

Software development Life cycle

Profesor PhD. Daniela DĂNCIULESCU

SOFTWARE DEVELOPMENT LIFE CYCLE

Phases, Models, Process and Methodologies



What does *Software Development Life Cycle (SDLC)* mean?

- SDLC is a framework defining tasks performed at each step in the software development process.
- SDLC is a structure followed by a development team within the software organization.
- It consists of a detailed plan describing how to develop, maintain and replace specific software.
- The life cycle defines a methodology for improving the quality of software and the overall development process.
- The software development life cycle is also known as the software development process.

ACTIVITIES

- ▶ Requirement gathering and analysis
- ▶ Design
- ▶ Implementation or coding
- ▶ Testing
- ▶ Deployment
- ▶ Maintenance



1. Requirement Gathering and Analysis

- ▶ During this phase, all the relevant information is collected from the customer to develop a product as per their expectation.
- ▶ Any ambiguities must be resolved in this phase only.
- ▶ It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry.
- ▶ This stage need teams to get detailed and precise requirements. This helps companies to finalize the necessary timeline to finish the work of that system.
- ▶ **For Example:** A customer wants to have an application which involves money transactions.
 - ▶ In this case, the requirement has to be clear like
 - ▶ what kind of transactions will be done
 - ▶ how it will be done
 - ▶ in which currency it will be done
 - ▶ Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product.
 - ▶ In case of any ambiguity, a call is set up for further discussion.
 - ▶ Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created.
 - ▶ This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

2. Design

- ▶ In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.
- ▶ This design phase serves as input for the next phase of the model.
- ▶ There are two kinds of design documents developed in this phase:
 - ▶ High-Level Design (HLD)
 - ▶ Brief description and name of each module
 - ▶ An outline about the functionality of every module
 - ▶ Interface relationship and dependencies between modules
 - ▶ Database tables identified along with their key elements
 - ▶ Complete architecture diagrams along with technology details
 - ▶ Low-Level Design(LLD)
 - ▶ Functional logic of the modules
 - ▶ Database tables, which include type and size
 - ▶ Complete detail of the interface
 - ▶ Addresses all types of dependency issues
 - ▶ Listing of error messages
 - ▶ Complete input and outputs for every module

3. Implementation or Coding

- ▶ Implementation/Coding starts once the developer gets the Design document.
- ▶ The Software design is translated into source code.
- ▶ All the components of the software are implemented in this phase.
- ▶ Developers start build the entire system by writing code using the chosen programming language.
- ▶ Developer needs to follow certain predefined coding guidelines. They also need to use programming tools like compiler, interpreters, debugger to generate and implement the code
- ▶ In the coding phase, tasks are divided into units or modules and assigned to the various developers.
- ▶ It is the longest phase of the Software Development Life Cycle process.

4. Testing

- ▶ Testing starts once the coding is complete and the modules are released for testing.
- ▶ During this phase, QA and testing team may find some bugs/defects which they communicate to developers.
- ▶ Retesting, regression testing is done until the point at which the software is as per the customer's expectation.
- ▶ Testers refer SRS document to make sure that the software is as per the customer's standard.

5. Deployment

- ▶ Once the product is tested, it is deployed in the production environment or first [UAT \(User Acceptance testing\)](#) is done depending on the customer expectation.
- ▶ In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing.
- ▶ If the customer finds the application as expected, then sign off is provided by the customer to go live.

