SDLC Models
****

1.Waterfall Model

2.Iterative Model

3.Spiral Model

4.V-Model

5.Big Bang Model

Waterfall Model:

Definition:

*Waterfall Model is a sequential model that divides software development into pre-defined phases.

*Each phase is designed for performing specific activity during the SDLC phase.

*It was introduced in 1970 by Winston Royce.

The stages of the waterfall model:

1.Requirement Gathering

2.System Analysis

3.System Design

4.Coding

5.Implementation

6.Testing

7.Deployment

8.Operation and Maintainance

Examples of Waterfall Model:

1.Waterfall model was used to develop enterprise applications like ,

2.Customer Relationship Management (CRM) systems

3.Human Resource Management Systems (HRMS)

4.Supply Chain Management Systems

5.Inventory Management Systems,

6.Point of Sales (POS) systems for Retail chains etc.

A real life example of a waterfall model:

1.Department of Defence

2.military and aircraft programs

When to use SDLC Waterfall Model?

1.Waterfall Methodology can be used when:

2.Requirements are not changing frequently

3.Application is not complicated and big

4.Project is short

5.Requirement is clear

6.Environment is stable

7.Technology and tools used are not dynamic and is stable

8.Resources are available and trained

Advantages and Disadvantages of Waterfall Model:

Advantages:

1.Before the next phase of development, each phase must be completed.

2.Suited for smaller projects where requirements are well defined.

3.They should perform quality assurance test (Verification and Validation) before completing each stage.

4.Elaborate documentation is done at every phase of the software's development cycle.

5.Project is completely dependent on project team with minimum client intervention.

6.Any changes in software is made during the process of the development.

Dis-Advantages:

1.Error can be fixed only during the phase.

2.It is not desirable for complex project where requirement changes frequently.

3.Testing period comes quite late in the developmental process.

4.Documentation occupies a lot of time of developers and testers.

5.Clients valuable feedback cannot be included with ongoing development phase.

6.Small changes or errors that arise in the completed software may cause a lot of problems.

Iterative Model

****

Definition:

*The Iterative Model allows the accessing earlier phases, in which the variations made respectively.

*In the Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed.

*Iterative development is sometimes called as circular or evolutionary development.

* An iterative life cycle model does not attempt to start with a full specification of requirements.

* Instead, development begins by specifying and implementing just part of the software, which is then reviewed to identify further requirements.

* This process is then repeated, producing a new version of the software at the end of each iteration of the model.

When to use the Iterative Model?

When requirements are defined clearly and easy to understand.

When the software application is large.

When there is a requirement of changes in future.

Advantages of Iterative Model:

Testing and debugging during smaller iteration is easy.

A Parallel development can plan.

It is easily acceptable to ever-changing needs of the project.

Risks are identified and resolved during iteration.

Limited time spent on documentation and extra time on designing.

Disadvantages of Iterative Model:

It is not suitable for smaller projects.

More Resources may be required.

Design can be changed again and again because of imperfect requirements.

Requirement changes can cause over budget.

Project completion date not confirmed because of changing requirements.

Spiral Model

****

*The Spiral Model is a risk – driven model, the focus is on managing risk through multiple iterations of the software development.

*The best-suited example for Spiral Model is MS-Excel because MS-Excel sheet is having several cells, which are the components of an excel sheet.

* is an evolutionary software process model that couples the iterative feature of prototyping with the controlled and systematic aspects of the linear sequential model.

*It implements the potential for rapid development of new versions of the software. Using the spiral model, the software is developed in a series of incremental releases.

*During the early iterations, the additional release may be a paper model or prototype. During later iterations, more and more complete versions of the engineered system are produced.

Risk Management in Spiral Model:

risk analysis and handling in every phase, improving security and the chances of avoiding attacks and breakages.

Each cycle in the spiral is divided into four parts:

Objective setting: Each cycle in the spiral starts with the identification of purpose for that cycle, the various alternatives that are possible for achieving the targets, and the constraints that exists.

Risk Assessment and reduction: The next phase in the cycle is to calculate these various alternatives based on the goals and constraints. The focus of evaluation in this stage is located on the risk perception for the project.

Development and validation: The next phase is to develop strategies that resolve uncertainties and risks. This process may include activities such as benchmarking, simulation, and prototyping.

Planning: Finally, the next step is planned. The project is reviewed, and a choice made whether to continue with a further period of the spiral. If it is determined to keep, plans are drawn up for the next step of the project.

The development phase depends on the remaining risks. For example, if performance or user-interface risks are treated more essential than the program development risks, the next phase may be an evolutionary development that includes developing a more detailed prototype for solving the risks.

*The risk-driven feature of the spiral model allows it to accommodate any mixture of a specification-oriented, prototype-oriented, simulation-oriented, or another type of approach.

Advantages:

1.High amount of risk analysis

2.Useful for large and mission-critical projects.

Disadvantages:

1.Can be a costly model to use.

2.Risk analysis needed highly particular expertise

3.Doesn't work well for smaller projects.

