

SOFTWARE DEVELOPMENT LIFECYCLE MODELS

1

Guided by:

Mr. Darsan

Presented by:

Nivyamol P Varghese

M.Tech-CSE

CONTENTS

- SPIRAL MODEL
- ITERATIVE MODEL
- V-MODEL

SPIRAL MODEL

- It is combining elements of both design and prototyping-in-stages, in an effort to combine advantages of top-down and bottom-up concepts.
- This model of development combines the features of the prototyping model and the waterfall model.
- The spiral model is favored for large, expensive, and complicated projects.
- This model uses many of the same phases as the waterfall model, in essentially the same order, separated by planning, risk Assessment, and the building of prototypes and simulations.
- The diagrammatic representation of this model appears like a spiral with many loops.

- Exact number of loop is not fixed and can vary.
- Each loop is called phase of the software process.
- More flexible compared to other models, since exact number of phases are not fixed.
- Over each loop, one or more features of the product are elaborated and analyzed and the risks at that point of time are identified and are resolved through prototyping. Based on this the identified features are implemented.
- Each phase of this model is split into four sectors or quadrants.

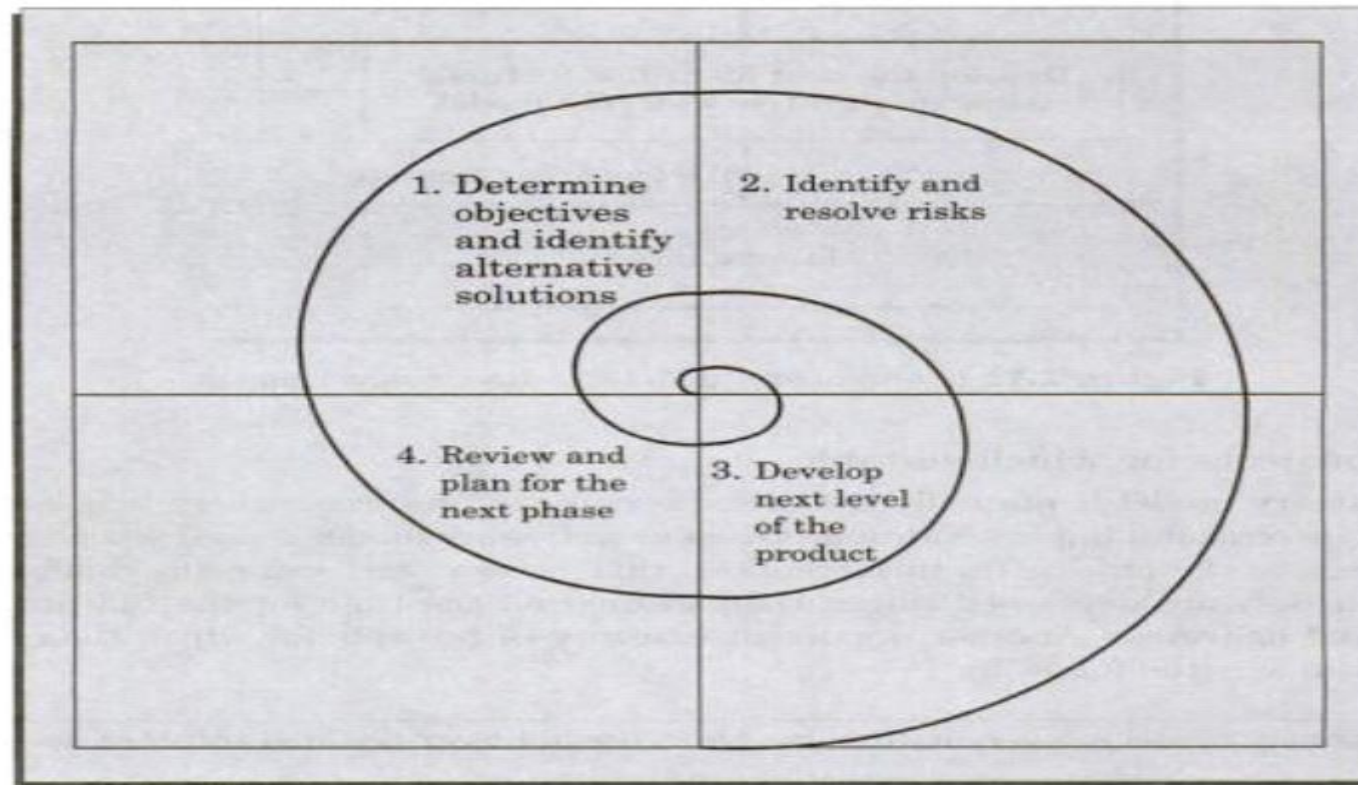


Figure 2.8: Spiral model of software development.