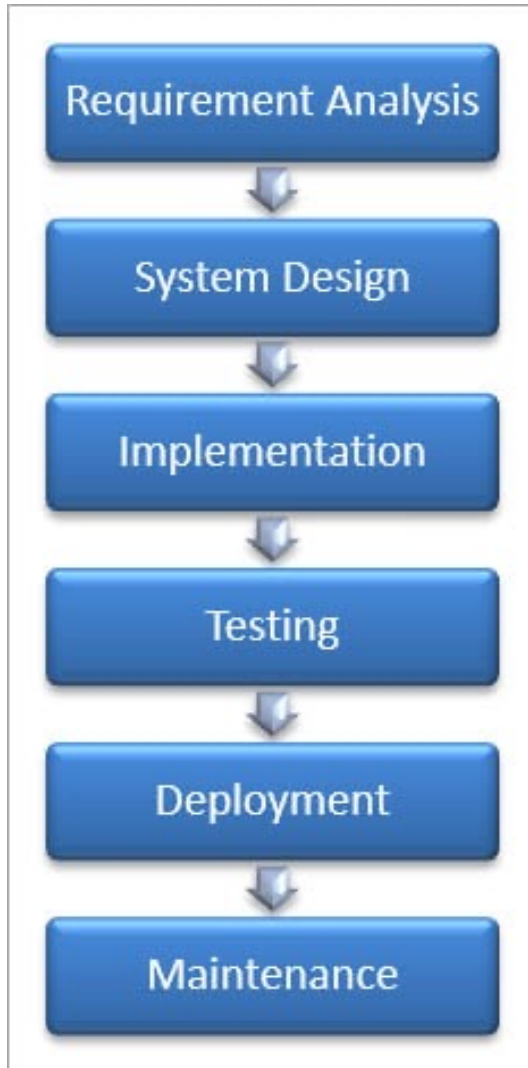
6. Maintenance

- ▶ Once the system is deployed, and customers start using the developed system, following 3 activities occur:
 - ▶ Bug fixing - bugs are reported because of some scenarios which are not tested at all
 - ▶ Upgrade - Upgrading the application to the newer versions of the Software
 - ▶ Enhancement - Adding some new features into the existing software
- ▶ The main focus of this phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.

SOFTWARE DEVELOPMENT MODELS

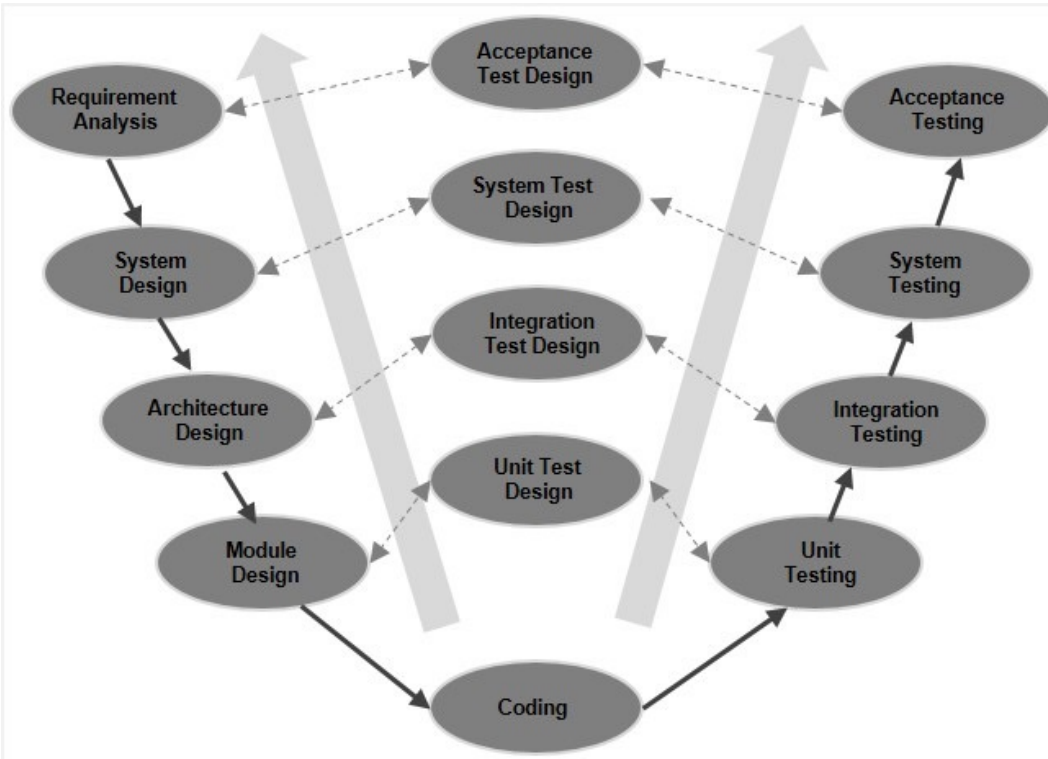
- ▶ Waterfall Model
- ▶ V-Shaped Model
- ▶ Prototype Model
- ▶ Spiral Model
- ▶ Big Bang Model
- ▶ Agile Model

