
**Java Institute for Advanced Technology**

**UNIT NAME:** SOFTWARE ENGINEERING II

**UNIT ID:** HF2W 04

**ASSIGNMENT ID:** HF2W 04/AS/04

**NAME:** M.R.P.N.THARUKSHA RAJAPAKSHA

**STUDENT ID:** 2019/2020/CO/SE/I2/029

**SCN NO:** 207977608

**NIC:** 200019401866

**BRANCH:** JAVA INSTITUTE, COLOMBO

1. What is the best testing model?

- V Model

2. What's the difference between alpha and beta testing?

| Alpha Texting | Beta Testing |
|---|---|
| Performed by the employees of the organization. | Done by clients who are not part of the organization. |
| Requires a specific environment for testing. | Does not require any environment for testing. |
| Robustness and security test is not performed. | Robustness and security parameters are checked. |
| Performed before the product launches into the market. | Performed at the time of product marketing. |
| Performs many cycles to complete the testing. This may vary with the number of issues found. | Performs 1-2 cycles to complete the testing. This may vary with the user's feedback. |
| The main goal is to evaluate the quality of the product. | The main goal is to evaluate customer satisfaction. |
| Both white-box & Black-box testing is involved. | Only involves black-box testing. |
| Activities can be controlled since it's performed on the developer's site. | Activities can't be controlled, since it's performed in the real environment. |
| Done by highly-skilled employees. They have knowledge about the software product. | Done by the end-users. They don't have the technical knowledge of the software product. |
| Stakeholders are the product management team, quality assurance team, and engineers. | Stakeholders are the product management team, user experience team, and quality management team. |
| Developers can resolve the bugs in alpha testing after testers inform them. | The feedback collected from the users is implemented in the future or in the next version of the application. |

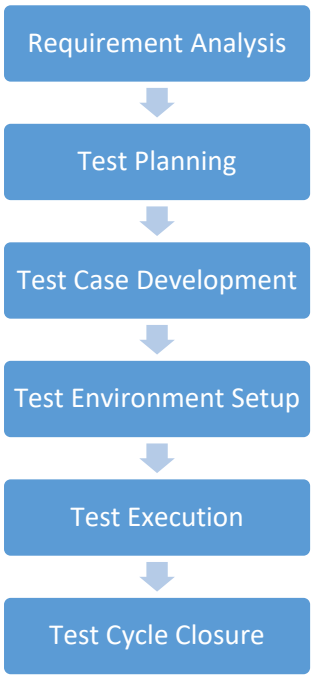
3. What's the difference between System testing and Acceptance testing?

| System testing | Acceptance testing |
|---|---|
| Performed to test end-to-end functionality of the software. | Performed to test whether the software is conforming to specified requirements and user requirements or not. |
| Only developers and testers can perform. | Testers, stakeholders, and customers can perform. |
| Uses demo input values that are selected by the testing team. | Uses the actual real-time input values provided by the user. |
| Test the performance of the whole system. | Test whether the system is conforming to requirements or not. |
| Can be both non-functional and functional testing. | Can be only functional testing. |
| Include the testing of complete specifications including software and hardware, memory and number of users. | Test whether the software is fulfilling all the needs of the user or not. |
| Performed before the Acceptance testing. | Performed after the System testing. |
| Involves load and stress testing under non-functional testing. | Involves boundary value analysis, equivalence portioning, and decision table under functional testing. |
| Defects found in system testing are considered to be fixed. | Defects found in acceptance testing are considered product failures. |
| Combination of System Testing and Integration testing. | Combination of alpha testing and beta testing. |

4. What are the advantage of unit testing?

- Reduces bugs in newly developed features or modifies existing functionality.
- Errors are captured at an early stage, thus reducing test costs (Reduce costs).
- Allows design improvements and code to be better redesigned.
- Unit tests also provide build quality when integrated with construction.
- Makes the process agile.
- Quality of code.
- Finds software bugs early.
- Facilitates changes and simplifies integration.
- Provides documentation.
- Debugging process.

5. What is Software Testing Life Cycle (STLC) and if phases explain them?




The Software Testing Life Cycle (STLC) is a testing process that has specific steps that need to be implemented in a specific sequence to ensure that quality goals are met. In the STLC process, every activity is done in a planned and systematic manner. Each stage has different goals and distributions. Different institutions in STLC have different stages; however, the basis remains the same.

- Requirement Analysis
  Needs analysis is one of the most important stages because you can fix project errors almost completely free of charge. The requirements analysis phase identifies potential requirements for automated testing and allows for economic calculations of labor costs based on the project estimate. This is also the case when the entry criteria and exit criteria are discussed and documented.

- Test Planning
  At this stage, a test plan is developed. This combines all the stages, timing, participants, and responsibilities of the test. As a result, we get data: Participants and their roles in testing, required test tools, required test environment.

- Test Case Development
  Test development means the use of manual and automated tests to obtain full coverage of the functionality of the software, based on pre-established requirements of the process. Often, test cases for automated tests are written separately, as opportunities for manual testing are described in the form of fraudulent forms.

