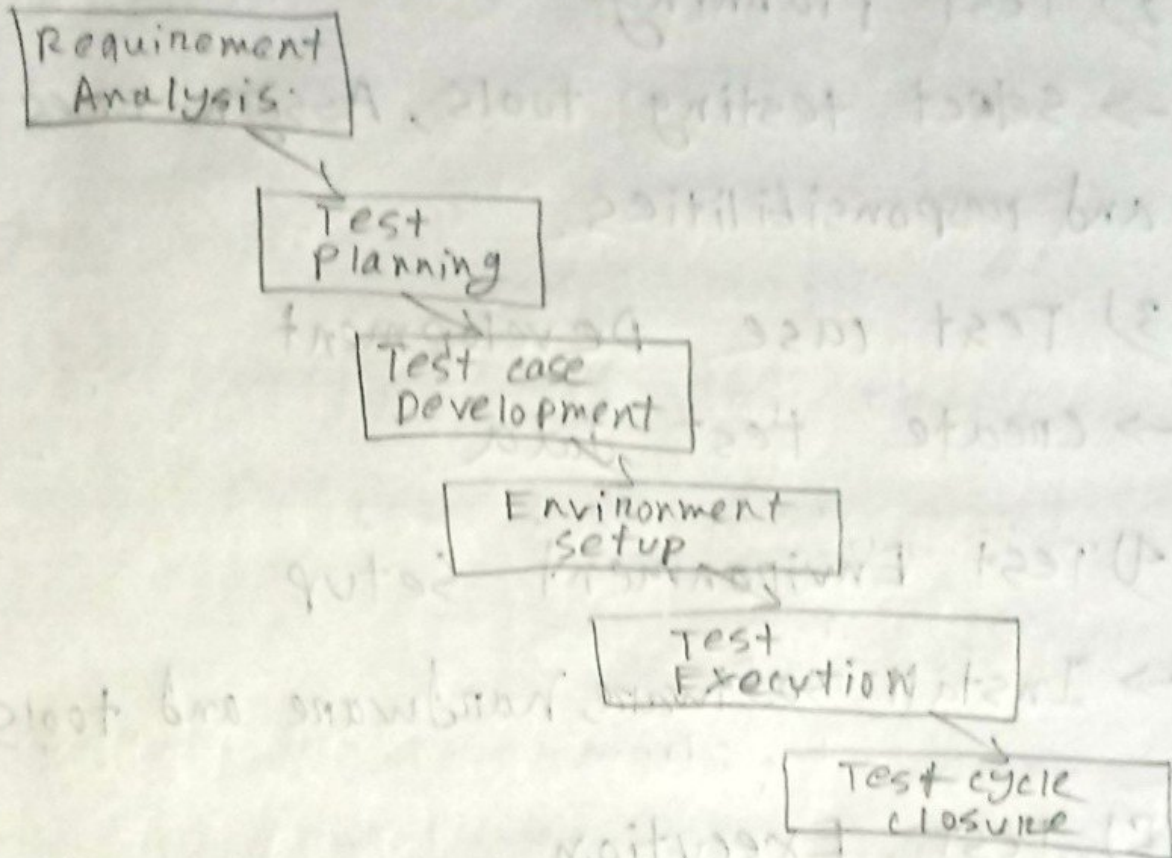
→ A unit testing framework for Java applications.

V model in ST

* There is a corresponding testing phase for each software development phase.



* Software Testing Life Cycle



* STLC is a sequence of activities carried out during the testing process.

1) Requirement Analysis

→ Understand what to test

→ prepare RTM (Requirement Traceability matrix)

2) Test planning

→ Select testing tools, Assign roles and responsibilities.

3) Test case Development

→ Create test data

4) Test Environment Setup

→ Install software, hardware and tools.

5) Test Execution

→ Run test cases

→ Map defects to RTM

6) Test cycle closure

→ Check if testing is complete

→ Create test closure report.

Sample Questions.

1) Generate test ideas to test an alarm clock.

Ans: Below are the test ideas,

i) Functional Testing

- check if the clock shows correct time.
- check if the alarm rings at the set time.
- check if stop button stops the alarm immediately
- check if SNOOZE button works correctly.
- Check if alarm rings again after snooze interval

ii) Domain/Boundary Testing

- Check if alarm rings before the set time
- check if alarm keeps ringing after set time.

- Check how long alarm rings if User does not press any button.
- Press Snooze + Stop together and observe behavior.
- Change the time after setting alarm and check what happens.

iii) performance Testing

- Check if alarm rings instantly at exact time (no delay).
- Set alarm repeatedly every 10 minutes for 1 hour and observe stability.
- Check if alarm rings with low battery.
- Press stop button many times check if button still works smoothly.

iv) Usability Testing

- check how easy it is to set the alarm
- check if buttons are easy to identify.
- check if alarm sound is loud enough.
- check if battery is easy to remove and replace.

v) Interoperability Testing

- check if alarm clock fits commonly available stands.

vi) Maintainability Testing.

- check if spare parts are available.
- check what happens to stored time if unused for long period.

2 | Give test ideas to test a basic calculator.

Ans:

i) Functional Testing

- Check addition, subtraction, multiplication, division.
- Check division by zero error
- Check square root, percentage, memory functions.
- Check clear (C) and all clear (AC) buttons.

ii) Domain/Boundary Tests

- Very large numbers.
- Negative numbers
- Decimal values
- Long arithmetic expressions.

iii) performance Testing

- Fast repeated calculations.
- Long continuous usage without freezing.

iv) Usability Testing

- Are buttons easy to press?
- Are numbers clearly visible?
- Is the display readable in dim or bright light?

v) Interoperability

- Does it work with different battery types?

vi) Maintainability

- Battery replacement, Key durability, Display quality.

— x —