Quality Assurance Practices - CSC2213
Lecture 6

Binuri Raigamkorale
binuriyr@dcs.ruh.ac.lk

Course Outline

- 6.1 Software testing overview
  - What is software testing?
  - Who is responsible for quality of a software?
  - Why testing is important?
- 6.2 Basics of Software Testing
  - Testing phase in SDLC
  - Bug Life Cycle and reporting
- 6.4 Types of testing
  - Manual testing
  - automation testing
  - unit testing
    - 3A’s in Unit Testing, Structure of a Unit Test, Unit test Frameworks
  - integration testing
  - system testing
  - non functional testing
  - regression testing
- 6.3 Testing Concepts
  - black box testing
  - white box testing
- 6.5 SQL for Software Testing
- 6.6 Testing in different domains - Backend, Web and Mobile testing fundamentals
- 6.7 Best practice in Software Testing

Objectives

- To have a solid understanding about software testing
- The understanding about Software testing tools SW testing
- testing types and testing approaches (white box, black box testing)

6.1 Software testing overview

6.11 What is Software Testing?

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.

In other words, **software testing** is an activity which involves **controlling** the **quality** of the **software** and to ensure that it matches with the **“Software requirements”** and makes our customers happy.

6.12 who is responsible for testing?

“All Team” responsible for **Quality of the software**. The main **goal** of the team is **delivering** a **high-quality product**, **efficiently** and in a **short** time.
All team members should focus on quality with the guidance of the **software tester**. Software tester should lead the team in terms of quality and make sure that **the product meets with the customer needs.**

6.13 Why software testing is important ?

In IT industry, software testing was underestimates and as consequence, many companies had lots of major issues which caused millions of damage to them.

6.1 Basics of software testing

6.13 Why software testing is important ?

Example of the live issues: Knights $440 million error

One of the biggest American market makers for stocks struggled to stay afloat after a software bug triggered a 440 million dollars in just 30 minutes . The firm’s shares lost 75 percent in two days after the faulty software flooded the market with unintended trades.

- One of the main responsibility of software tester is finding defects as soon as possible.

  Imagine if you find a bug during the testing phase:

  Cost: the cost of testers retesting bug fixes / finding bugs

- But if you find an issue on final product the cost will be :

  The cost of customers finding bugs, plus help desk, plus escalations, plus bug fixes, plus retesting, plus redeploying.

6.2 Basics of Software Testing

What is Testing :

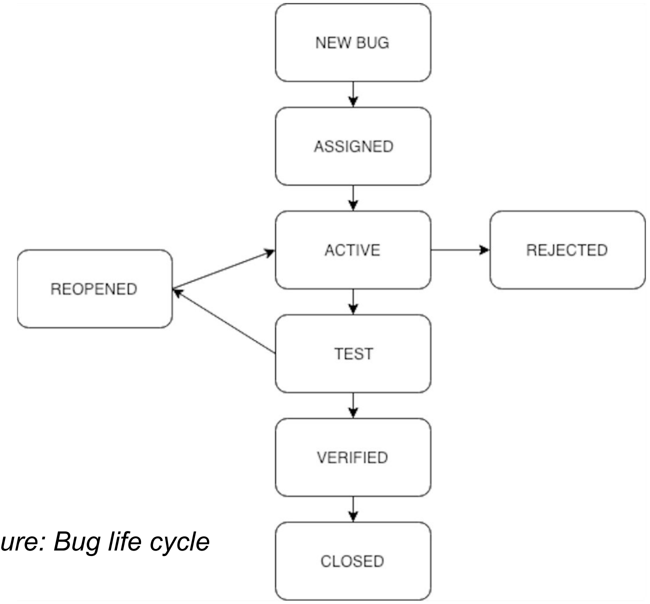
It **e**xplains all about **“Did we get what we want?”**

In IT world, delivered software equal to “Food preparing”.
At the end of the Software Development Life Cycle activities, you can deliver the end product to customers. Each part should be perfect in order to receive a great final product.

6.2 Basics of Software Testing

What is bug life cycle?

- It’s a cycle in which a defect goes through during it’s lifetime
- It starts when a bug is found and ends when it is closed
- Bug has different states in the life cycle.



