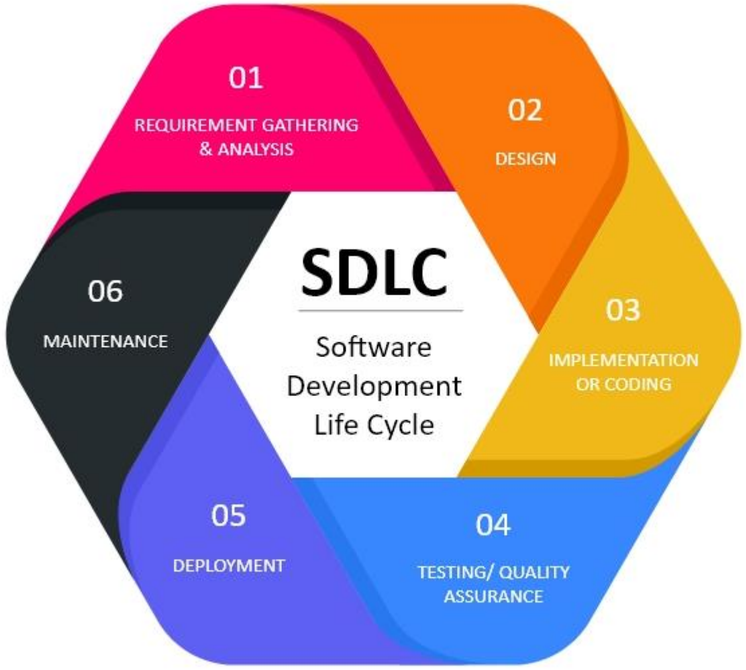
TestLeaf

Always Ahead

SDLC (Software Development Life Cycle)



TestLeaf
Always Ahead

SDLC (Software Development Life Cycle)

SDLC is a structured process for the production of high-quality, low-cost software, in the shortest possible production time

SDLC Phases

1) Requirement Analysis and Planning
2) Design
3) Development
4) Testing
5) Deployment
6) Maintenance

TestLeaf
Always Ahead

SDLC Phases

Requirement Analysis and Planning

- To identify the expectation of the user/client for an application/software that has to be newly built or modified.
- Recognition of the risks involved is also done at this stage.
- The outcome of this phase is requirement documents FRD (Functional requirement Document) or BRD (Business Requirement Document), the naming convention of the document varies in each and every organization.

BRD - Business Requirement Document

- The Business/Client/other Stakeholders provide a requirement.
- One requirement can be small or big.
- If its a big requirement has to be broken wherever it requires and taken as multiple requirements.

FRD - Functional Requirement Document:

- The Process to reach the expectancy of the BRD is an FRD.
- How to develop the expected requirement, What are the features and Functionalities or Tools / Systems used and what sort of inter-dependencies they have.
- And how the systems will react with the two newly created system when they start working together when this Functionality takes place.
- What are the Assumptions a BA has made and what are the Limitation the system will have if went with the current approach.
- It will be helpful if there are any supporting Process Flow-charts / Flow diagrams etc..

The Business Requirement Document (BRD) describes the high-level business needs whereas the Functional Requirement Document (FRD) outlines the functions required to fulfill the business need. BRD answers the question what the business wants to do whereas the FRD gives an answer to how should it be done. FRD is derived from a BRD.

SDLC Phases

Design

High-level Design

- Brief description of the overall architecture of the application with the integrated modules
- Macro level design
- Converts business requirements to high level solution
- Solution Architect creates it before any other tech documentation
- Results: database design, functional design, review record

Low-level Design

- Detailed description about each and every module with functional logic, interfaces, and Database Tables which was mentioned in HLD
- Micro level design
- Convers high level solution to detailed solution
- Designers & Developers creates it after the high level design document
- Results: program specification and unit test plan

Development

- Developers start to build the entire system by developing code. In the coding phase, tasks are divided into units or modules and assigned to the various developers

SDLC Phases

Testing

- After completion of development, the testing team starts testing the functionality of the entire system.
- This is done to verify that the entire application works according to the customer's requirements. During this phase, QA and testing team may find some bugs/defects which they communicate to developers.
- The development team fixes the bug and sends it back to QA for a re-test. This process continues until the software is bug-free, stable, and working according to the business needs of that system.