- In first quadrant, some features of the product are identified based on the severity of the risk and how crucial it is to the overall product development.
- Implementation of the identified features forms the objective of the phase analyzed.
- The objectives are investigated, elaborated, and Alternative solutions are considered.
- During the second quadrant, solutions are evaluated to select the best possible solution.
- At the end of the third iteration, the identified features have been implemented and the next version of the product is available.
- Results are reviewed in the fourth quadrant .
- Suitable for projects having many unknown risks.

CONT...

Advantages	Disadvantages
<ul style="list-style-type: none">▪ Estimates become more realistic as work progresses.▪ Changing requirements can be accommodated.▪ Development can be divided into smaller parts and more risky parts can be developed earlier which helps better risk management.▪ Users see the system early and give corrective feedback.▪ A more accurate end product.	<ul style="list-style-type: none">▪ High cost and time to reach the final product.▪ Needs special skills to evaluate the risk and assumptions.▪ Highly customized limiting re-usability.▪ Time spent planning, setting objectives, doing risk analysis and prototyping will be huge.▪ Complex model, Requires knowledgeable staffs, Not suitable for outsourced projects.

ITERATIVE MODEL

8

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 can then be reviewed in order to identify further requirements. This process is then repeated, producing new version of software for each cycle of a model.

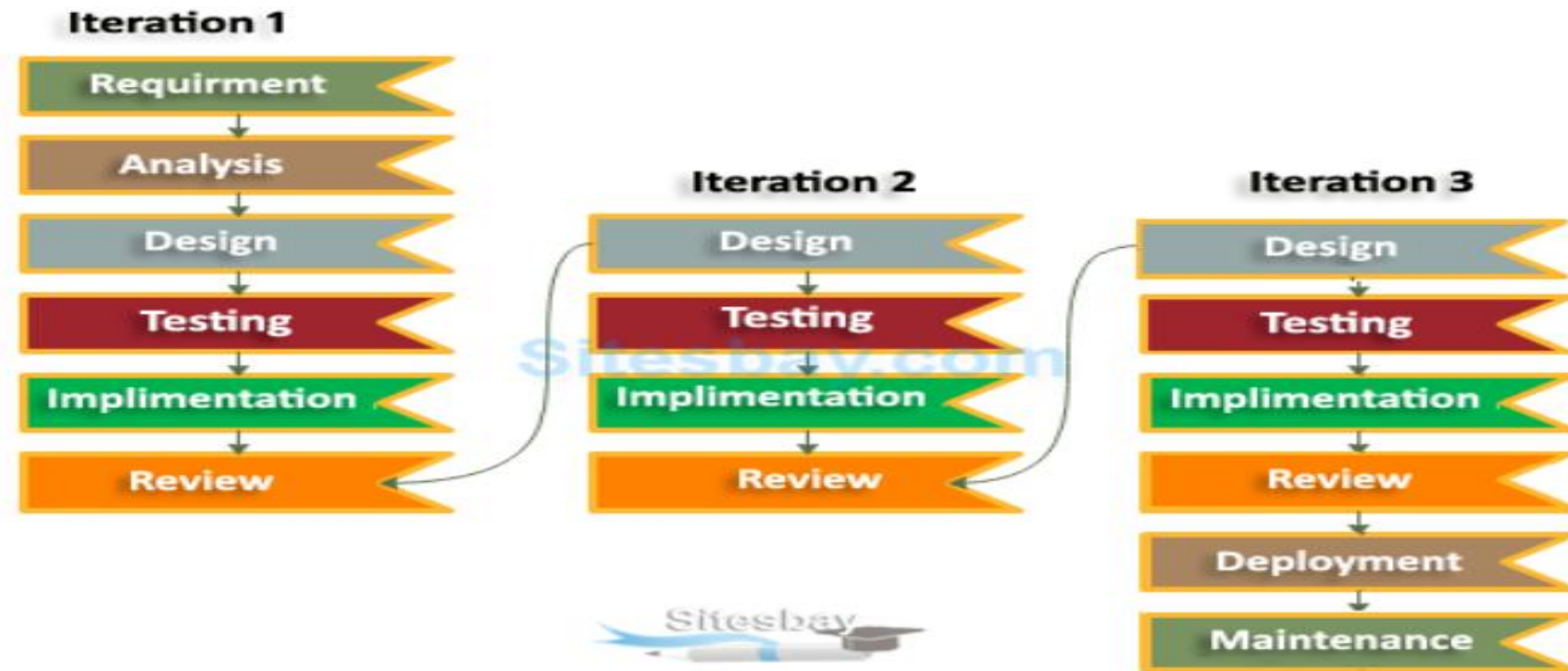


Fig: Iterative Model

- The Iterative Model allows the accessing earlier phases, in which the variations made respectively. The final output of the project renewed at the end of the Software Development Life Cycle (SDLC) process. Consider an iterative life cycle model which consists of repeating the following phases in sequence;