NEW BUG

ASSIGNED

ACTIVE

REJECTED

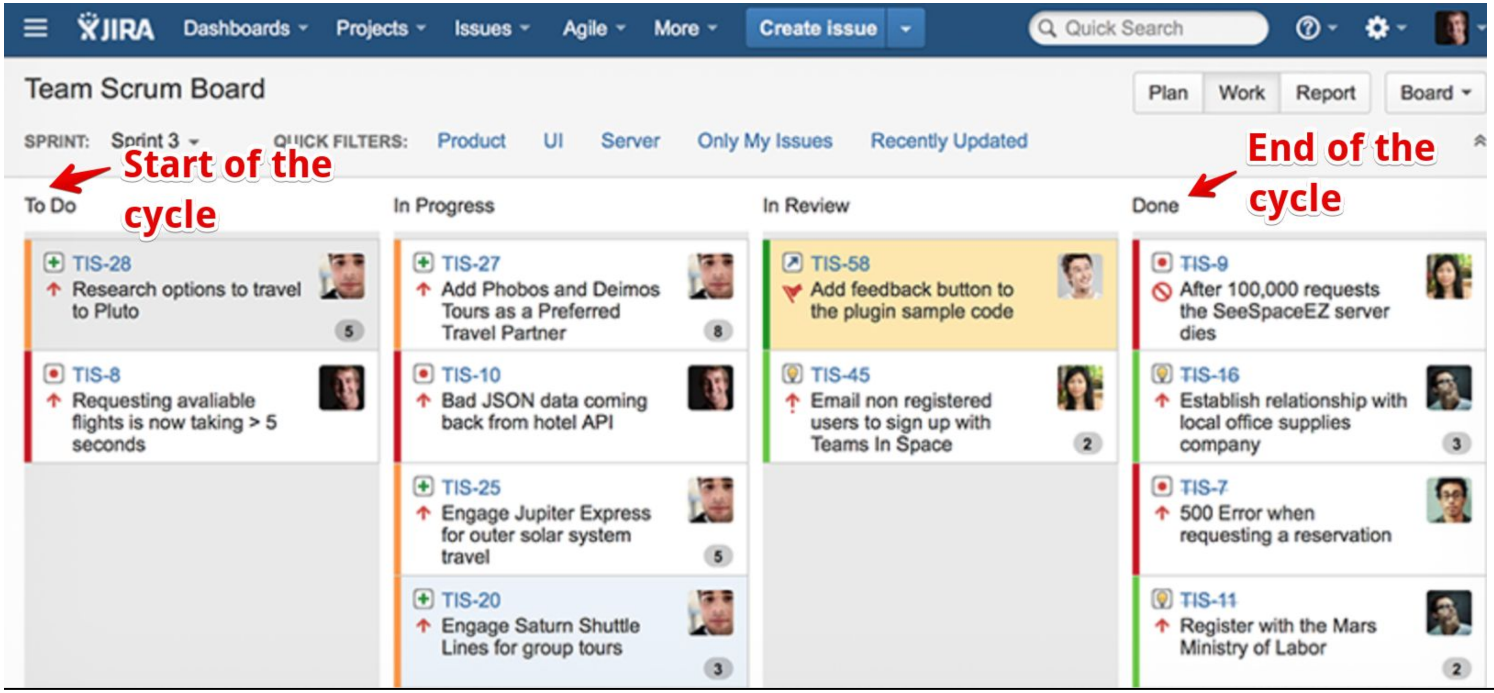
REOPENED

TEST

VERIFIED

CLOSED

Figure: Bug life cycle



Team Scrum Board

SPRINT: Sprint 3 QUICK FILTERS: Product UI Server Only My Issues Recently Updated

Start of the cycle

End of the cycle

To Do	In Progress	In Review	Done
TIS-28 Research options to travel to Pluto 5	TIS-27 Add Phobos and Deimos Tours as a Preferred Travel Partner 8	TIS-58 Add feedback button to the plugin sample code	TIS-9 After 100,000 requests the SeeSpaceEZ server dies
TIS-8 Requesting avaliable flights is now taking > 5 seconds	TIS-10 Bad JSON data coming back from hotel API	TIS-45 Email non registered users to sign up with Teams In Space 2	TIS-16 Establish relationship with local office supplies company 3
	TIS-25 Engage Jupiter Express for outer solar system travel 5		TIS-7 500 Error when requesting a reservation
	TIS-20 Engage Saturn Shuttle Lines for group tours 3		TIS-11 Register with the Mars Ministry of Labor 2

Figure: bug state in JIRA

6.3 Types of testing

Manual testing

- Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools.
- The process of manual testing is greatly dependent on the skills of the tester.
- A tester performs the end-user role and verifies if all the features are working properly or not. Tester manually executes the test cases.

