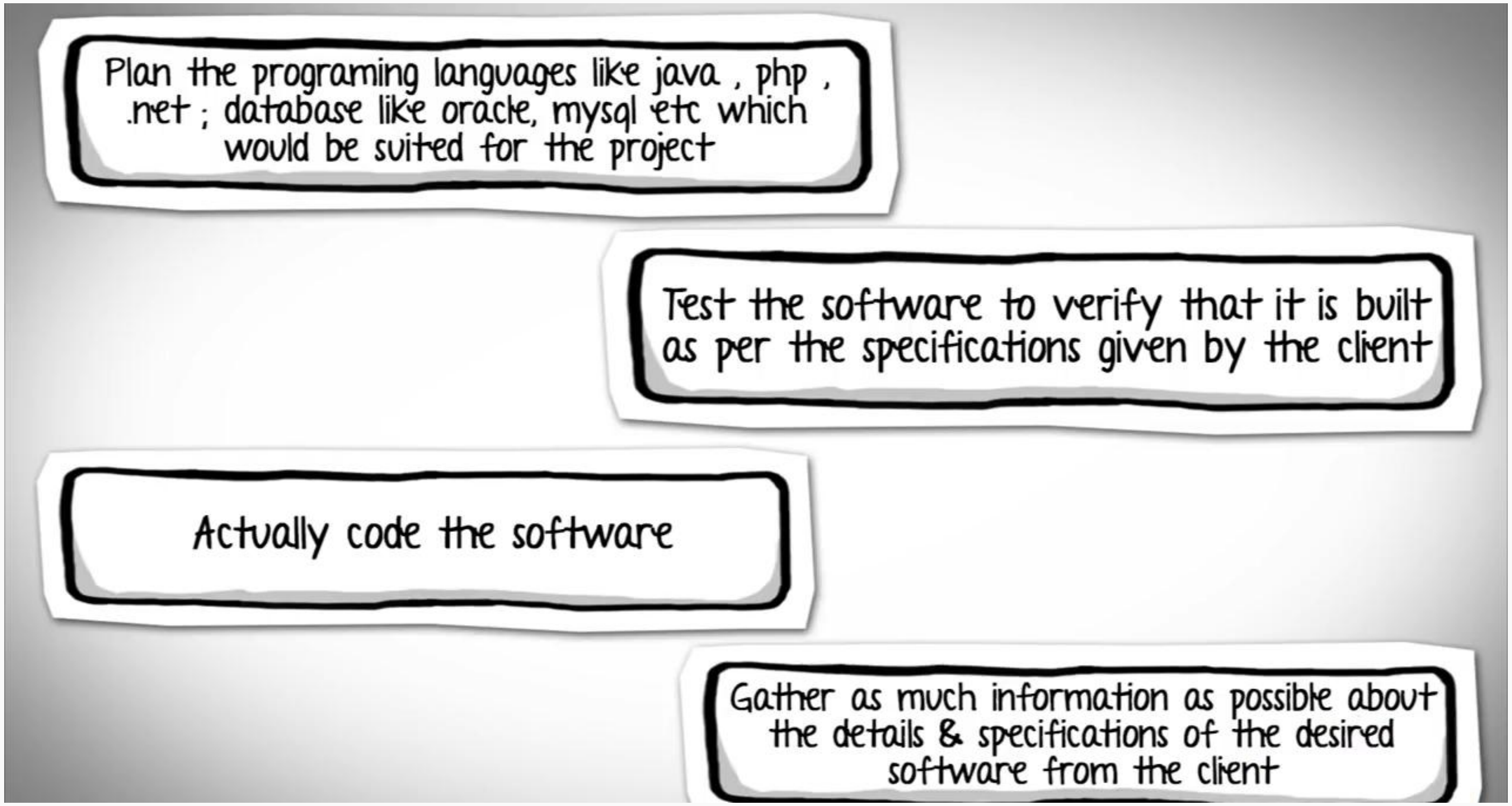
Sofware Testing

#02 - Testing on Software Life Cycle

Herman Kabetta

SUPPOSE, YOU ARE ASSIGNED A TASK, TO DEVELOP A CUSTOM SOFTWARE FOR A CLIENT