1. WATERFALL MODEL



- ▶ Is the very first model that is used in SDLC.
 - ▶ It is also known as the linear sequential model
 - ▶ Involves finishing each phase completely before commencing the next one.
- ▶ **Advantages:**
- ▶ Waterfall model is the simple model which can be easily understood and is the one in which all the phases are done step by step.
 - ▶ Deliverables of each phase are well defined, and this leads to no complexity and makes the project easily manageable.
- ▶ **Disadvantages:**
- ▶ Waterfall model is time-consuming
 - ▶ Cannot be used in the short duration projects
 - ▶ Cannot be used for the projects which have uncertain requirement

2. V-SHAPED MODEL



- ▶ Is also known as Verification and Validation Model.
- ▶ In this model Verification & Validation goes hand in hand i.e. development and testing goes parallel.
- ▶ V model and waterfall model are the same except that the test planning and testing start at an early stage in V-Model.

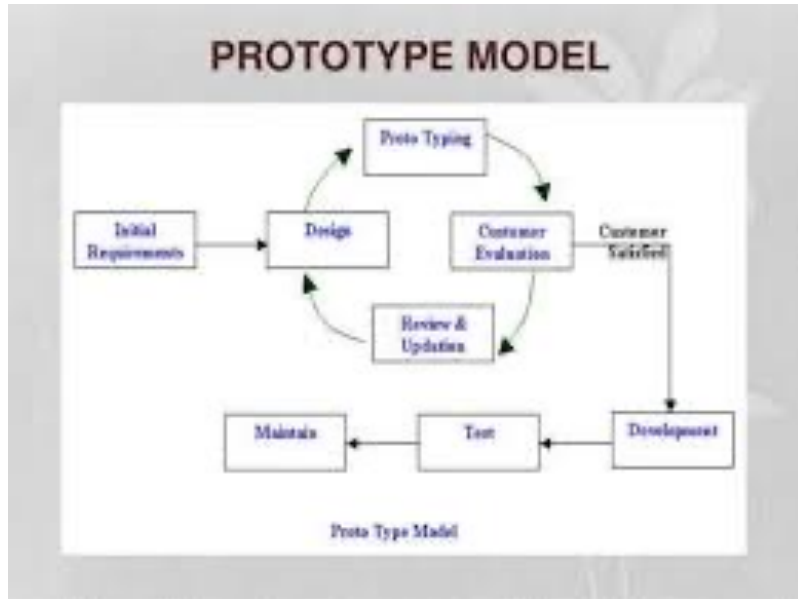
▶ Advantages:

- ▶ It is a simple and easily understandable model.
- ▶ V -model approach is good for smaller projects wherein the requirement is defined and it freezes in the early stage.
- ▶ It is a systematic and disciplined model which results in a high-quality product.

▶ Disadvantages:

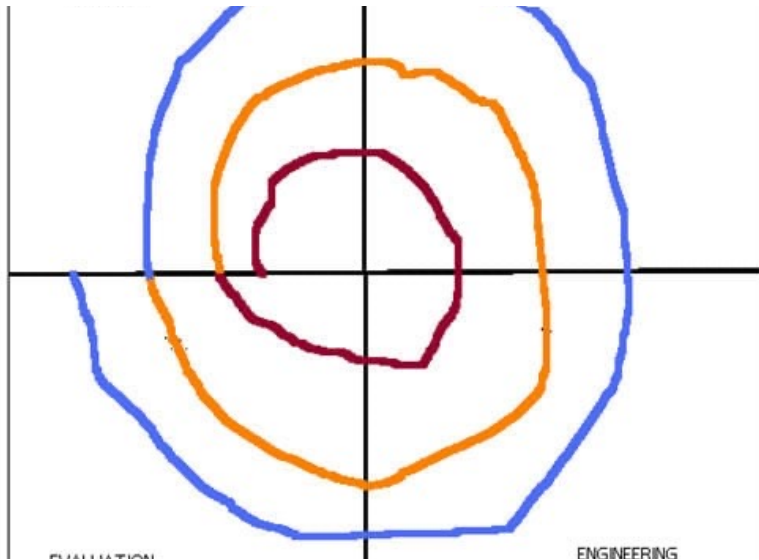
- ▶ V-shaped model is not good for ongoing projects.
- ▶ Requirement change at the later stage would cost too high.