Challenges

- We cannot gather all information at one time
- Lots of discussions needed
- Sometimes the speed of discussion would be too fast or too slow to understand

Best Practice

- Draw diagram or rough block code for better understanding and for reference

6.3 types of testing

Automation testing

What is automation testing?

It performs using automated testing software tools to execute a test case.

Why Automation testing ?

test Automation in software testing does not require Human intervention. Non Technical testers can be involved in code

Minimum usage of code

Which test cases to automate?

High Risk – Business Critical test cases

Test cases that are repeatedly executed

Test Cases that are difficult to perform manually

Test Cases which are time-consuming

6.3 types of testing

Unit testing

- A unit test is code written by a developer that tests as small a piece of functionality (the unit)
- Here we'll test one thing at a time
- This is developer's job
- One function may have multiple unit tests according to the usage and outputs of the function

- A good unit test is,
  - Isolated/independent
  - Repeatable
  - Fast
  - Self-Documenting

6.3 types of testing

Unit Testing

3A s in Unit Testing

- Arrange : You’re arranging things prior to calling the method/function of interest
- Act : Action stage, calling a method or function
- Assert :The part that ensures that your expectations are met

Structure of a Unit Test

- Setup
- Prepare an input
- Call a method
- Check an output
- Call for TearDown

6.3 types of testing

Unit Testing

Unit test Frameworks

- Framework support is available for Unit testing
- It's standard, everybody writes their tests the same way, so everybody knows:
  - Where to find tests
  - How to run tests
  - How to interpret results
  - How to add/modify tests
- Testing is easily automated and also integrates with many IDEs

Automated unit test softwares:

