GROUP 1 PRESENTATION

NAMES	REG NO	SIGN
BARMOT ALEX	SC150/4057/2022	……………..
MILLIAM KARITHI	SC150/0379/2022	………….....
HILLARY KORIR	SC150/0794/2022	…………….
EDWARD MURITHI	SC150/0115/2022	………….......
VIVIAN WAIRIMU	SC150/0095/2022	……………….

System development methodologies

- It's a framework that is used to structure, plan and control the process of developing an information system.
- It provides a step by step approach to ensure that the system is developed efficiently
- It's a formalized, standardized and documented set of activities used to manage system
- development.
- It's also a frame work used to structure, plan and develops systems effectively.
- They are many different system methodologies available and each and its own strengths and weaknesses. Some of the methodologies include:

Types of system development methodologies

I. **STRUCTURED SYSTEM ANALYSIS AND DESIGN METHODOLOGY (SSADM)**

SSADM is a set of standards for system analysis and application design. Its based on the waterfall model.

Characteristics of SSADM

- division of projects into sub processes with well-defined objectives
- activities are performed in a sequence
- object oriented system analysis and design methodology

II. OBJECT ORIENTED SYSTEM ANALYSIS AND DESIGN METHODOLOGIES(OOSADM)

- ✓ Its software engineering approach that models a system as a group of interacting objects.
- ✓ Each object represents some entity of interest in the system being modeled and its
- ✓ characterized by it class, its states and its behavior.
- ✓ Object oriented analysis applies object modeling techniques to analyze the functional
- ✓ requirement of a system

GROUP 1 PRESENTATION

III. RAPID APPLICATION DEVELOPMENT (RAD)

- ✓ It’s a software development methodology that as minimal planning in favor of rapid
- ✓ prototyping
- ✓ Planning of software developed using RAD is interleaved with writing the software itself
- ✓ The lack of extensive preplanning allows the software to be written much

PHASES OF RAD

Requirements planning phase

This phase combines elements of the planning and system analysis phase of the SDLC. User’s managers and ICT staff members discuss and agree on business needs project scope constraints and system requirements.

User design phase

Users interact with system analyst and develop models and prototypes that represents all system processes, inputs and outputs.

Construction phase

This phase focuses on program and application development tasks similar to the SDLC. In RAD however, users continue to participate and still suggest changes or improvement as actual screens, forms or report are developed.

Change over phase

This resembles the final tasks in the SDLC implementation phase ,including data conversion, testing, and change over a new system and user training.

ADVANTAGE OF RAD

o Reduced development time

o Reduced cost

o Better project management

o Greater customer satisfaction

DISAVDAVANTAGES OF RAD

o Requires highly skilled developers designers

o Depend on strong team and individual performance for identifying business requirements

IV. JOINT APPLICATION DEVELOPMENT (JAD)

JAD is methodology that involves the clients or end user in the design and development of an application through a succession of collaborative workshops called JAD sessions.

Advantages of JAD

o Time saving

o Cost saving

o Ownership of the system (meets everyone expectation)

Disadvantages of JAD

o Time commitment

-all JAD participants must be able to meet at designed times.

o JAD commitment

-JAD can only produce effective and productive results if the organization is firmly committed to the approach.

Factors affecting choice of a system development methodology.

1. Well-structured problem situation with well-defined clear requirements can be handled with traditional SDLC approach.
2. Unclear user requirements can be addressed by RAD through prototyping where a model of proposed system can be presented to the user in order to assist in capturing and understanding user requirements.

Software Development Approaches.