Deployment

- Once testing is completed, the application meets the client's requirement then deployment of the application will be done in the Production Environment

Maintenance

- Customers/ End users start using the application/Software, Enhancements or bug fixes will be done based on the user's feedback.

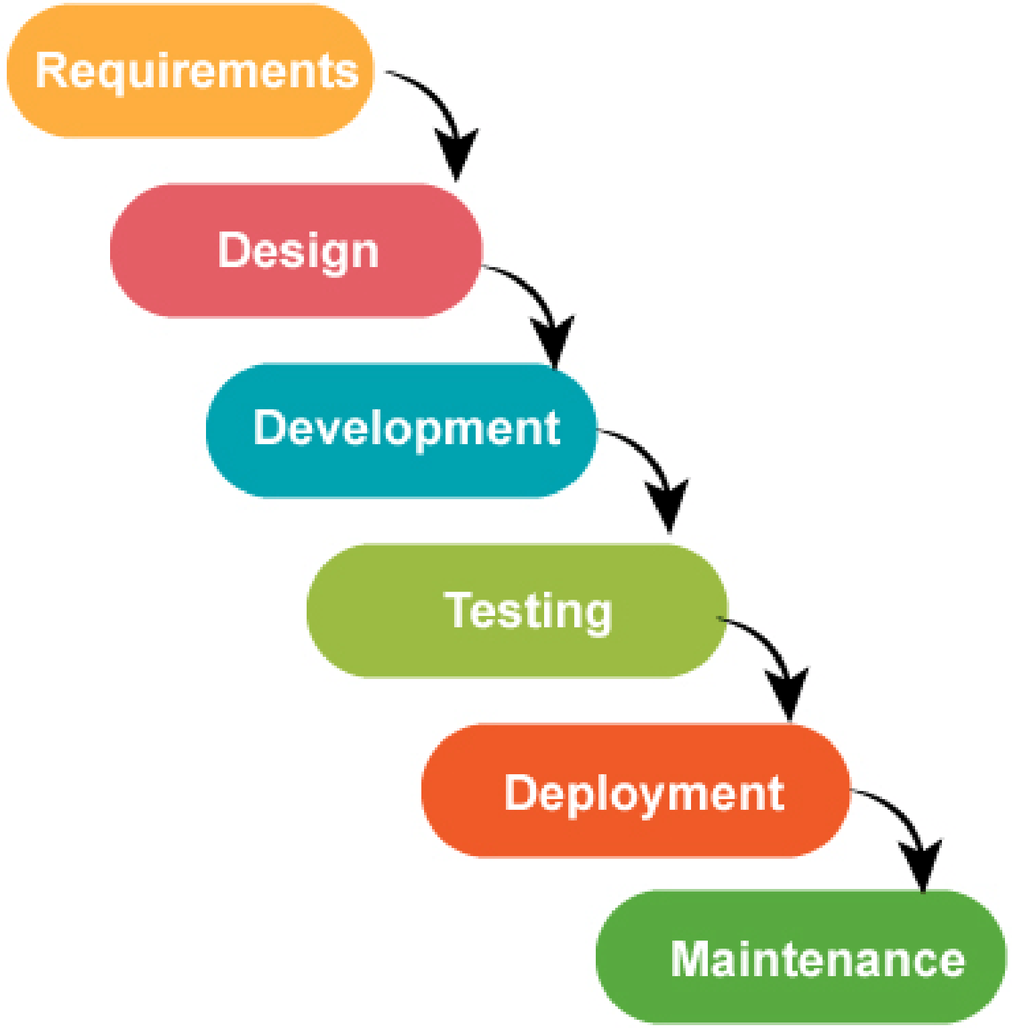
Life Cycle Models

- Waterfall Model
- V- Model Model

TestLeaf
Always Ahead

Waterfall Model

Linear sequential Model. The next phase begins only after the completion of the previous Phase



Advantages of Waterfall Model

- Each and every phase is clearly defined and there is no overlapping of the phases as it is a step-by-step process.
- Each phase is completely done one at a time.
- Requirements are clear to everyone at the beginning of the project
- Works well for the smaller projects and gets complicated for large projects.
- Clear planning of project team structure reduces the problematic issues.
- Release date and final cost will be calculated before starting the project.
- Best suitable for small projects as the requirements are clearly stated.