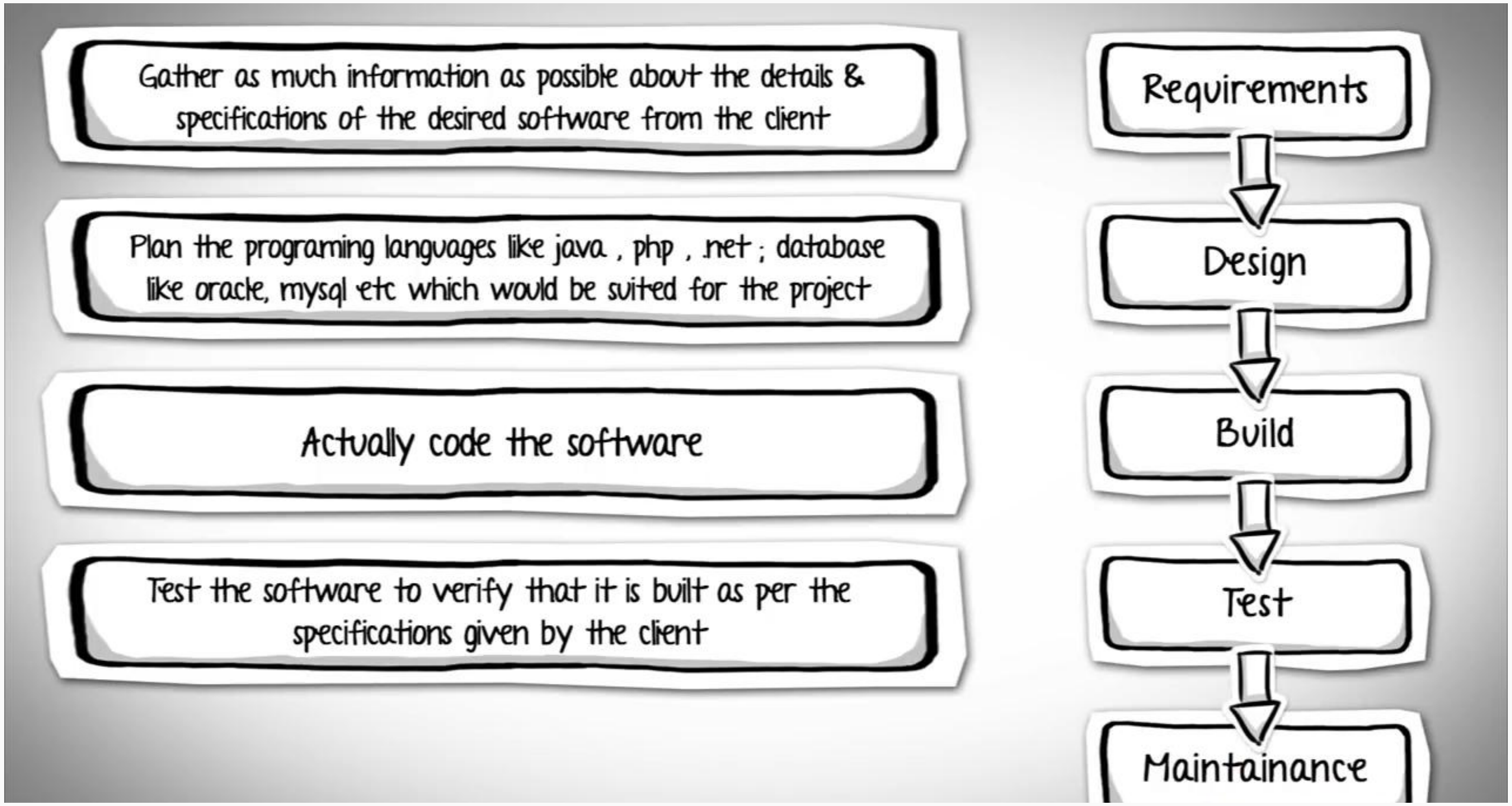


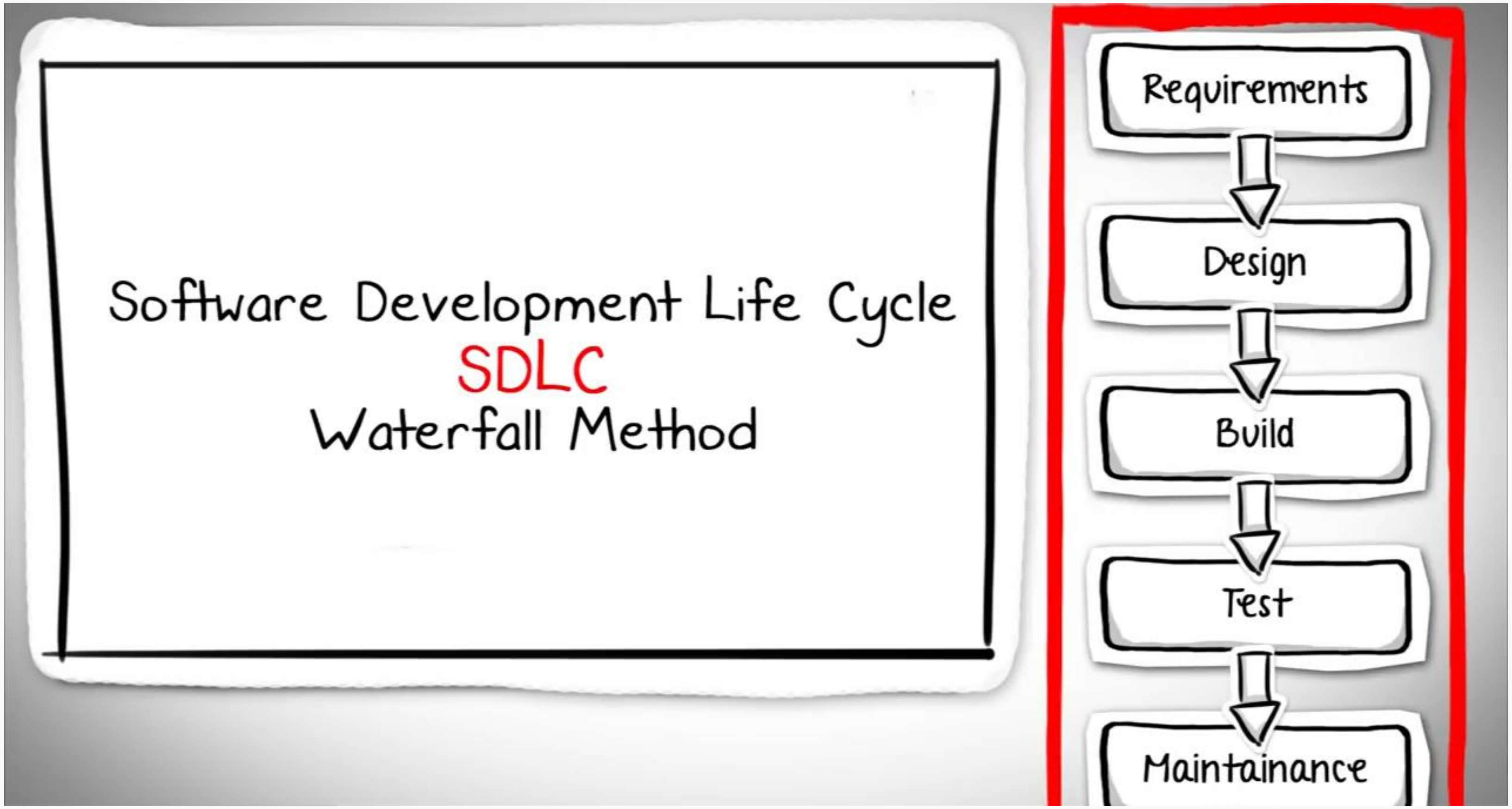
Plan the programing languages like java , php , .net ; database like oracle, mysql etc which would be suited for the project