Junit

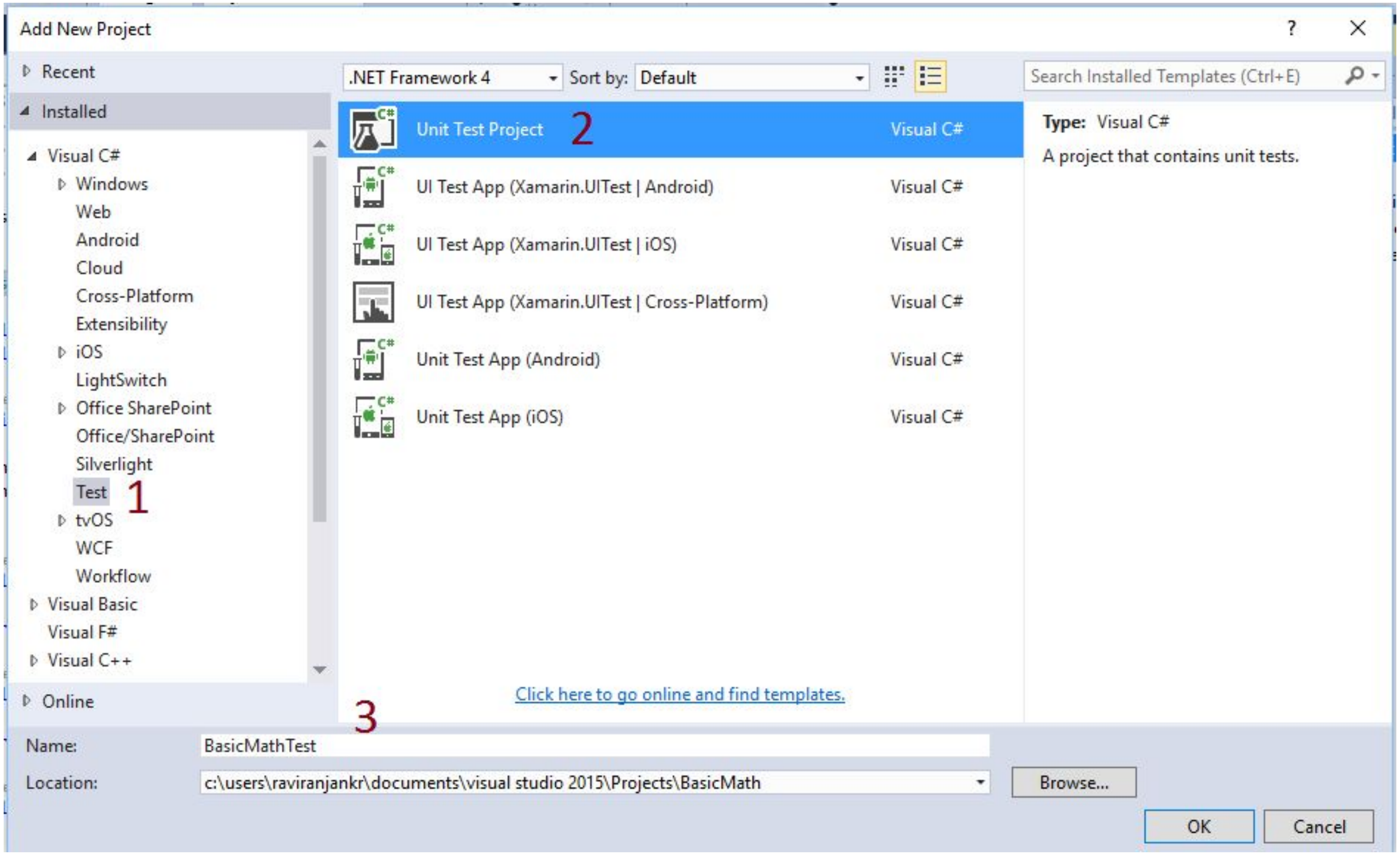
Nunit

PHPUnit

6.3 types of testing

Unit Testing

Will discuss more in the practical class



Add New Project

.NET Framework 4

Sort by: Default

Unit Test Project	Visual C#
UI Test App (Xamarin.UITest | Android)	Visual C#
UI Test App (Xamarin.UITest | iOS)	Visual C#
UI Test App (Xamarin.UITest | Cross-Platform)	Visual C#
Unit Test App (Android)	Visual C#
Unit Test App (iOS)	Visual C#

Type: Visual C#

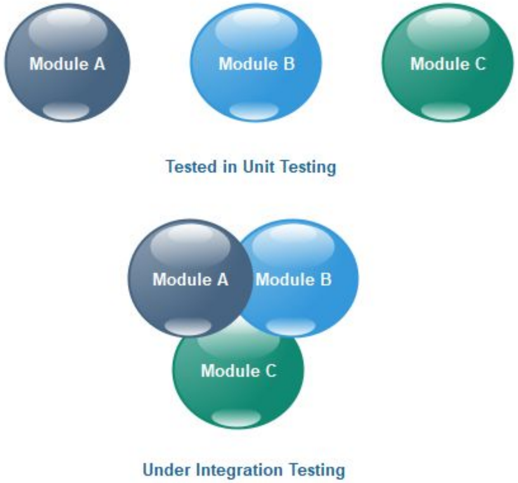
A project that contains unit tests.

Name: BasicMathTest

Location: c:\users\raviranjankr\documents\visual studio 2015\Projects\BasicMath

6.3 types of testing

Integration testing



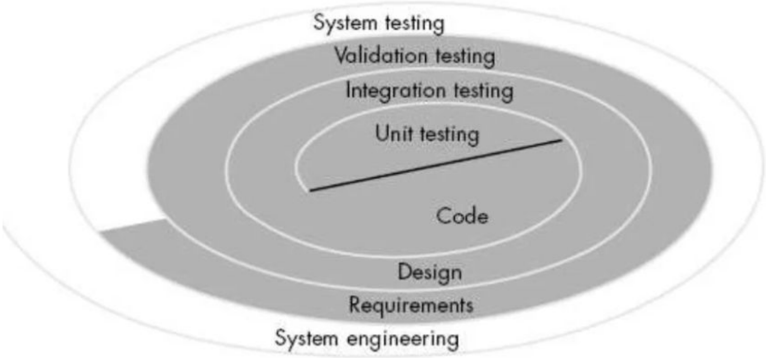
software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers.

The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated.

System testing

validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.