- Requirements
- Analysis
- Design
- Testing
- Implementation
- Review
- Maintenance

- **Requirements** : A Requirements phase, in which the requirements for the software are gathered and analyzed. Iteration should eventually result in a requirements phase that produces a complete and final specification of requirements.
- **Design phase**: A Design phase, in which a software solution to meet the requirements is designed. This may be a new design, or an extension of an earlier design.
- **Analysis Phases**: In this phase, requirements are gathered from customers and check by an analyst whether requirements will fulfil or not. Analyst checks that need will achieve within budget or not. After all of this, the software team skips to the next phase.
- **Testing Phases**: when the software is coded, integrated and tested.
- **Implementation Phases**: In the implementation, requirements are written in the coding language and transformed into computer programmes which are called Software.
- **Maintenance**: In the maintenance phase, after deployment of the software in the working environment there may be some bugs, some errors or new updates are required. Maintenance involves debugging and new addition options.
- **Review**: A Review phase, in which the software is evaluated, the current requirements are reviewed, and changes and additions to requirements proposed.

CONT...

► When to use 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

- Generates working software quickly and early during the software life cycle.
- More flexible - less costly to change scope and requirements.
- Easier to test and debug during a smaller iteration.
- Easier to manage risk because risky pieces are identified and handled during its iteration.
- Each iteration is an easily managed milestone.