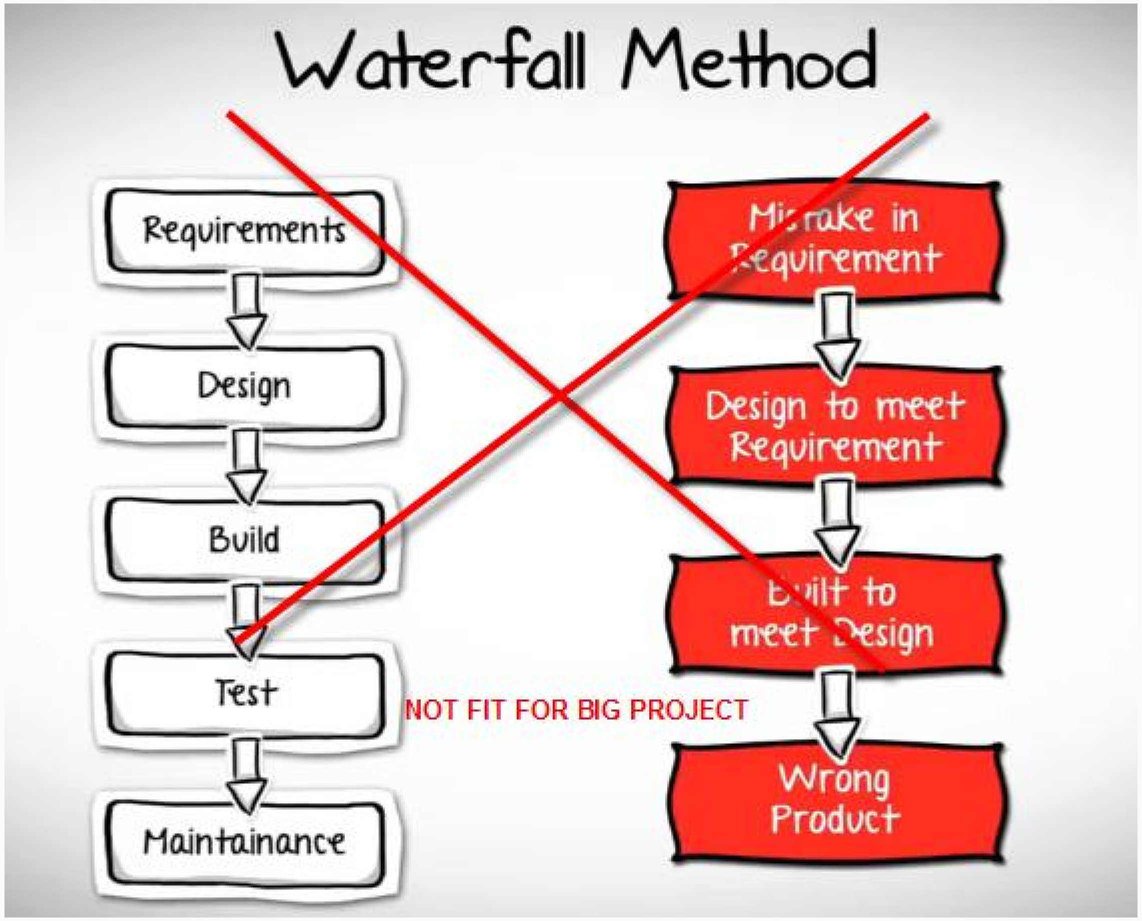
Test the software to verify that it is built as per the specifications given by the client

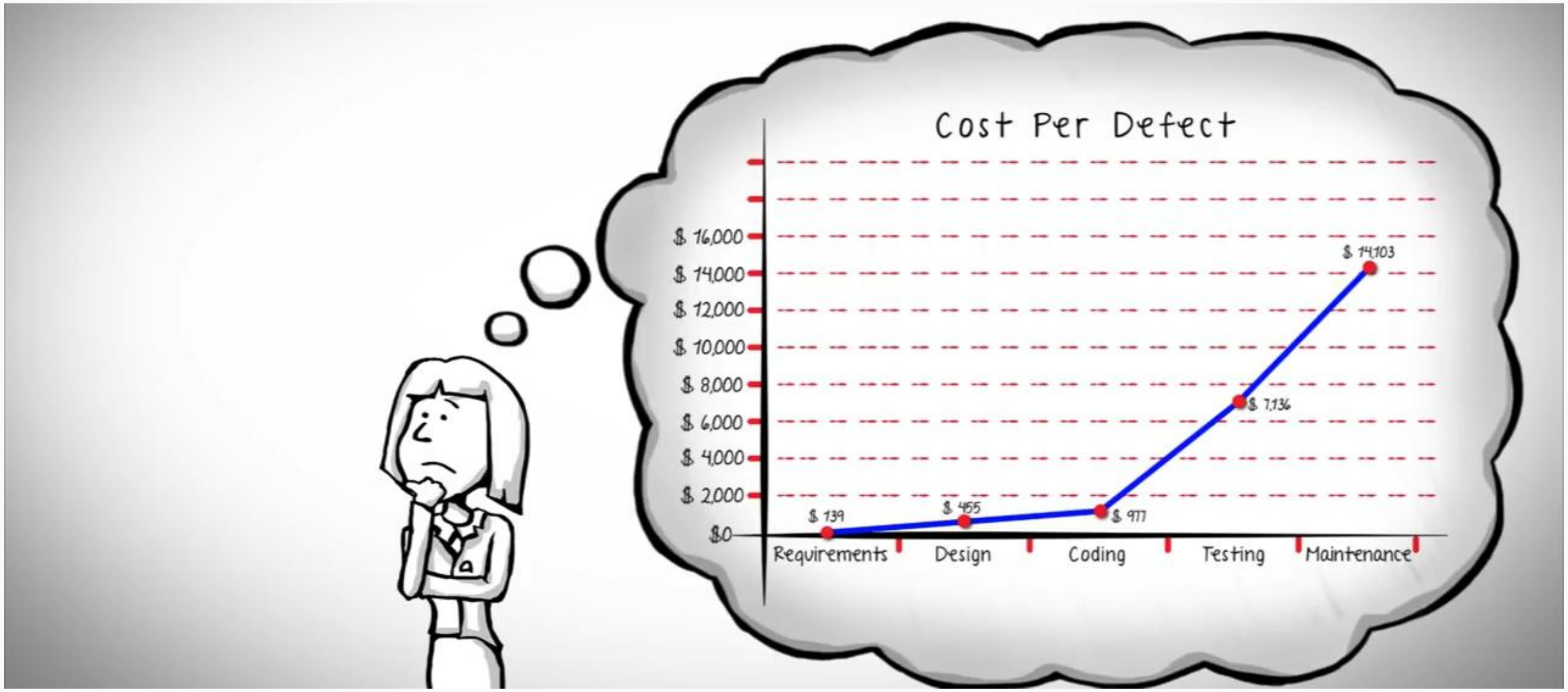
Actually code the software

Gather as much information as possible about the details & specifications of the desired software from the client

