**Testing**

Process of checking the functionality of an application to ensure that it runs a per the requirements

RTM – Requirement Traceability Matrix

* **Test plan**
* **Test strategy**
* **Test cases**
* **Test scenarios**
* **Test script**
* **Test data**

**Various type of testing**

1. **Unit testing –** refers to the testing of small segments of code.
2. **Dev/Integration Testing**
3. **UAT User acceptance Testing**
4. **Pre-production Testing**
5. **Smoke Testing**
6. **Regression**
7. **Performance**
8. **Black box –** examines the functionality of software without knowing the internal structure or code. Its primary source is the customer requirement specification.
9. **White box –** checking of predefined inputs against expected & desired outputs. Knows the internal structure & code.

Unit testing 🡪 2 ways Manual testing & Automated testing

* Single component of software is tested.
* Pick each unit (code) and verify it.

**JUnit**

* unit testing framework for java programming lang.
* used by java developers to write and execute tests.
* Concept-First testing then coding.
* Every time a new code is added, all the test cases have to be re-executed.
* This approach is like "test a little, code a little, test a little, code a little."
* Increases the productivity of the programmer
* Increases the stability of program code, which in turn reduces the stress on the programmer and the time spent on debugging.

**Features of JUnit**

* Provides annotations to identify test methods.
* Provides assertions for testing expected results.
* Provides test runners for running tests.
* JUnit tests can be organized into test suites containing test cases and even other test suites. JUnit shows test progress in a bar that is green if the test is running smoothly, and it turns red when a test fails.

**Testing Concepts**

* Quality Center: it is a tool

**Waterfall**

* In this nxt phase starts only after the completion of earlier phase.
* Here while build team is working,

**Agile**

* In the form of Sprints…each sprint spans upto around 2 weeks.
* Scrum Master(may be 1 or 2)🡪 Scrum Meeting/Stand meeting
* The no. of requirements that can fit in a 2 weeks span.

**RTM (Requirement Traceabilty Matrix)**

* This is a table format which will have the mapping between the requirements and Testcases.
* R1 TC1 Link to the testcase doc 🡪 XLS sheet
* RTM is been created by your Testing engineer/ test lead to have a track and mapping between all the requirements and the testcases – status.

**Test strategy**

* Defines how the testing should be carried out!

**Test plan**

* Detailed document that defines the scope, objective etc.
* List of all activities
* Schedule, estimates, deliverables and test data etc

**Test Scenario**

* Given a requirement/use case there can be many possibilities

When the test scenario broken up 🡪 test case

**Test case**

* Emphasizes one particular flow/ has the steps to execute one single flow. This is written in a xls file.

**STLC** (Software testing life cycle)

* Software testing life cycle
* Requirement is confirmed🡪 Test planning 🡪 RTM 🡪 Test Strategy 🡪 Test plan 🡪 Test Scenarios 🡪 Test cases/Test scrips 🡪 Test Execution 🡪 Go/ no Go (defect fixing by developers)

**Bug and a defect**

* Bug 🡪 mistake in the program created by programmer. This can be caught at unIt/ dev -integration phases.
* Defect 🡪 this is a deviation in the requirement. Program is not behaving as per the requirement.

QA (Quality Analyst)

* May be one person act as a QA
* He is also the part of testing team
* Defines the scope

Scope has to be very much defined

RTM

Automation scope has to be defined

STLC Phases:

Acceptance criteria: excepted behaviour of the system as per the requirement.

Epic🡪 piece of user story

(Not in paragraph)

* Entry and exit criterias:
* Defects can become the new requirement
* Any new requirement can be called as CR (changed requirement)

Levels of Testing:

Levels of testing include the different methodologies that can be used while conducting Software Testing. It includes,

* Functional Testing
* Non- functional Testing

**Functional Testing**

This is a type of black box testing that is based on the specifications of the software that is to be tested.

* **Unit Testing**
* This type of testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases.
* Unit testing is performed by the respective developers on the *individual units* of source code assigned areas.
* The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.
* Limitations of Unit Testing- Testing cannot catch each and every bug in an application, it is impossible to evaluate every execution path in every software application, There is a limit to the number of scenarios and test data that the developer can use to verify the source code.
* **Integration Testing**
* The testing of *combined parts* of an application to determine if they function correctly together is Integration testing.
* There are two methods of doing Integration Testing Bottom-up Integration testing and Top Down Integration testing.
* **System Testing**
* Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.
* This type of testing is performed by a specialized testing team.
* **Regression Testing**
* Whenever a change in a software application is made it is quite possible that other areas within the application have been affected by this change.
* To verify that a fixed bug hasn’t resulted in another functionality or business rule violation is Regression testing.
* The intent of Regression testing is to ensure that a change, such as a bug fix did not result in another fault being uncovered in the application.
* **Acceptance Testing**
* Check whether the application meets the intended specifications and satisfies the client’s requirements.
* The QA team will have a set of pre written scenarios and Test Cases that will be used to test the application.
* 2 stages-Alpha & Beeta testing.