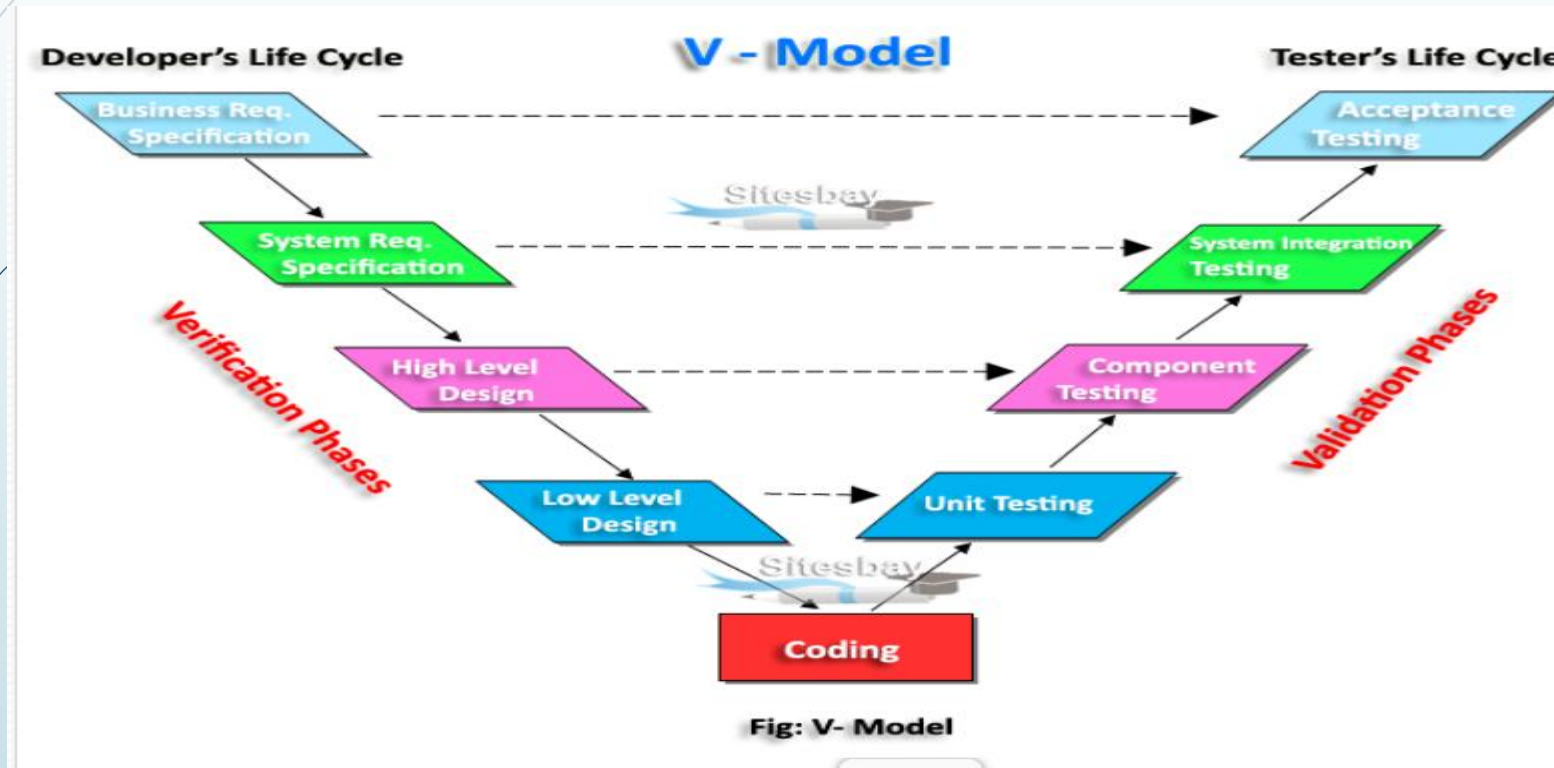
► Disadvantages of Iterative Model

- Each phase of an iteration is rigid and do not overlap each other.
- Problems may arise pertaining to system architecture because not all requirements are gathered up front for the entire software life cycle.

V-MODEL

13

- V- 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.



- The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model.

➤ **V-Model - Verification Phases**

(i) Requirement Analysis

- In this phase, all the required information is gathered & analyzed. Verification activities include reviewing the requirements.

(ii) System Design

- Once the requirement is clear, a system is designed i.e. architecture, components of the product are created and documented in a design document.

(iii) High-Level Design

- High-level design defines the architecture/design of modules. It defines the functionality between the two modules.

(iv) Low-Level Design

- Low-level Design defines the architecture/design of individual components.

(v) Coding

- Code development is done in this phase.

► V-Model - Validation Phases

(i) Unit Testing

15

Unit testing is performed using the unit test cases that are designed and is done in the Low-level design phase. Unit testing is performed by the developer itself. It is performed on individual components which lead to early defect detection.

(ii) Integration Testing

- Integration testing is performed using integration test cases in High-level Design phase. Integration testing is the testing that is done on integrated modules. It is performed by testers.

(iii) System Testing

- System testing is performed in the System Design phase. In this phase, the complete system is tested i.e. the entire system functionality is tested.

(iv) Acceptance Testing

- Acceptance testing is associated with the Requirement Analysis phase and is done in the customer's environment.

► When to use V-Model?

- When the requirement is well defined and not ambiguous.
- The V-shaped model should be used for small to medium-sized projects requirements are clearly defined and fixed.

► Advantages of V – Model

- It is a simple and easily understandable model.
- Testing Methods like planning, test designing happens well before coding.
- Works well for small plans where requirements are easily understood.
- 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 of V-Model

- Very rigid and least flexible.
- Not a good for a complex project.
- V-shaped model is not good for Ongoing projects.
- Requirement change at the later stage would cost too high.

Thank You.....