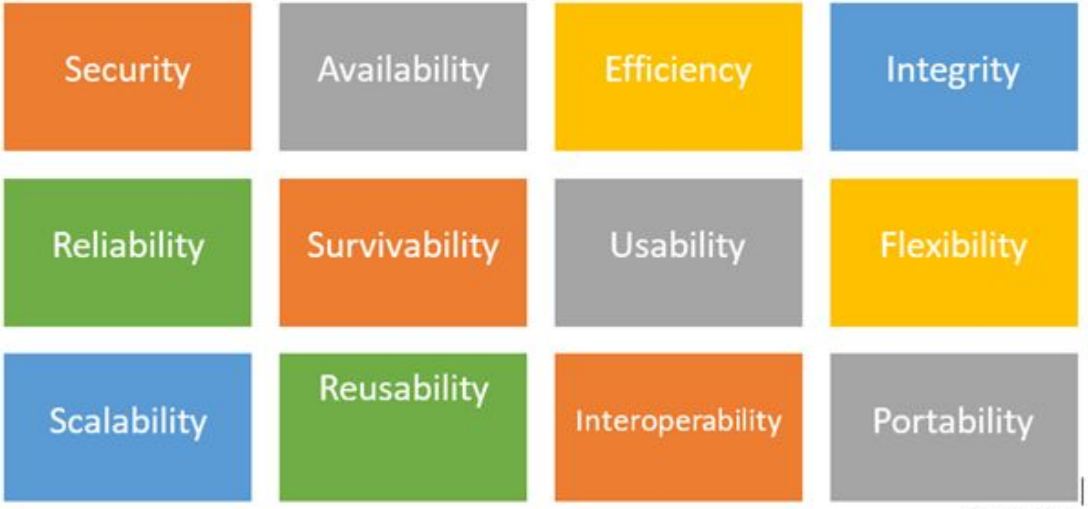
Ex:
recovery testing
Migration testing



6.4 types of testing

Non functional testing

To check non-functional aspects (performance, usability, reliability, etc.) of a software application



Security	Availability	Efficiency	Integrity
Reliability	Survivability	Usability	Flexibility
Scalability	Reusability	Interoperability	Portability

6.3 Testing Concepts

White box testing

- White box means you get a little, or a lot more, than a superficial view of the product. The focus of white box, is **what is inside.**

- White box testing involves the testing of the software code for the followings:
- The functionality of conditional loops
- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- Testing of each statement, object, and function on an individual basis

- **Types of White Box Testing**

  Unit testing

6.3 Testing Concepts

Black box testing

- Black box testing is a software testing method, in which the internal implementation of the function is not known by the tester.
- This method is named “Black box”, because the software program - in to the eyes of the tester- is like a black box where the inside is **not visible.**

Types of blackbox testing

Functional testing - This black box testing type is related to the functional requirements of a system; it is done by software testers.

Non-functional testing - is defined as a type of Software testing to check non-functional aspects, such as performance, usability, reliability, etc.

Regression testing - is where we are testing existing software applications to make sure that a change or addition hasn’t broken any existing functionality.

6.3 Testing Concepts

- Blackbox testing methods attempt to find errors in:
  - Incorrect or missing function
  - Interface / Visual errors
  - Behaviour issues
  - Initialization and termination errors.

- Black box testing techniques

Equivalence Class Testing: A black-box test technique in which test cases are designed to exercise one representative member of each partition.
aim: to reduce the number of repetitive test cases.

Boundary Value Testing: Boundary value testing is focused on the values at boundaries.

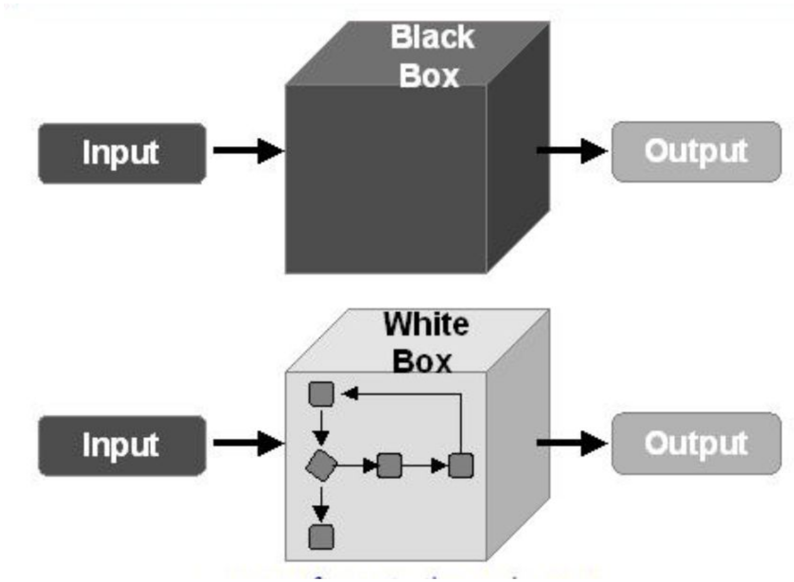
Decision Table Testing: Is a visual representation for which actions to perform depending on given conditions.