3. PROTOTYPE MODEL



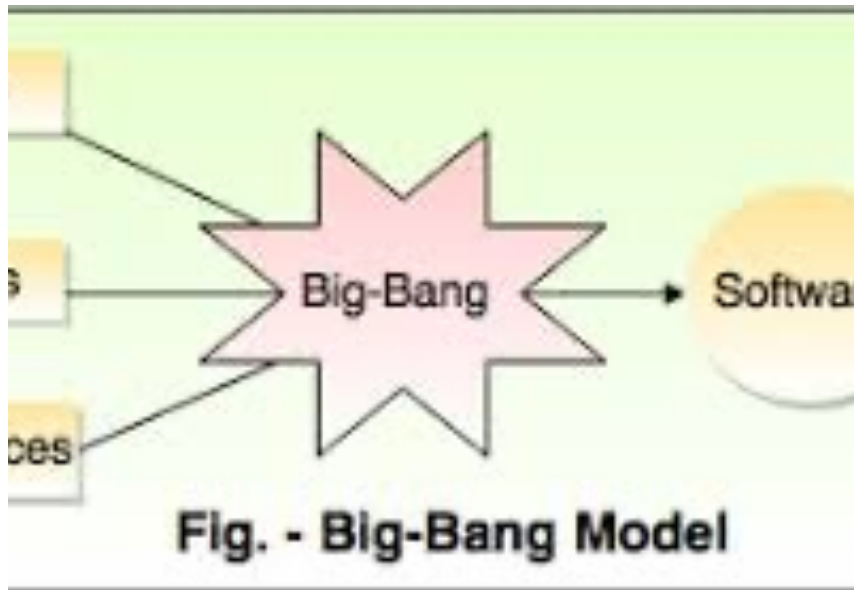
- ▶ Is a model in which the prototype is developed prior to the actual software.
- ▶ Have limited functional capabilities and inefficient performance when compared to the actual software.
- ▶ This is a valuable mechanism for understanding the customers' needs.
- ▶ **Advantages:**
 - ▶ Reduces the cost and time of development as the defects are found much earlier.
 - ▶ Missing feature or functionality or a change in requirement can be identified in the evaluation phase and can be implemented in the refined prototype.
 - ▶ Involvement of a customer from the initial stage reduces any confusion in the requirement or understanding of any functionality.
- ▶ **Disadvantages:**
 - ▶ Since the customer is involved in every phase, the customer can change the requirement of the end product which increases the complexity of the scope and may increase the delivery time of the product.

4. SPIRAL MODEL



- ▶ Includes iterative and prototype approach.
- ▶ Spiral model phases are followed in the iterations.
- ▶ The loops in the model represent the phase of the SDLC process.
- ▶ Spiral Model has four phases:
 - ▶ Planning
 - ▶ Risk Analysis
 - ▶ Engineering
 - ▶ Evaluation
- ▶ **Advantages:**
 - ▶ Risk Analysis is done extensively using the prototype models.
 - ▶ Any enhancement or change in the functionality can be done in the next iteration.
- ▶ **Disadvantages:**
 - ▶ The spiral model is best suited for large projects only.
 - ▶ The cost can be high as it might take a large number of iterations which can lead to high time to reach the final product.

5. BIG BANG MODEL



- ▶ Does not have any defined process.
- ▶ Money and efforts are put together as the input and output come as a developed product which might be or might not be the same as what the customer needs.
- ▶ Does not require much planning and scheduling.
- ▶ The developer does the requirement analysis & coding and develops the product as per his understanding.
- ▶ This model is used for small projects only.
- ▶ There is no testing team and no formal testing is done, and this could be a cause for the failure of the project.