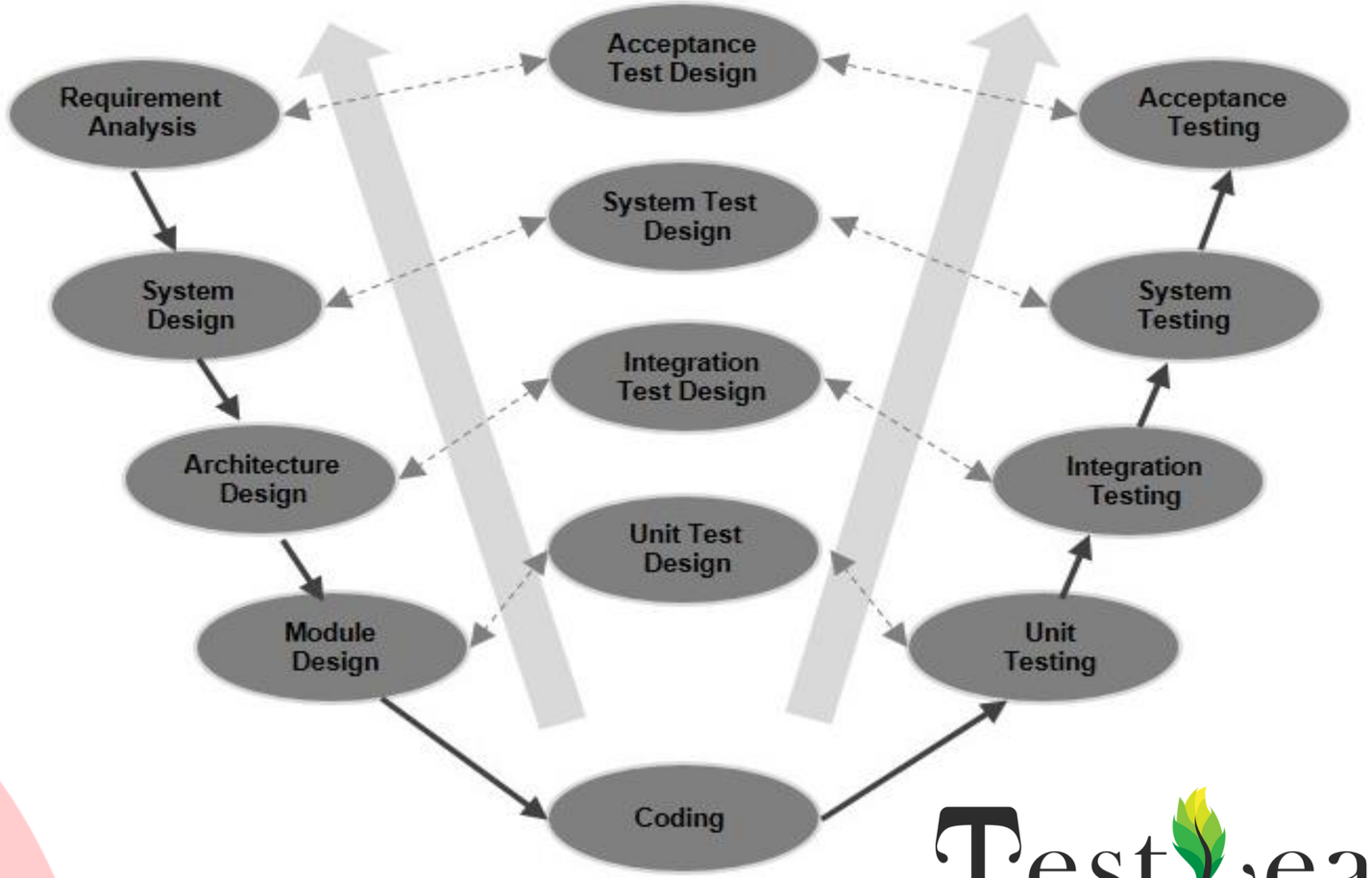
TestLeaf
Always Ahead

Disadvantages of Waterfall Model

- The problem in Adapting Changes to the Requirement
- Time to market for the product is high
- Testing is delayed since it is the fourth phase
- Integration is done at the end
- Not suitable for complex projects
- Clients valuable feedback cannot be included with the ongoing development phase

TestLeaf
Always Ahead

V-Model (Verification and Validation Model)