State Transition Testing

Use case testing

6.4 Testing Concepts

White box testing vs BlackBox testing



Black Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is **NOT** known to the tester.

White Box Testing is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.

6.5 SQL for Software Testing

SQL is great tool to test and verify if the application store and retrieve the data correctly . Also insert new users or fields to create “test data”.

Using SQL queries you can reach user information in the database , you can insert new users for testing, and update/delete the users.

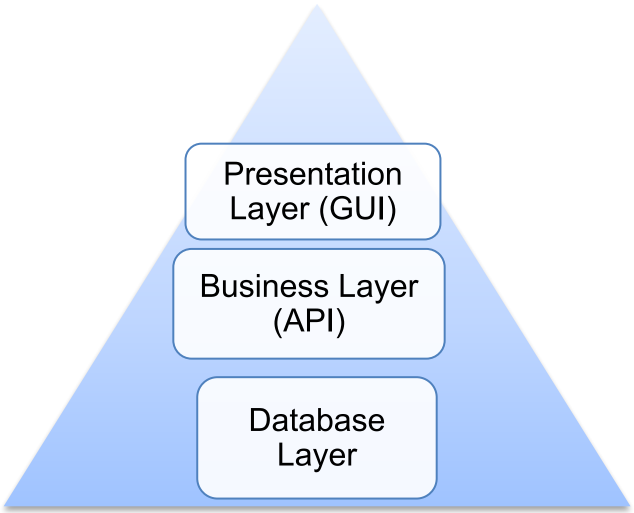
Ex: Keys and indexes testing
Database, column testing
stored procedure testing
triggers testing
data integrity and consistency testing

SQL queries types will used for testing:
Select (distinct) count
Select NULL
Where clause
Group by
Sub queries
Exists and not exists , ALL

https://swansoftwaresolutions.com/the-importance-of-sql-skills-for-software-testers/

6.6 Testing in different domains

Backend testing



- A typical web application has three layers**:** **front-end** (user interface), **business logic**, and a **database**.
- Testing the frontend involves **validating** those parts of the application that are **visible to the end-users**
- **Backend testing**, deals with all those elements that **users can't see**. This means **validating** the application's **database** and **API** that represent its **business logic.**
  (API is the business layer of the software that creates a connection between the presentation layer and the data layer.)

Ex:
Imagine a banking app where the user can see **a list of bank accounts** on **front-end**.
Behind that, there can be a **multiple API's**, the application use to show data on the page.

for example,
One API to fetch all **active user** accounts,
Another API to **fetch inactive user** accounts and
Another API to **delete** an account.

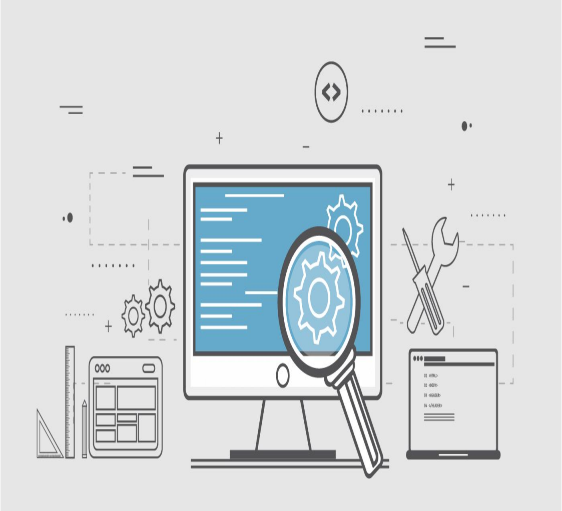
6.3 Testing in different domains

Benefits of API testing

- You can start testing before the front-end is ready and fix logic-related bugs much cheaper
- Discover bugs that can damage other products and business processes if your API is public
- Increase the test coverage and its accuracy
- You can run regression tests to ensure that the recent changes didn’t break your APIs and the systems using them.