- The software development approach is a systematic method framework that is used to structure, plan, and control the process development of an information system.
- Software development approach can also be referred to as the software development lifecycle (SDLC) since it informs about the phases that a software will go through before the final product is achieved.
- There are various software development methodologies that exist which include:
- Waterfall
- Prototyping

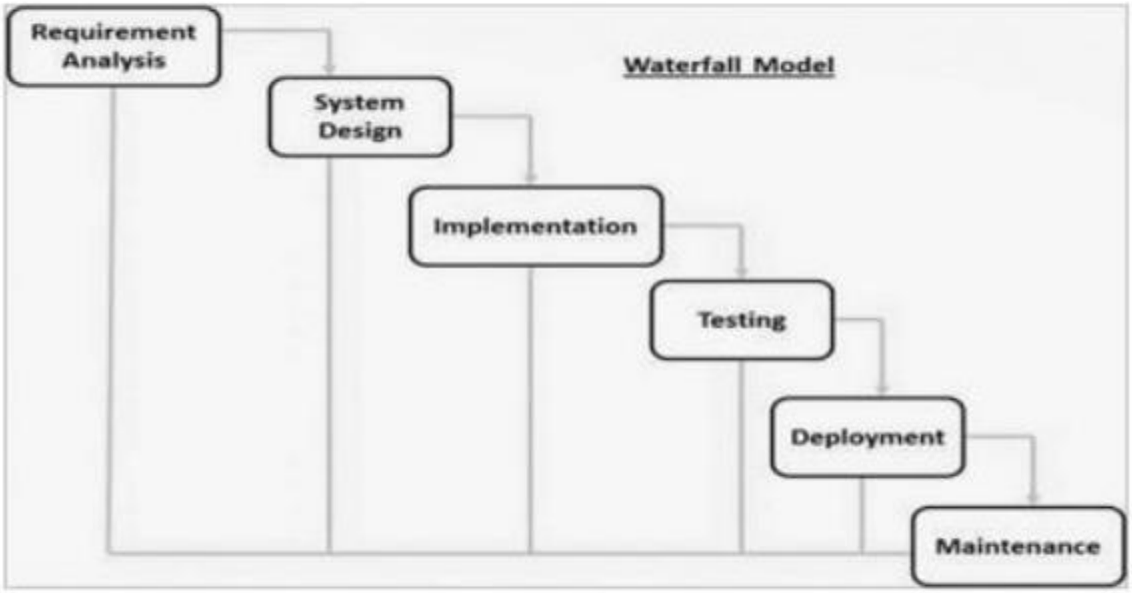
GROUP 1 PRESENTATION

- Agile
- Spiral

Waterfall Model

- The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model.
- It is organized into phases and each phase must be completed before the next phase can begin and there is no overlapping in the phases.
- This is so because in Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.
- All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases.

Stages of waterfall models.



Requirement Gathering and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- **System Design −** The requirement specifications from first phase are studied in this phase and the system design is prepared using UML diagrams. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − With inputs from the system design, the system is first developed (programmed) in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer's environment.

<u>Advantages of Waterfall Model</u>

- Waterfall model is very simple and easy to understand. That is why it is beneficial for the beginner or novice developer.
- It is easy to manage the projects because of the rigidity of the model. Moreover, each phase has specific deliverables and an individual review process.
- This model saves a significant amount of time at all the phases processed and completed at a given time.
- The requirements are very well understood/defined in this type of

development model. Also, it works effectively for smaller projects.

- You can easily do the testing that refers to the defined scenarios in the earlier functional specification.

Disadvantages of Waterfall Model

- Useful only when the requirements are well known and available
- This model is not applicable to projects that demand continuous maintenance.
- Once an application is in the testing stage, it is not advisable to go back and do any amendments changes for completed software, it may cause a lot of problems.
- There is no possibility that we can to develop any working software until it reaches the last stage of the cycle
- You cannot include the client's valuable feedback within the ongoing development phase.
- In this model, Documentation occupies a lot of time for developers and testers.

Prototyping Methodology

- It is iterative in nature since it involves developing a prototype (crude version of the software) then giving it to the users for review who after review either accept it and recommend adjustments or reject it.
- Unlike Water fall model and spiral model prototyping makes it

possible to develop software that are acceptable by customers.