TestLeaf
Always Ahead

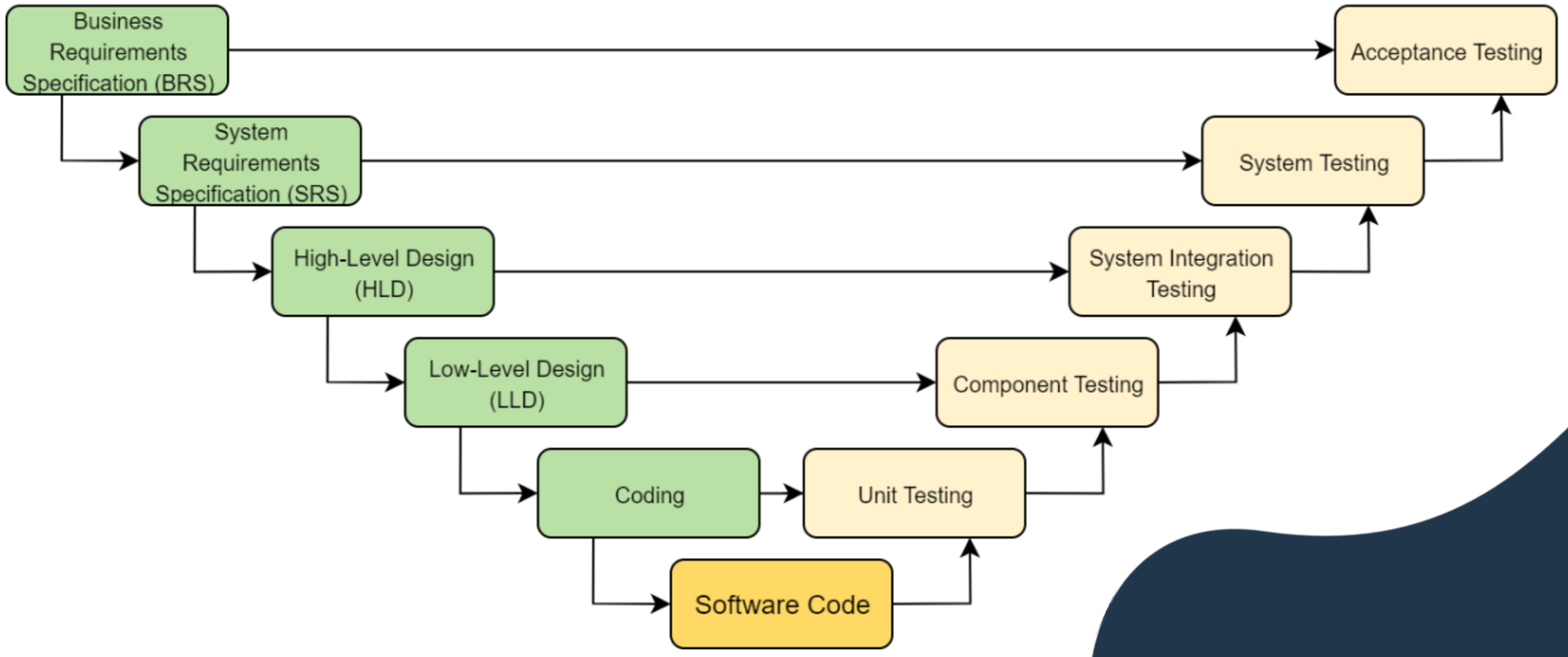
V-Model (Verification and Validation Model)

Developer

The timeline for developer starts way before the coding starts.

QA Engineer

The testers start with the code and then move up to the business requirements.



TestLeaf
Always Ahead

V-Model (Verification and Validation Model)

Verification:
Static Testing process. Develop the Testcases at all levels with respect to the requirement document to ensure that the Business requirement is satisfied before the development of the application

Validation:
Dynamic Testing Process which has been done once after the development of code has been completed with respect to the application developed.

Advantages of V Model

- Testing Methods like planning, test designing happens well before coding.
- This saves a lot of time. Hence a higher chance of success over the waterfall model.
- Avoids the downward flow of the defects.
- Works well for small plans where requirements are easily understood.

TestLeaf
Always Ahead

Disadvantages of V Model

- High risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing.
- Once an application is in the testing stage, it is difficult to go back and change a functionality.
- No working software is produced until late during the life cycle.

TestLeaf
Always Ahead