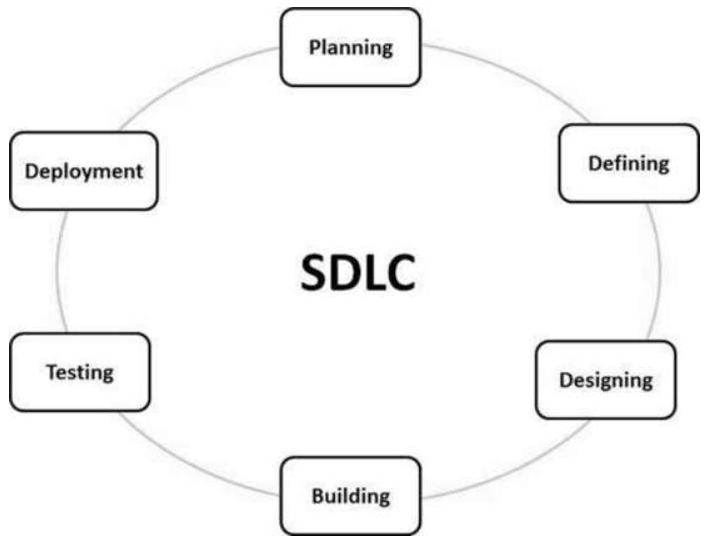
**Learned so for in Software Testing Program**

The first session of the software training program began with a small task for us to find out the meaning of a few basic terms such as

1. Software – A collection of programs that coordinates with the hardware to run a computer or a machine.
2. Application – It is a package that performs a specific task for end users and this comes under “software” but not vice versa.
3. Program – A fully functioning or ready to use application is known as program.
4. Code – It is a set of instructions written in a particular programming language.
5. Product - A product is any item or service you sell to serve a customer's need or want. They can be physical or virtual. Physical products include durable goods (like cars, furniture, and computers) and nondurable goods (like food and beverages).

This task provided a clarity and a chance to understand the difference among them. Then the discussion started with Software Development Life Cycle (SDLC).

**SDLC** – A process used to produce a high-quality software, that includes the following stages

 0 – Requirement analysis

1 – Planning

2 – Designing

3 – Development

4 – Testing

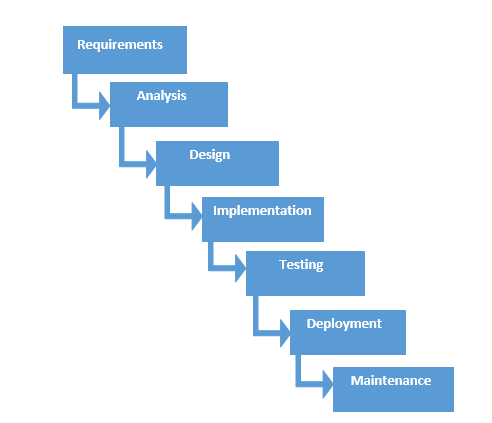
5 – Deploy

6 – Maintain

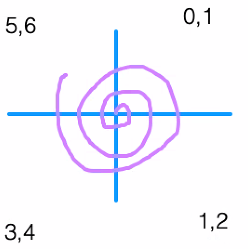
Different techniques or approaches in software testing are,

* + FAST – Facilitated Application Specification Technique
  + JAD – Joint Application Development
  + RAD – Rapid Application Development

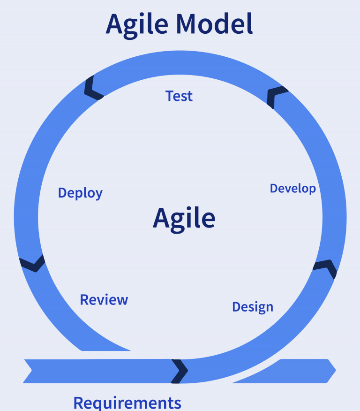
There are different models in SDLC like,