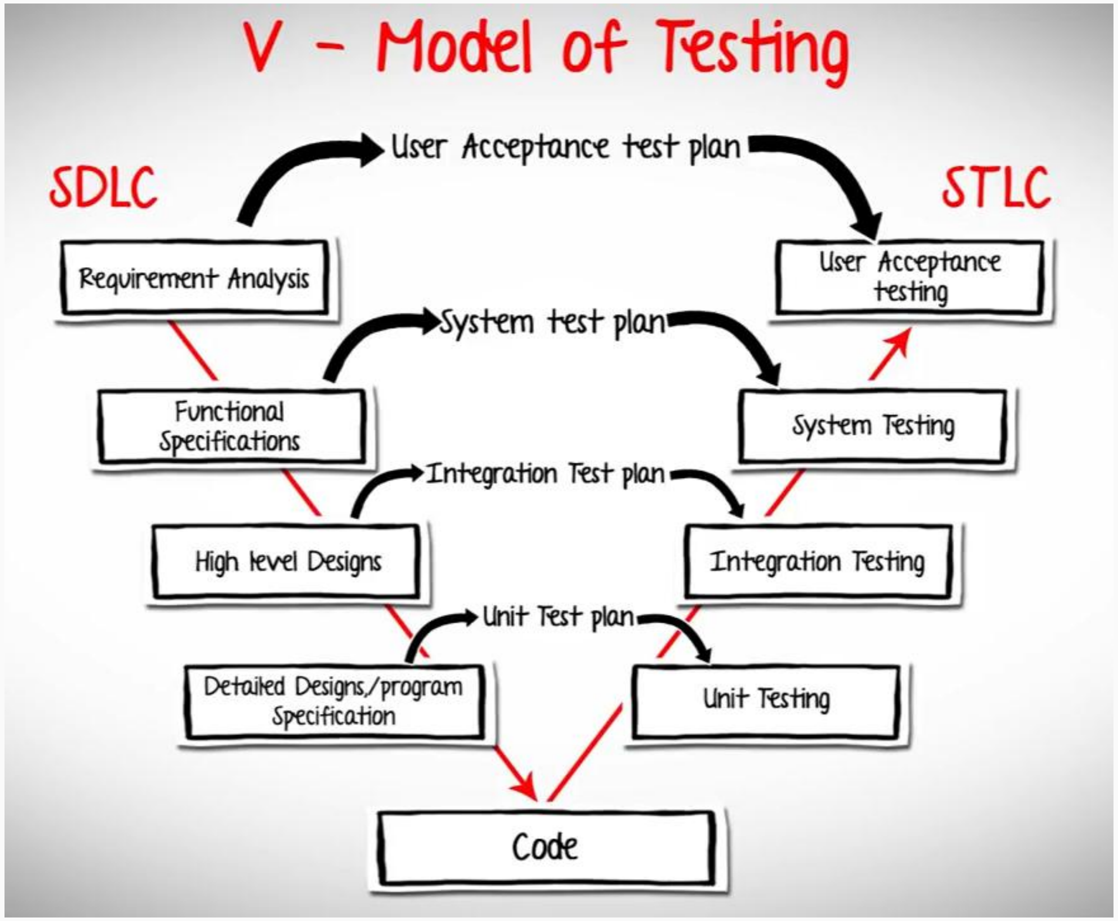

Gather as much information as possible about the details & specifications of the desired software from the client
Plan the programing languages like java , php , .net ; database like oracle, mysql etc which would be suited for the project
Actually code the software
Test the software to verify that it is built as per the specifications given by the client
Requirements
Design
Build
Test
Maintainance


Software Development Life Cycle
SDLC
Waterfall Method
Requirements
Design
Build
Test
Maintainance