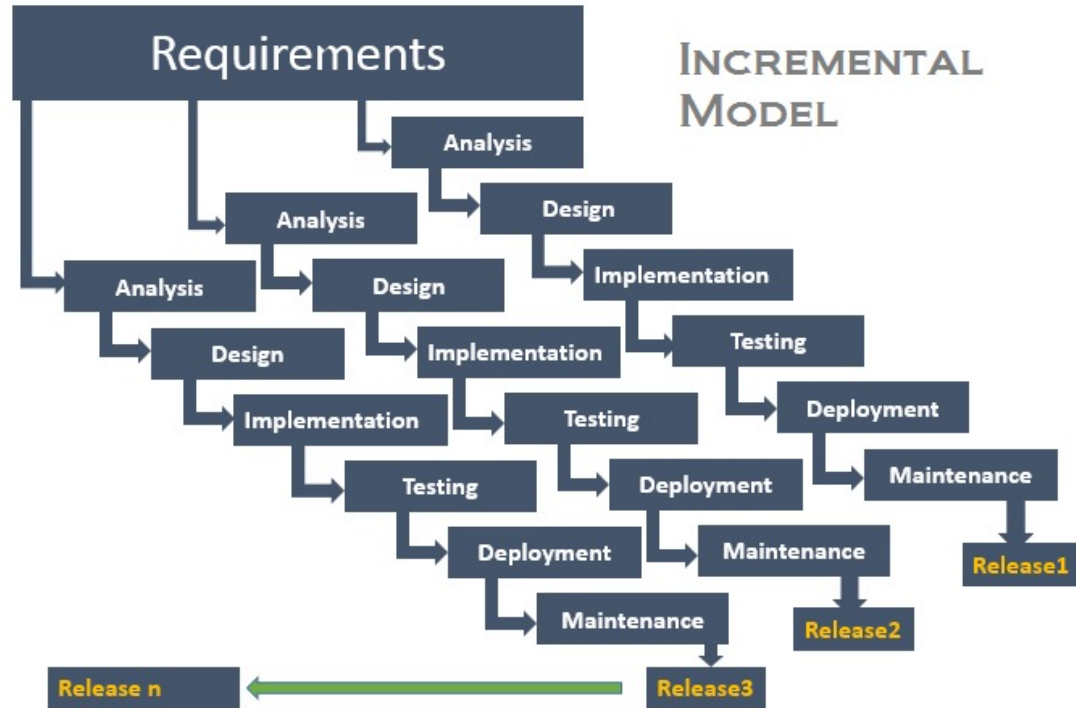
▶ Advantages:

- ▶ It's a very simple Model.
- ▶ Less Planning and scheduling is required.
- ▶ The developer has the flexibility to build the software of their own.

▶ Disadvantages:

- ▶ Big Bang models cannot be used for large, ongoing & complex projects.
- ▶ High risk and uncertainty.

5. INCREMENTAL MODEL



- ▶ Focuses more on flexibility while developing a product rather than on the requirement.
- ▶ This life cycle model involves multiple development cycles.
- ▶ The cycles are divided up into smaller iterations.
- ▶ It is not developed as a complete product in one go.
- ▶ In agile iterations are termed as sprints. Each sprint lasts for 2-4 weeks.
- ▶ At the end of each sprint, the product owner verifies the product and after his approval, it is delivered to the customer.
- ▶ Customer feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint.
- ▶ Testing is done in each sprint to minimize the risk of any failures.

6. INCREMENTAL MODEL

▶ Advantages:

- ▶ It allows more flexibility to adapt to the changes.
- ▶ The new feature can be added easily.
- ▶ Customer satisfaction as the feedback and suggestions are taken at every stage.

▶ Disadvantages:

- ▶ Lack of documentation.
- ▶ Agile needs experienced and highly skilled resources.
- ▶ If a customer is not clear about how exactly they want the product to be, then the project would fail.



▶ Let's talk!