- Test Environment Setup
  The test plan specifies which test environment should be used. During this STLC phase, operating systems and virtual machines are configured, testing tools such as Selenium, Katalon Studio, and the test environment and database of the project are deployed. We also make requests to DevOps and Admins if assistance is required.

- Test Execution
  Tests are performed based on prepared test documents and a properly configured test environment. All test results are recorded in the test management system. Negatively passed tests, when the actual result differs from the expected result, are noted as errors and transferred to the development team for revision with re-testing after correction.

- Test Cycle Closure
  The final stage of STLC is the final generation of test reports for the client. These should include the time taken, the percentage of errors found for positive test results, and the total number of errors detected and corrected. In the case of the testing department, this is an opportunity to analyze its work, summarize the results, analyze its effectiveness and make suggestions for improving the quality of the tests.

6. What are the type of test model are available?

- Waterfall Model
- V Model
- Spiral Model
- Iterative Model
- Agile Model
- Rapid Application Development

7. What is the testing level and write down the various test levels available?

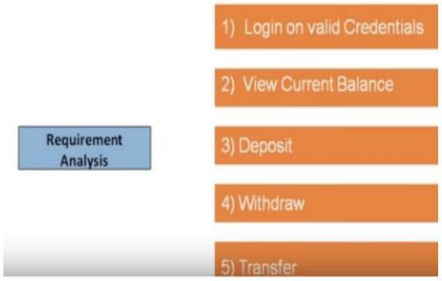
Test levels are the procedure for locating missing areas and preventing overlap and recurrence between developmental life cycle stages. We have already seen the various stages of SDLC (Software Development Lifecycle) requirements addition, design, coding testing, deployment, and maintenance.

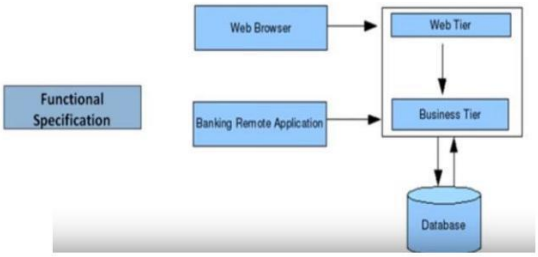
To test any application, we have to go through all the above stages in SDLC. Like SDLC, we have different levels of testing that help us maintain the quality of the software.

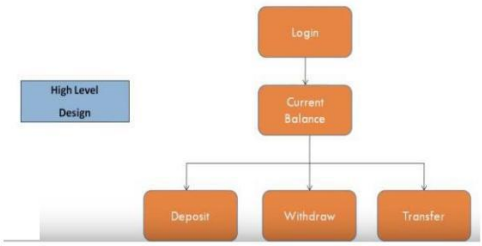
Various test levels:

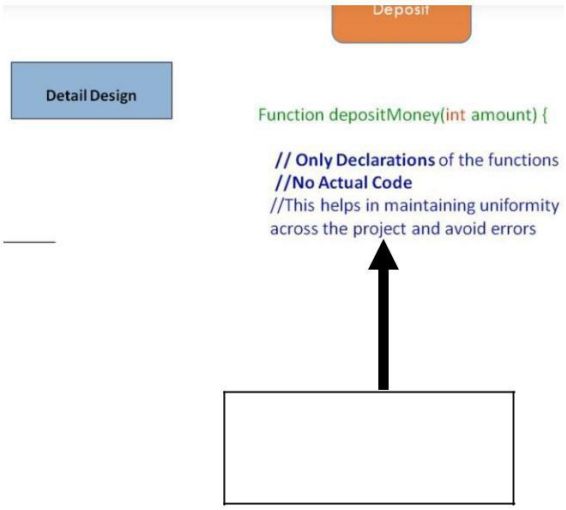
1. Unit/Component Testing - Checks whether software components meet functionality.
2. Integration Testing - Checks the data flow from one module to the other.
3. System Testing - Evaluate both functional and non-functional requirements for testing.
4. Acceptance Testing - Specification requirements of a specification or a contract are met according to its distribution.

8. You are haring to develop a banking application. Bellow mention the Functionality of a system. (SRS- requirement and functions)





Explain how each tasting phases take place according to your banking application.

- Login on Valid Credentials
    - Unit Testing
        - ✓ Check entered username string received correctly.
        - ✓ Check entered password string Received correctly.
        - ✓ Check database connection code.
        - ✓ Check if the code returns true when the username exists in the database.
        - ✓ Check passwords can be correctly extracted from the database.
    - Integration Testing
        - ✓ Check if the code correctly verifies the username and password entered by the user with the username and password extracted from the database.
    - System Testing
        - ✓ Enter the correct username and password to check whether the dashboard opens up.

        - ✓ Incorrect username and password to check whether software gives an error message.

    - User Acceptance Testing
        - ✓ Let end-user insert their username and password to check whether the user can log into the account successfully.