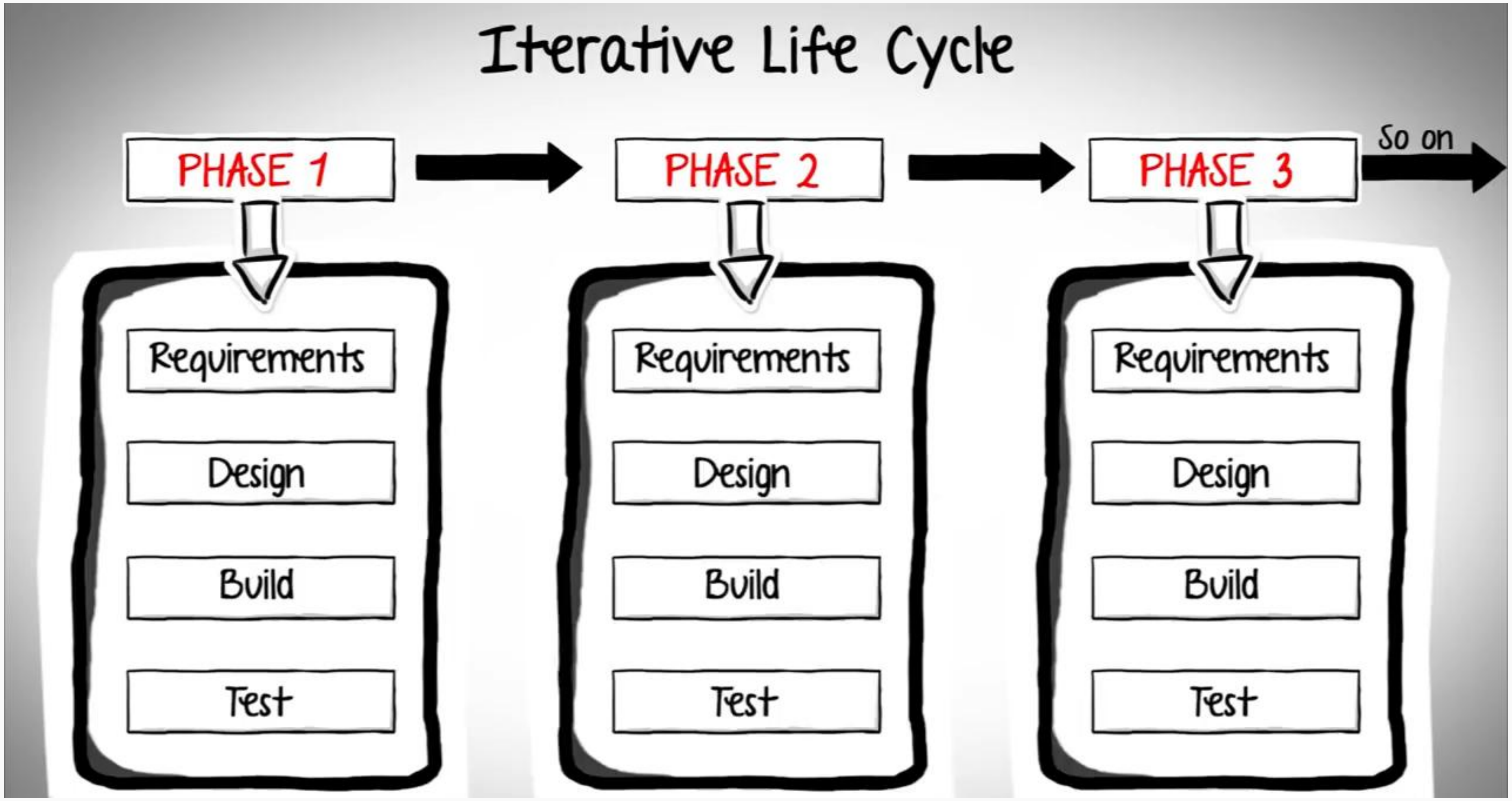

Waterfall Method
Requirements
Design
Build
Test
Maintainance
Mistake in Requirement
Design to meet Requirement
Built to meet Design
Wrong Product
NOT FIT FOR BIG PROJECT


Cost Per Defect
$ 16,000
$ 14,000
$ 12,000
$ 10,000
$ 8,000
$ 6,000
$ 4,000
$ 2,000
$0
$ 139
$ 455
$ 977
$ 7,136
$ 14,103
Requirements
Design
Coding
Testing
Maintenance

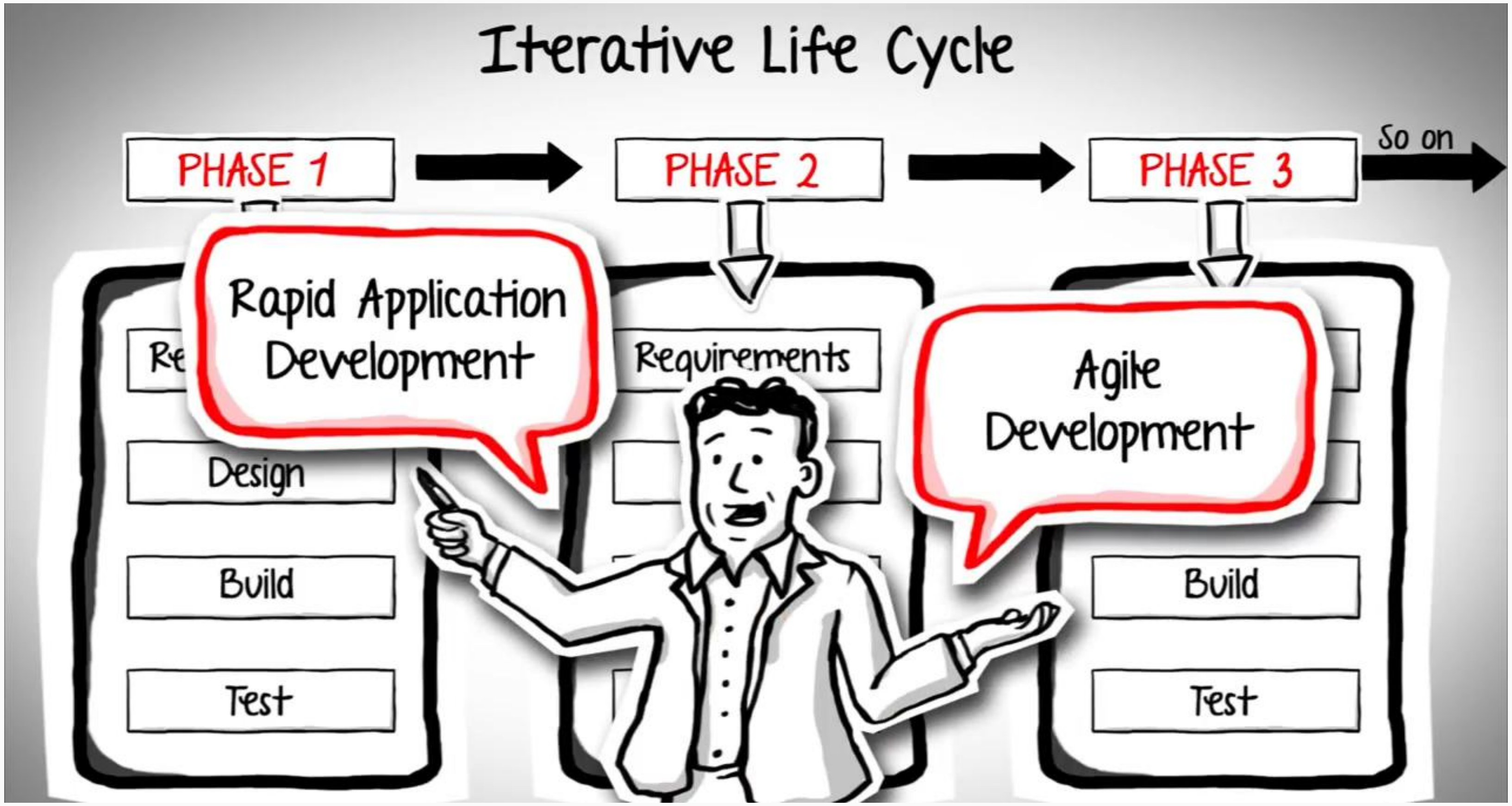
The V-Model


V - Model of Testing
SDLC
STLC
User Acceptance test plan
Requirement Analysis
User Acceptance testing
System test plan
Functional Specifications
System Testing
Integration Test plan
High level Designs
Integration Testing
Unit Test plan
Detailed Designs/program Specification
Unit Testing
Code

Another model



Another model



There are numerous development life cycle models. **Development model selected for a project depends on the aims and goals of that project.**

- Testing is not a stand-alone activity, and it has to adapt the development model chosen for the project.
- In any model, testing should be performed at all levels i.e. right from requirements until maintenance.