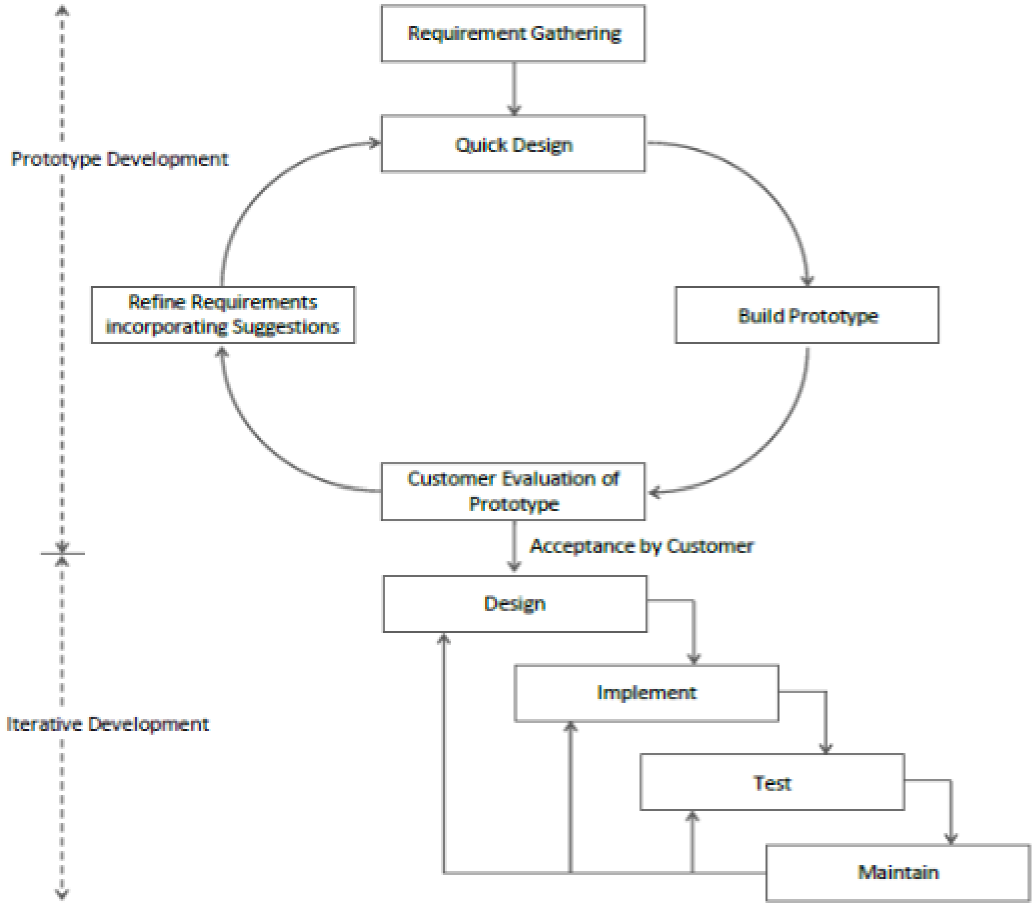


Figure 3: Prototype Model of Software Development

Types of Prototypes

• **Rapid throwaway prototype-** In this method, a developed prototype need not necessarily be a part of the ultimately accepted prototype. The initial prototype is discarded after getting customer feedback. Customer feedback helps in preventing unnecessary design faults and hence, the final prototype developed is of better quality.

• **Evolutionary Prototyping** – In this method, the prototype developed initially is incrementally refined on the basis of customer feedback till it finally gets accepted.

• In comparison to Rapid Throwaway Prototyping, it offers a better approach which saves time as well as effort. This is because developing a prototype from scratch for every iteration of the process can sometimes be very frustrating for the developers.

GROUP 1 PRESENTATION

- **Incremental Prototyping** – In this type of incremental Prototyping, the final expected product is broken into different small pieces of prototypes and each piece is developed individually.
- In the end, when all individual pieces are properly developed, then the different prototypes are collectively merged into a single final product in a predefined order.
- It’s a very efficient approach which reduces the complexity of the development process, where the goal is divided into sub-parts and each sub-part is developed individually.

Advantages of Prototyping

- The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
- New requirements can be easily accommodated as there is scope for refinement.
- Missing functionalities can be easily figured out.
- Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.
- The developed prototype can be reused by the developer for more complicated projects in the future.
- Flexibility in design.