**Waterfall Model: (FAST)**

1. Requirement
2. Planning
3. Design
4. Development
5. Testing
6. Deployment

**Spiral Model: (JAD)**

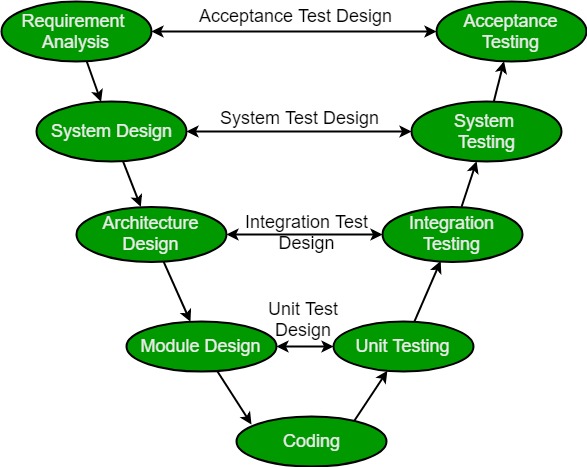
1. Requirements Analysis
2. Planning
3. Design
4. Development
5. Testing
6. Deploy
7. Maintenance

**Agile Model: (RAD)**

1. Requirements gathering
2. Design the requirements
3. Construction/ iteration
4. Testing/ Quality assurance
5. Deployment
6. Feedback

**V model**

The V-Model, also known as the Verification and Validation Model, is a software development and testing framework that emphasizes a structured and systematic approach to software development. It is an extension of the Waterfall model and is often used in industries where safety and reliability are critical, such as aerospace, automotive, and healthcare.



The V-Model is called so because of its characteristic V-shaped diagram that represents the various stages of development and testing. It illustrates the relationship between each phase of development and its corresponding testing phase

**Increment model**

The Incremental Model is a software development methodology where a project is divided into smaller, manageable parts or increments. Each increment represents a portion of the complete system's functionality, and these increments are developed and delivered one at a time in a sequential manner. In essence, it's an iterative approach to software development, but unlike some other iterative models, each increment is a fully functional and usable portion of the software.

**Prototype model**

The Prototype Model, also known as the Prototyping Model or Prototype Development Model, is a software development methodology that focuses on creating a prototype of the software to better understand and refine the requirements before the full-scale development of the final product. This model is particularly useful when the requirements are not well-defined or are subject to change. The primary goal of the prototype model is to involve users or stakeholders in the early stages of development to gather feedback and improve the software's design and functionality.

**Designations and their roles and responsibilities**

|  |  |
| --- | --- |
| **Roles/Responsibilities** | **Designations** |
| 1. Requirements Analysis | Client, Business Analyst |
| 1. Planning | Business Analyst, Product Owner or Product Manager |
| 1. Design | Product Owner or Product Manager, Designer Business Analyst |
| 1. Development | Developer |
| 1. Testing | Tester, Business Analyst, Product Owner or Product Manager |
| 1. Deploy | Tester, End User |

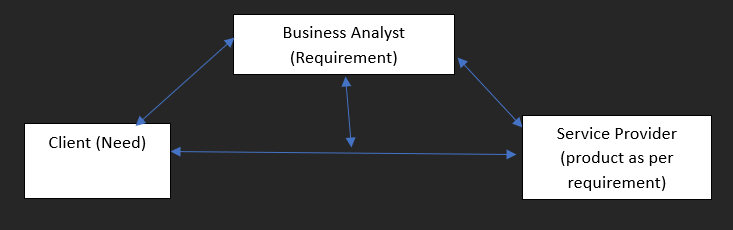
**Product Based Company** – Own product, own POs/PMs, own developers, own Bas. E.g.: Zoom, FANG.

**Service Based Company** – do not have a particular product but their primary business is to provide services. E.g.: CTS, TCS, INFOSYS.