Unit Testing

UNIT Testing is defined as a type of software testing where individual units/ components of a software are tested.

Unit Testing of software applications is done during the development (coding) of an application. The objective of Unit Testing is to isolate a section of code and verify its correctness. In procedural programming, a unit may be an individual function or procedure. Unit Testing is usually performed by the developer.

Unit Testing

Key reasons to perform unit testing :

1. Unit Tests fix bug early in development cycle and save costs.
2. It helps understand the developers the code base and enable them to make changes quickly
3. Good unit tests serve as project documentation
4. Unit tests help with code re-use. Migrate both your code and your tests to your new project. Tweak the code till the tests run again.

Unit Testing

Scenario

Your company is hired by a bank to develop an online banking application

Unit Testing

Requirement Analysis

1) Login on valid Credentials

2) View Current Balance

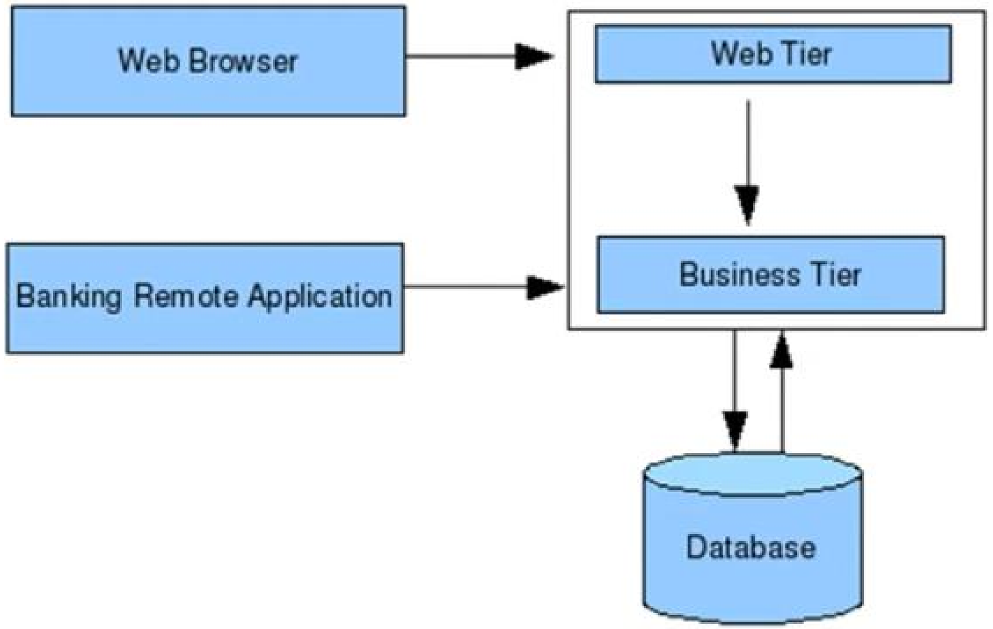
3) Deposit

4) Withdraw

5) Transfer

Unit Testing

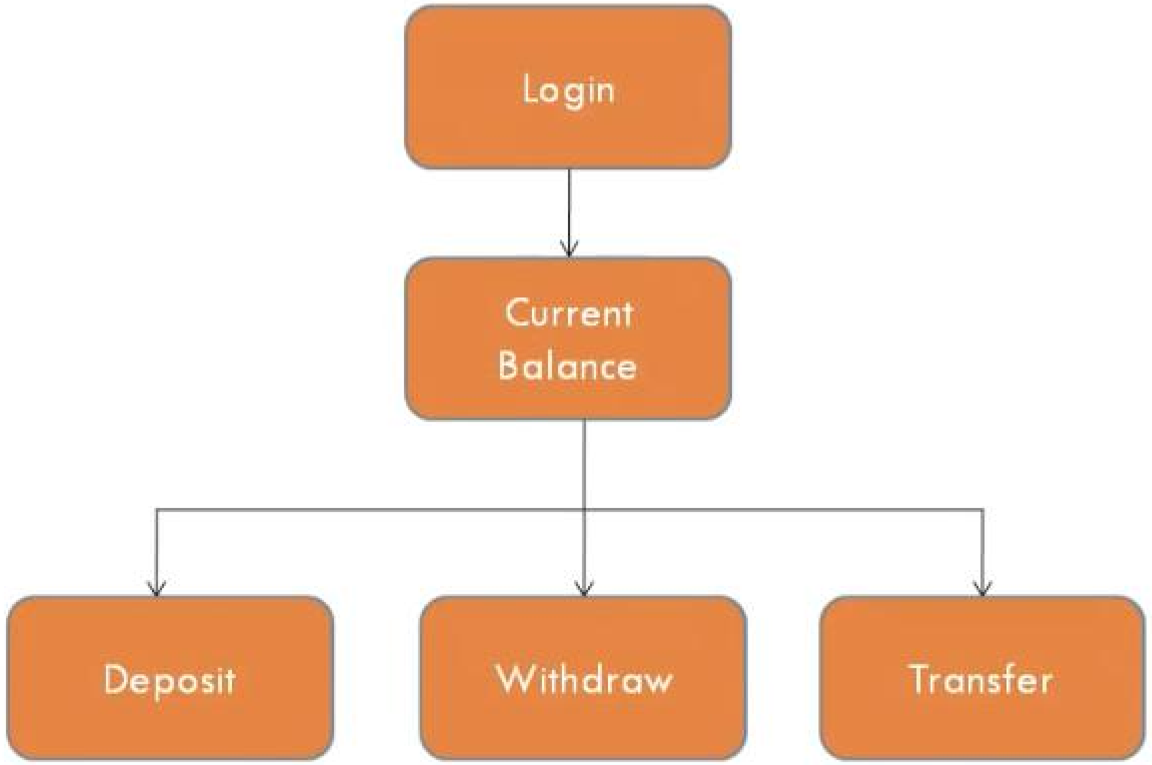
Functional Specification



Unit Testing

High Level

Design



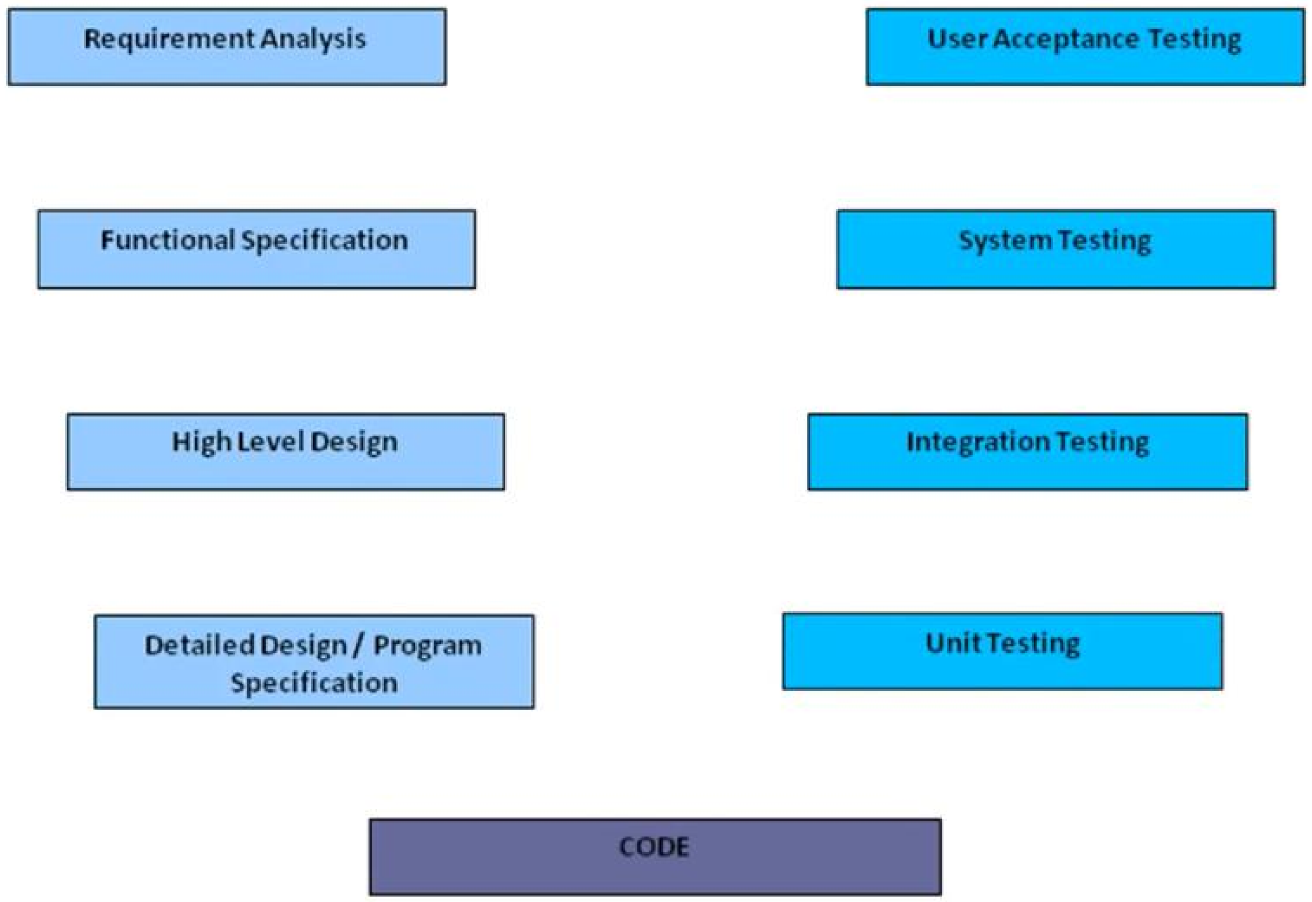
Unit Testing

Deposit

Detail Design

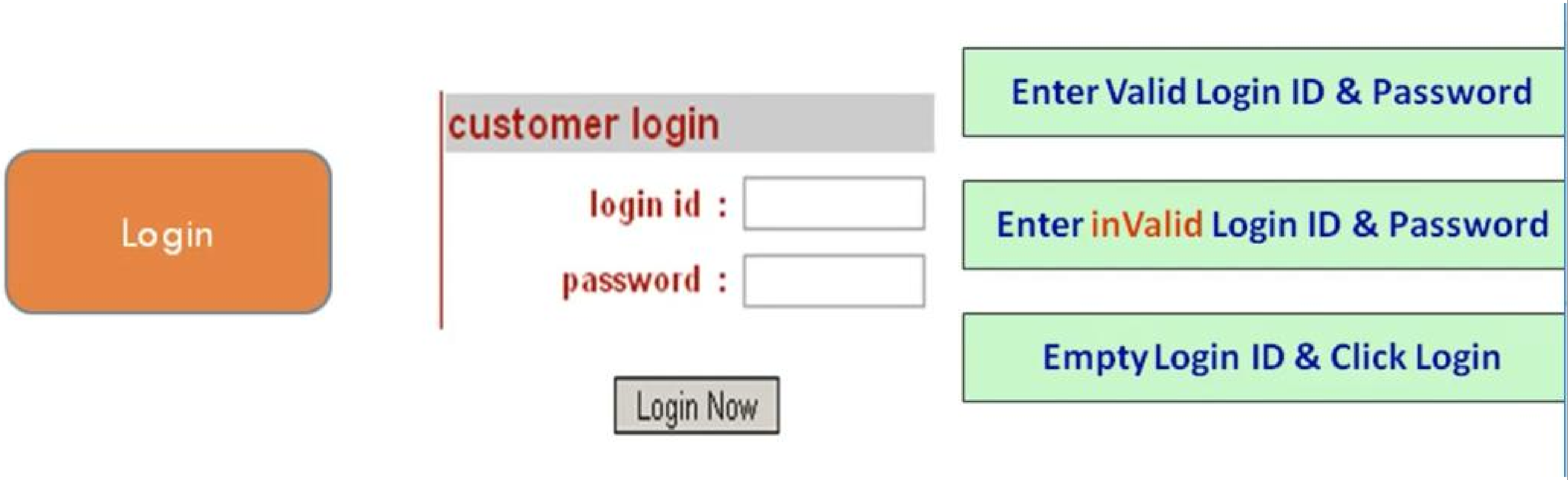
```
Function depositMoney(int amount) {

   // Only Declarations of the functions
   //No Actual Code
  //This helps in maintaining uniformity
  across the project and avoid errors

}
```