**Non Functional Testing**

Non-functional testing of Software involves testing the Software from the requirements which are non-functional in nature related but important a well such as performance, security, user interface etc.

* **Performance Testing**
* It is mostly used to identify any performance issues rather than finding the bugs in software.
* Performance testing is considered as one of the important and mandatory testing type in terms of some aspects such as Speed (i.e. Response Time, data rendering and accessing), Capacity, Stability and Scalability.
* 2 types- Load testing and Stress testing.
* **Load testing-** A process of testing the behaviour of the Software by applying maximum load in terms of Software accessing and manipulating large input data. This type of testing identifies the maximum capacity of Software and its behaviour at peak time.
* **Stress testing-** This testing type includes the testing of Software behaviour under abnormal conditions. Taking away the resources, applying load beyond the actual load limit. The main intent of this testing is to identify the breaking point of software.
* **Static testing** : (done without executing the application)

## Objectives of Static testing

* Decrease the flaws in production.
* Identify, anticipate and fix the bugs at the earliest possible time.
* Used to save both time and cost.
* Used to identify defects in the early stage of SDLC, where we can fix them easily.

Basically through these dummy features

1. Review 🡪 to crosscheck the documents across the requirements

* Process to find the possible bugs
* all the team members will understand about the project's progress.
* **Different parts in review**

1. *Informal reviews*

the document designer place the contents in front of viewers, and everyone gives their view

1. *Walkthroughs*

performed by a skilled person or expert to verify the bugs. Therefore, there might not be problem in the development or testing phase.

1. *Technical/peer review*

we can check one another's documents to find and resolve the bugs, which is generally done in a team.

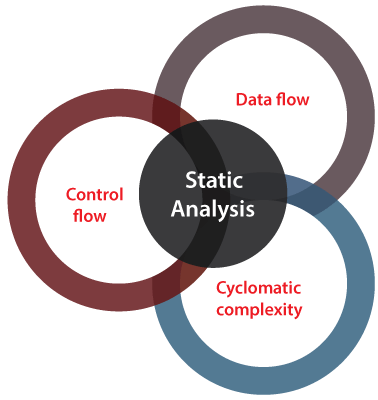
1. *Inspections*

verifying the document by the higher authority, **for example**, the **verification of SRS [software requirement specifications] document.**

1. Static analysis 🡪 to ensure if there is any kind of breakage in application.

* used to contain the assessment of the code quality, which is established by developers.
* help us to identify the below errors: **Dead code**, **Unused variables**, **Endless loops**, **Incorrect syntax**, **Variable with undefined value**
* **Static Analysis** can be further classified into **three parts**

1. **Data flow**
2. **Control flow**
3. **Cyclomatic complexity**



**Data Flow**: In static analysis, the data flow is connected to the stream processing.

**Control Flow**: Generally, the control flow is used to specify how the commands or instructions are implemented.

**Cyclomatic Complexity**: It is the measurement of the program's complexity, which is mostly linked to the number of independent paths in the control flow graph of the program.

Test management, Risk analysis, QA/QE, Defects life cycle

Using a tool u need to manage the testcases and defects

* Risk Analysis
* Defects life cycle

New🡪 assign🡪in progress🡪 Test 🡪 re-test(to QA) 🡪 Closed/re-opened

**QA – Quality assurance team / quality analyst**

* Responsible for the quality of the products
* SQA
* QA -> testing role

**QE: quality engineer**

* He puts in the stringent rules to what tools to be used and policing on the quality regulations so that the quality is assured.
* One who automates quality procedures to minimize manual testing efforts

**Test Pyramid**

Some model or a structure that should be followed to have a quality product which is made faster and more efficient.

Integration/Component/Modules testing.