6.3 Testing in different domains

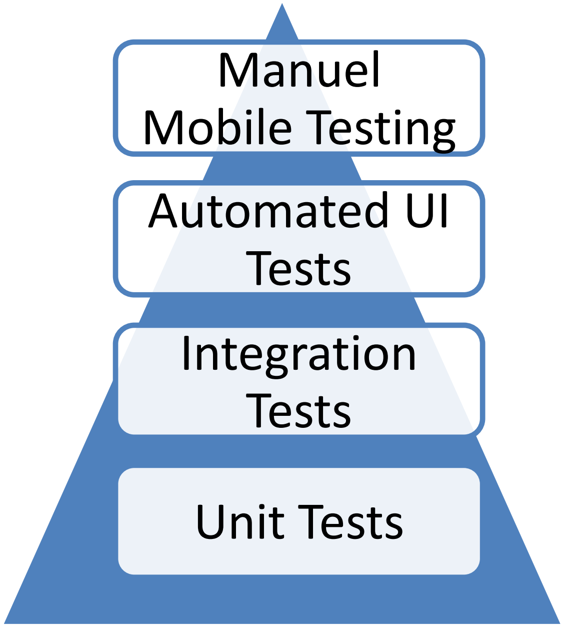
- Web testing



Website testing is **very hard due to high complexity.** It might look simple to open a website and use it but think about the **different variations and environments**.

6.3 Testing in different domains

- Mobile testing

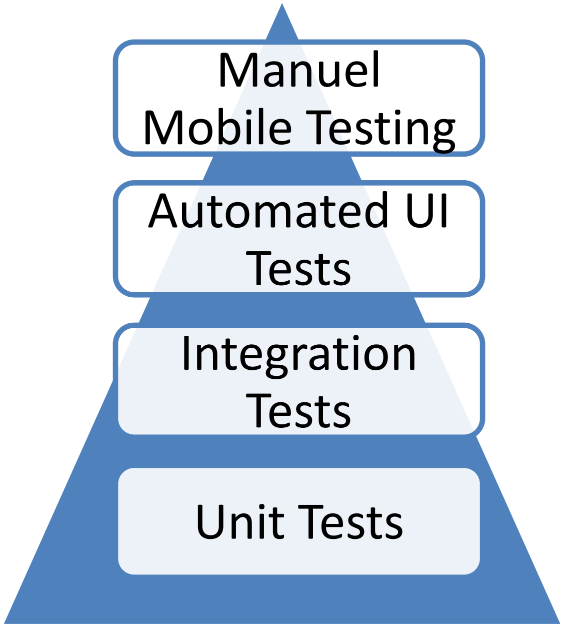


Unit tests:
are the fundamental tests in our application testing strategy. By creating and running unit tests against our mobile code, we can easily verify that the logic of individual units is correct. Running unit tests after every mobile build helps us to quickly catch and fix & changes to your app.

Integration test:
applications will integrate with some other parts such as databases, file systems, network calls to other applications. When writing unit tests these are usually the parts you leave out in order to come up with better isolation and **faster tests.**
It test the **integration** of your application with all the parts that live outside of your application.

6.3 Testing in different domains

- Mobile testing



Automated UI tests

checks if the user interface of your application works correctly. User input should trigger the right actions, data should be presented to the user and the UI state should change as expected.

UI tests are most fragile ones since they rely on many different aspects, such as internet speed, device status and so on. That's why UI tests at the top of the testing pyramid, and therefore the number of UI tests should be limited.

Manual Mobile Testing:

These tests performed manually by Mobile Test Engineers. It is important to consider different facts and use different device set when we perform testing.

Manual mobile testing is a really important task since we test the system as a whole. All the application layers perform to provide us the UI and we are interacting with our system as our users do

6.7 Useful Tools in Software Testing

.

TestLodge

Jira : bug reporting
Git hub : bug reporting
Confluence: bug documentation
Testrail : testcase management
TestLodge enhances the testing process by providing way to organize, collaborate, and track progress of your tests.

6.7 Best Practice in Software Testing

- Maintaining a Dedicated Software Security Testing Team
- Pair Programming to Measure Code Quality
- Run Regular QA Technical Reviews
- Understand Product Objectives in Order to Design Effective Test Strategies

Activity

1. You need to implement following method to facilitate cash withdrawal activity in a banking system

- Method Name: CashWithdrawal
- Variables:
  - currentBalance <<double>>
  - requestedAmount <<double>>
- If requestedAmount exceeds currentBalance, then method should display an error message to the user

2. Which things to consider testing a mobile application through **Black Box Technique ?**

- By testing your application on multiple devices.
- By changing the port and IP addresses to make sure the device is getting connected and disconnected properly.
- By making calls and sending messages to other devices.
- By testing your web application on different mobile browsers like Chrome, Firefox, opera, dolphin etc