**HYBRID** – A combination of product based and service-based companies. E.g.: IBM, MOOLYAED.

**Role of a Business Analyst (BA)**

A BA must be a domain expert as he acts as a mediator between client and service provider and oversees everything.



**Designations and Tools**

|  |  |
| --- | --- |
| **Designations** | **Tools** |
| Developers | IDE (Visual Studio Code, Eclipse) . |
| Testers | IDE, Cloud-based tools, Analytics tools for debugging. |
| Business Analyst | Analytics tools, Management tool. |
| PO/PM | Analytics tools, Management tool, Cloud-based tools, IDE. |

**Need for Software Testing**

The primary reason for software testing is to check and match the product with the official documents such as SRS, FRS etc or the fulfilment of product requirement. Added reason for software testing is to meet customer satisfaction.

**Basic terms in Software Testing**

Feature – a single component.

Module – a collection of multiple components as a single unit.

System – a collection of units as a single system.

**Approaches in Software Testing**

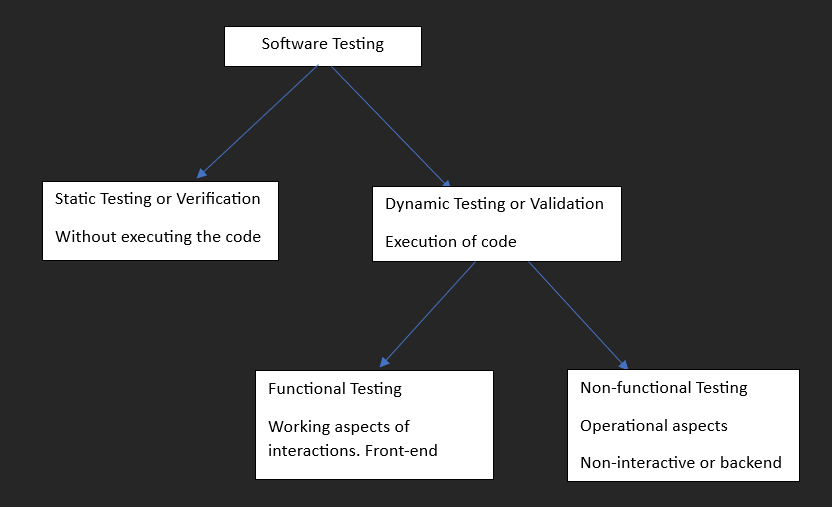
Manual Testing:

Manual Testing is a kind of software testing in which a software tester develops and executes the test cases without using any automated testing tools. The main objective of manual testing is to detect the issues, bugs, and defects of a software application.

The Testers uses the app like an end user operates manually and find the bugs

Automation Testing:

Automated testing is a software testing technique that automates the process of validating the functionality of software and ensures it meets requirements before being released into production. With automated testing, an organization can run specific software tests at a faster pace without human testers. The testers write a code to control the app, write code to break the code they find the bugs

**Classification of Software Testing **

**Levels of Testing**

* + Unit Testing
  + Integration Testing
  + System Testing
  + Acceptance Testing

**Unit Testing**

* + Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually scrutinized for proper operation.
  + When you test a unit of code.
  + Unit test done by developers.

**Integration Testing**

* + Integration testing is a type of software testing where components of the software are gradually integrated and then tested as a unified group. Usually, these components are already working well individually, but they may break when integrated with other components.
  + When you Integrate 2 units or more and test flow of the code.
  + Done by Developer and Tester.

**System Testing**

* System testing, also referred to as system-level testing or system integration testing, is the process in which a quality assurance (QA) team evaluates how the various components of an application interact together in the full, integrated system or application.
* Testing the entire system,
* Done by Tester.

**Acceptance Testing**

* + Acceptance testing is a quality assurance (QA) process that determines to what degree an application meets end users' approval. Depending on the organization, acceptance testing might take the form of beta testing, application testing, field testing or end-user testing.
  + Done by Tester and End User.

**By**

**Bug Squashers**