[Learn Software Testing Tutorial - javatpoint](https://www.javatpoint.com/software-testing-tutorial)

**V&V Verification & validation testing**

**Verification testing**

* includes different activities such as business requirements, system requirements, design review, and code walkthrough while developing a product.
* We check whether we are developing the right product or not

**Validation testing**

* This is testing where tester performed functional and non-functional testing. Here **functional testing** includes [Unit Testing](https://www.javatpoint.com/unit-testing) (UT), [Integration Testing](https://www.javatpoint.com/integration-testing) (IT) and System Testing (ST), and **non-functional** testing includes User acceptance testing (UAT).
* We check whether the developed product is right.
* User stories.
* FRD: Functional Requirement Document
* BRD: Business Requirement Document
* HLD: High level Design Document
* LLD: Low Level Design Document

Software Development Life Cycle (SDLC)

* SDLC describes various phases of software development and the order of execution of phases.
* Each phase requires deliverable from the previous phase in a life cycle of software development.
* Requirements are translated into design, design into development and development into testing; after testing, it is given to the client.

Software Development Life Cycle

[Requirement Phase](https://www.javatpoint.com/software-development-life-cycle" \l "requirement-phase)

* During this phase, the client states requirements, specifications, expectations, and any other special requirement related to the product or software..
* All these are gathered by the business manager or project manager or analyst of the service providing company

[Design Phase](https://www.javatpoint.com/software-development-life-cycle" \l "design-phase)

* [the logical designing of the system is converted into physical designing.](https://www.javatpoint.com/software-development-life-cycle" \l "design-phase)
* The output of the requirement phase is a collection of things that are required, and the design phase gives the way to accomplish these requirements.

Build /Development Phase

* the next step is to implement the design into the development of a software system.
* In this phase, work is divided into small units, and coding starts by the team of developers according to the design discussed in the previous phase and according to the requirements of the client discussed in requirement phase to produce the desired result.

[Testing Phase](https://www.javatpoint.com/software-development-life-cycle#testing-phase)

* last step of completing a software system.
* In this phase, after getting the developed GUI and back-end combination, it is tested against the requirements stated in the requirement phase.

Deployment/ Deliver Phase

* When software testing is completed with a satisfying result, and there are no remaining issues, it is delivered to the customer.
* Besides this, if any type of defect is encountered while a customer using the software; it will be informed to the development team of that particular software to sort out the problem.
* After the solution of all types of bugs and changes, the software finally deployed to the end-user.

[Maintenance](https://www.javatpoint.com/software-development-life-cycle#maintenance)

* The maintenance phase is the last and long-lasting phase of SDLC because it is the process which continues until the software's life cycle comes to an end.

various software development models or methodologies:

* **Waterfall model 🡪** It is the first sequential-linear model because the output of the one stage is the input of the next stage.
* **Spiral model**
* **Verification and validation model**
* **Prototype model**
* **Hybrid model**

**Requirement testing**

Based on the requirement provided by the client. All my test cases, test scenarios, test data are inclined from requirements.

Functional(based on the req.) & non functional (performance, system hardware etc) testing

**Types of requirement testing**

Implicit requirements 🡪 **The Things Your Customers Will Expect**

Explicit requirements 🡪 **The Things You Wrote Down**

Latent requirement 🡪 **Things That Will Delight Your Customers**

**Comparisons**

|  |  |
| --- | --- |
| **Black box testing** | **White box testing** |
| Testing is done without the knowledge of the internal structure of program or application | Testing is done with the knowledge of internal structure of program |
| Main goal- test the behaviour of software | Main goal- test the internal operation of software |
| Focused on external or end user perspective | Focused on code, structure, condition paths & branches |
| Less time consumption | More time consumption |

|  |  |
| --- | --- |
| **ALPHA TESTING** | **BEETA TESTING** |
| Test each user journey and confirm they work as intended- It always done by developers or tester at the software development site. | Test how the software perform in the real world scenario-It always performed by the customers at their own site |
| Done before the software release where it will address & fix bugs and correct any minor issue | User provide feedback & result to the product team for in co-operate in the next version |

|  |  |
| --- | --- |
| **SMOKE TESTING** | **SANITY TESTING** |
| Goal-To verify stability | Goal-To verify rationality |
| Done by both developers &Testers | Done by Testers |
| Verifies the critical functionalities of the system | Verifies the new functionalities like bug, fixes |
| Subset of acceptance testing | Subset of regression testing |
| Verifies the entire system from end to end | Verifies only a particular component. |

|  |  |
| --- | --- |
| **TEST CASE** | **TEST SCENARIO** |
| Test cases have a single step | Test scenarios has several steps |
| It is a detailed document consisting of application requirements, preconditions,  Test data, post conditions and expected results | It is a detailed test procedure consisting of test cases which helps to find problems in the system and evaluating the results |

Difference between Testing and Debugging:

Testing:

* + - It involves the identification of bug/error/defect in the software without correcting it.
* Done in testing phase

Debugging:

* It involves identifying, isolating and fixing the problems/bug.
* Debugging is the part of white box or unit testing
* Done in developing phase