Unit Testing



Unit Testing

- It is also called component testing
- It is performed on standalone module to check whether it is developed correctly


Login
customer login
login id :
password :
Login Now
Enter Valid Login ID & Password
Enter inValid Login ID & Password
Empty Login ID & Click Login

- Unit testing is done by developers

Unit Testing (Tools)

- **JUnit**: JUnit is a free to use testing tool used for Java programming language. It provides assertions to identify test method. This tool test data first and then inserted in the piece of code.
- **NUnit**: NUnit is widely used unit-testing framework use for all .net languages. It is an open source tool which allows writing scripts manually. It supports data-driven tests which can run in parallel.
- **PHPUnit**: PHPUnit is a unit testing tool for PHP programmer. It takes small portions of code which is called units and test each of them separately. The tool also allows developers to use pre-define assertion methods to assert that a system behave in a certain manner.

Unit Testing (Advantages)

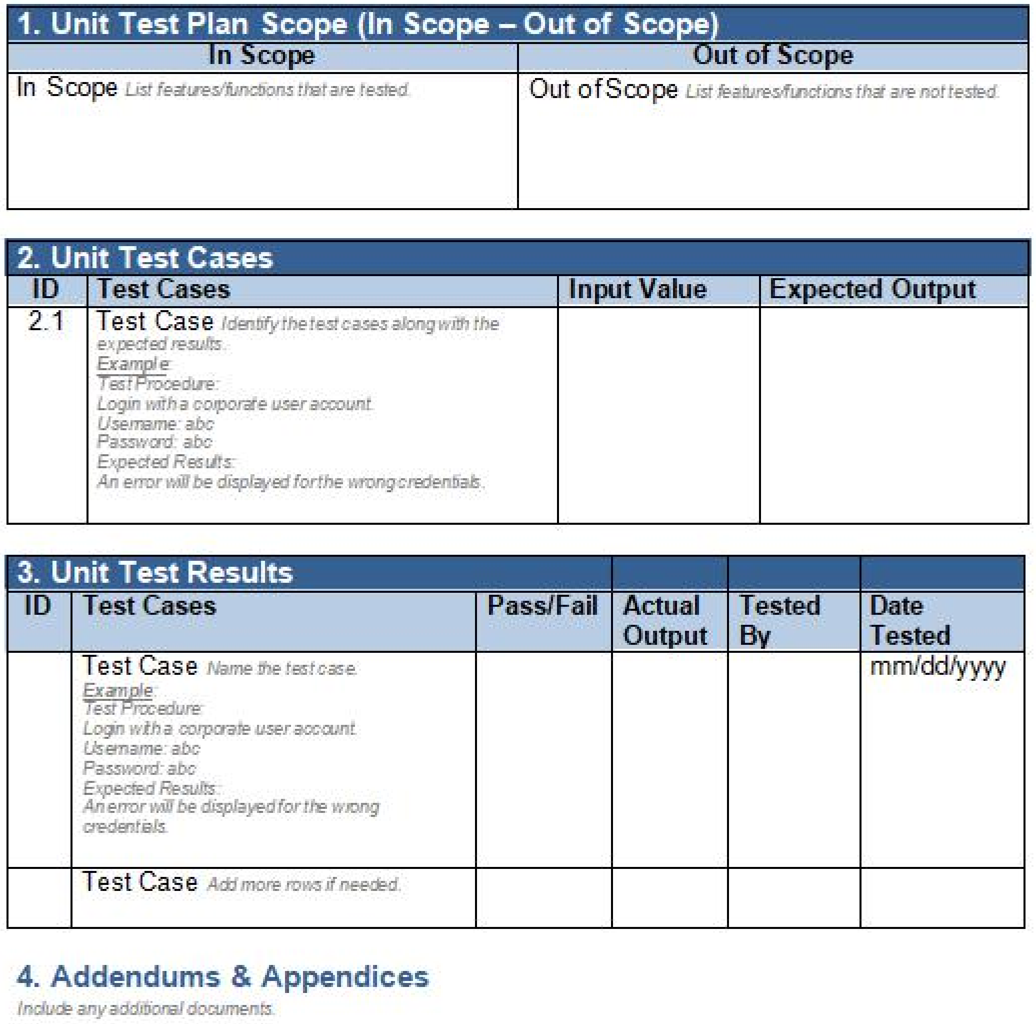
- Developers looking to learn what functionality is provided by a unit and how to use it can look at the unit tests to gain a basic understanding of the unit.
- Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly (i.e. Regression testing). The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.
- Due to the modular nature of the unit testing, we can test parts of the project without waiting for others to be completed.

Unit Testing (Disadvantages)

- Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs
- Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.

Unit Testing (Practices)

- Download Unit Test Plan Template
- Test the features and functions of your app
- Use JUnit, PHPUnit, py.test, NUnit, etc.
- Present your test plan document.



1. Unit Test Plan Scope (In Scope – Out of Scope)

In Scope	Out of Scope
In Scope *List features/functions that are tested.*	Out of Scope *List features/functions that are not tested.*

2. Unit Test Cases

ID	Test Cases	Input Value	Expected Output
2.1	Test Case *Identify the test cases along with the expected results.* *Example:* *Test Procedure:* *Login with a corporate user account.* *Username: abc* *Password: abc* *Expected Results:* *An error will be displayed for the wrong credentials.*		

3. Unit Test Results

ID	Test Cases	Pass/Fail	Actual Output	Tested By	Date Tested
	Test Case *Name the test case.* *Example:* *Test Procedure:* *Login with a corporate user account.* *Username: abc* *Password: abc* *Expected Results:* *An error will be displayed for the wrong credentials.*				mm/dd/yyyy
	Test Case *Add more rows if needed.*				

4. Addendums & Appendices

Include any additional documents.