V-Model

****

V-Model also referred to as the Verification and Validation Model. In this, each phase of SDLC must complete before the next phase starts. It follows a sequential design process same as the waterfall model. Testing of the device is planned in parallel with a corresponding stage of development.

Verification:

*It involves a static analysis method (review) done without executing code.

*It is the process of evaluation of the product development process to find whether specified requirements meet.

Validation:

*It involves dynamic analysis method (functional, non-functional), testing is done by executing code.

*Validation is the process to classify the software after the completion of the development process to determine whether the software meets the customer expectations and requirements.

*So V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation process is joined by coding phase in V-shape. Thus it is known as V-Model.

There are the various phases of Verification Phase of V-model:

Business requirement analysis: This is the first step where product requirements understood from the customer's side. This phase contains detailed communication to understand customer's expectations and exact requirements.

System Design: In this stage system engineers analyze and interpret the business of the proposed system by studying the user requirements document.

Architecture Design: The baseline in selecting the architecture is that it should understand all which typically consists of the list of modules, brief functionality of each module, their interface

relationships, dependencies, database tables, architecture diagrams, technology detail, etc. The integration testing model is carried out in a particular phase.

Module Design: In the module design phase, the system breaks down into small modules. The detailed design of the modules is specified, which is known as Low-Level Design

Coding Phase: After designing, the coding phase is started. Based on the requirements, a suitable programming language is decided. There are some guidelines and standards for coding. Before checking in the repository, the final build is optimized for better performance, and the code goes through many code reviews to check the performance.

There are the various phases of Validation Phase of V-model:

Component/Unit Testing: In the V-Model, Unit Test Plans (UTPs) are developed during the module design phase. These UTPs are executed to eliminate errors at code level or unit level. A unit is the smallest entity which can independently exist, e.g., a program module. Unit testing verifies that the smallest entity can function correctly when isolated from the rest of the codes/ units.

Integration Testing: Integration Test Plans are developed during the Architectural Design Phase. These tests verify that groups created and tested independently can coexist and communicate among themselves.

System Testing: System Tests Plans are developed during System Design Phase. Unlike Unit and Integration Test Plans, System Tests Plans are composed by the client’s business team. System Test ensures that expectations from an application developer are met.

Acceptance Testing: Acceptance testing is related to the business requirement analysis part. It includes testing the software product in user atmosphere. Acceptance tests reveal the compatibility problems with the different systems, which is available within the user atmosphere. It conjointly discovers the non-functional problems like load and performance defects within the real user atmosphere.

* Alpha Testing is done within the organization, while Beta Testing is done in the user's environment.

*During Alpha Testing only functionality and usability are tested,

*while, During Beta Testing usability, functionality, security and reliability are tested to the same depth.

Big Bang Model

****

Definition:

The Big Bang model is a start-from-scratch SDLC model. It's the simplest SDLC (Software Development Life Cycle) model because it doesn't require any planning.

When to use Big Bang Model?

As we discussed above, this model is required when this project is small like an academic project or a practical project. This method is also used when the size of the developer team is small and when requirements are not defined, and the release date is not confirmed or given by the customer.

Advantages of Big Bang Model:

There is no planning required.

Simple Model.

Few resources required.

Easy to manage.

Flexible for developers.

Disadvantages of Big Bang Model:

There are high risk and uncertainty.

Not acceptable for a large project.

If requirements are not clear that can cause very expensive.

The Real time example of Big - Bang Model:

"HACKATHON" refers to a platform, codefest, or event in which developers or students compete to create a working piece of software in a short amount of time. This Hackathon is one of the best example that comes under the category of Big Bang Model.

RAD MODEL

****

Definition:

*RAD Model is Rapid Application Development model.

*It is a type of incremental model.

*In RAD Model the components or functions are developed in parallel as if they were mini projects.

*The developments are time boxed, delivered and then assembled into a working prototype.

The various phases of RAD :

1.Business modeling: The information flow is identified between various business functions.

2.Data modeling: Information gathered from business modeling is used to define data objects that are needed for the business.

3.Process modeling: Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective. Description are identified and created for CRUD(create, read, update, and delete.) of data objects.

4.Application generation: Automated tools are used to convert process models into code and the actual system.

5.Testing and turnover: Test new components and all the interfaces.

When to use RAD Model:

RAD should be used when there is a need to create a system that can be modularized in 2-3 months of time.

It should be used if there's high availability of designers for modeling and the budget is high enough to afford their cost along with the cost of automated code generating tools.

RAD SDLC model should be chosen only if resources with high business knowledge are available and there is a need to produce the system in a short span of time (2-3 months).

Advantages of the RAD model:

Reduced development time.

Increases reusability of components

Quick initial reviews occur

Encourages customer feedback

Integration from very beginning solves a lot of integration issues.

Disadvantages of RAD model:

Depends on strong team and individual performances for identifying business requirements.

Only system that can be modularized can be built using RAD

Requires highly skilled developers/designers.

High dependency on modeling skills

Inapplicable to cheaper projects as cost of modeling and automated code generation is very high.