- View Current Balance

    - Unit Testing
        - ✓ Check database connection.
        - ✓ Check user ID is correctly received to pass to the database query to find balance.
        - ✓ Check select balance database query.
        - ✓ Check whether the value is received after executing the database query.
        - ✓ Check whether the value is a double value.

    - Integration Testing
        - ✓ Check getBalance() method parse user ID as a parameter and print the return value to check whether the correct balance is received.

    - System Testing
        - ✓ After login into the system check the balance on the screen to test whether the correct balance amount is shown on the screen as expected.

    - User Acceptance Testing
        - ✓ Client logs into their banking application and views their account balance.

- Deposit

    - Unit Testing
        - ✓ Get the deposit account number and check whether the value is received correctly.
        - ✓ Check whether the money counting device returns the correct amount.
        - ✓ Check whether the depositing amount is correctly received as a parameter for the method deposit().
        - ✓ Check database query to update account balance.
        - ✓ Execute database query to find whether it is executed correctly and the database is updated.
        - ✓ Check whether the method is written in the success value after successfully completing the deposit.

    - Integration Testing
        - ✓ Connect counting device to software and check whether the correct amount is received by the deposit method.
        - ✓ Check whether the receipt is correctly printed by the software after deposit.

    - System Testing

- ✓ Enter sample account number add notes to the counter and click deposit to find out whether the amount is correctly deposited into the bank account.
- ✓ Enter a larger amount that available balance to find if the error message is displayed.

- User Acceptance Testing
  - ✓ Customers use the deposit machine to deposit cash into their account by entering their account number placing money inside the money counter and clicking the deposit button.

- Withdraw

  - Unit Testing
    - ✓ Check withdrawal amount is correctly received by the withdraw() method by printing it out to the Console.
    - ✓ Check whether the account number is received to deduct the amount from.
    - ✓ Check database query for selecting existing account balance.
    - ✓ Check whether the update query is only executed when the existing account balance is equal to or higher than the amount to be withdrawn.
    - ✓ Check database query for updating account balance.
    - ✓ Execute database query and find whether it is executed correctly and the new balance is updated on the database.

  - Integration Testing
    - ✓ Check the whole withdrawal method by passing the account number and withdrawal amount to the method to find out the database is correctly updated.

  - System Testing
    - ✓ After login into the system enter the sample account number and withdrawal amount to find out it can be correctly withdrawn.

  - User Acceptance Testing
    - ✓ User login to the system and type withdrawal amount if the withdrawal amount is larger than the account balance error message should pop up. if not it can be successfully withdrawn.

- Transfer

  - Unit Testing
    - ✓ Check whether the bank account number to be transferred is correctly received to the method.
    - ✓ Check whether transferring amount is correctly received into the method.
    - ✓ Check database query for deducting the amount from a current bank account and updating the balance on the transferred bank account.
    - ✓ If the transfer account is not an account on this Bank check whether other bank API redirects are correctly working.

  - Integration Testing

- ✓ Check after transferring account balance is correctly deducted.
- ✓ Check whether communications with other bank APIs are working correctly.

- System Testing
  - ✓ Login to test account and transfer money to another test account to check transfer is working correctly.
  - ✓ Try to transfer a higher amount than the available account balance to check whether an error message will pop up.

- User Acceptance Testing
  - ✓ The user logs into the account and enters the account number to transfer funds and enter the amount to be transferred and click transfer and check whether the transfer is successful.

9. What is State Transition Testing? And when it use?

State Transition Testing is a black box testing technique in which changes made in input conditions cause state changes or output changes in the Application under Test (AUT). State transition testing helps to analyze the behavior of an application for different input conditions. Testers can provide positive and negative input test values and record the system's behavior. It is the model on which the system and the tests are based. Any system where you get a different output for the same input, depending on what has happened before, is a finite state system.

When it can use:

- When a tester is testing the application for a finite set of input values.
- When the tester is trying to test the sequence of events that occur in the application under test.
- When the system under test has a dependency on the events/values in the past.

10. What is the test report and explain included things?

Test Report is a document that contains a summary of all test activities and the final test results of a testing project. The test report is an assessment of how well the Testing is performed. Based on the test report, stakeholders can evaluate the quality of the tested product and make a decision on the software release.

Included things:

- Project Information
  All project information, such as the project name, product name, and version, should be described in the test report.

- Test Objective
  As stated in the test planning tutorial, the test report should include the objective of each test round, such as unit testing, performance testing, system testing, and etc.

- Test Summary
  This section includes a summary of testing activity in general. The information detailed here includes such as, the number of test cases executed, the number of test cases that pass, the number of test cases that fail, pass percentage, fail percentage, comments. This information should be displayed visually by using the color indicator, graph, and highlighted table.

- Defect
  One of the most important pieces of information in the Test Report is defect. The report should contain the following information total number of bugs, the status of bugs (open, closed, responding), number of bugs open, resolved, closed, breakdown by severity and priority, like test summary, you can include some simple metrics like Defect density, % of fixed defects.