Disadvantages of Prototyping

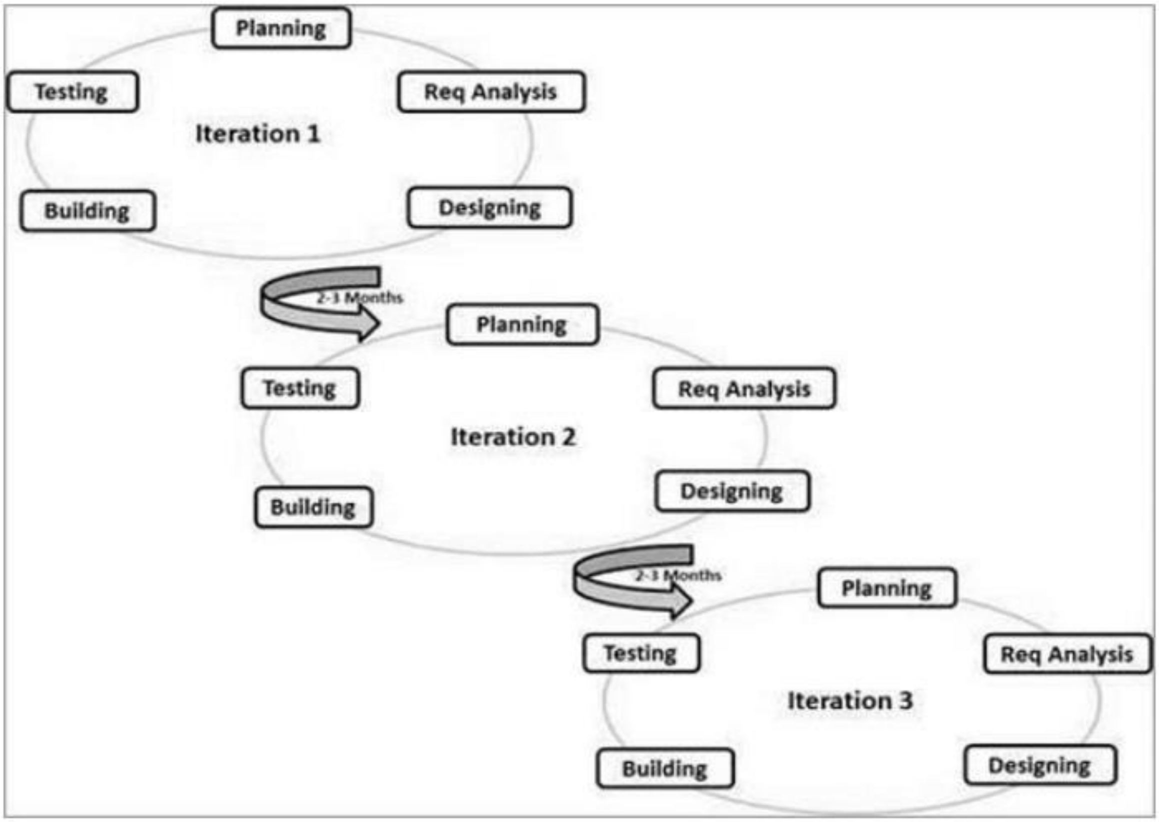
- Costly with respect to time as well as money.
- There may be too much variation in requirements each time the prototype is evaluated by the customer.
- Poor Documentation due to continuously changing customer requirements.
- It is very difficult for developers to accommodate all the changes demanded by the customer.
- There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted by the customer.
- After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.
- Developers in a hurry to build prototypes may end up with sub-optimal solutions.
- The customer might lose interest in the product if he/she is not satisfied with the

initial prototype.

Agile Software Development Approach.

- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile Methods break the product into small incremental builds. These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like −
- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing.
- At the end of the iteration, a working product is displayed to the customer and important stakeholders.
- Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.
- In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release.
- Iterative approach is taken and working software build is delivered after each iteration.
- Each build is incremental in terms of features; the final build holds all the features required by the customer.

GROUP 1 PRESENTATION



Agile Principles

Agile has 12 key principals in its manifesto. The principals are:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design

enhances agility.
- Simplicity–the art of maximizing the amount of work not done–is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Advantages of Agile Model
- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Suitable for fixed or changing requirements
- Delivers early partial working solutions.
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.
- Gives flexibility to developers.

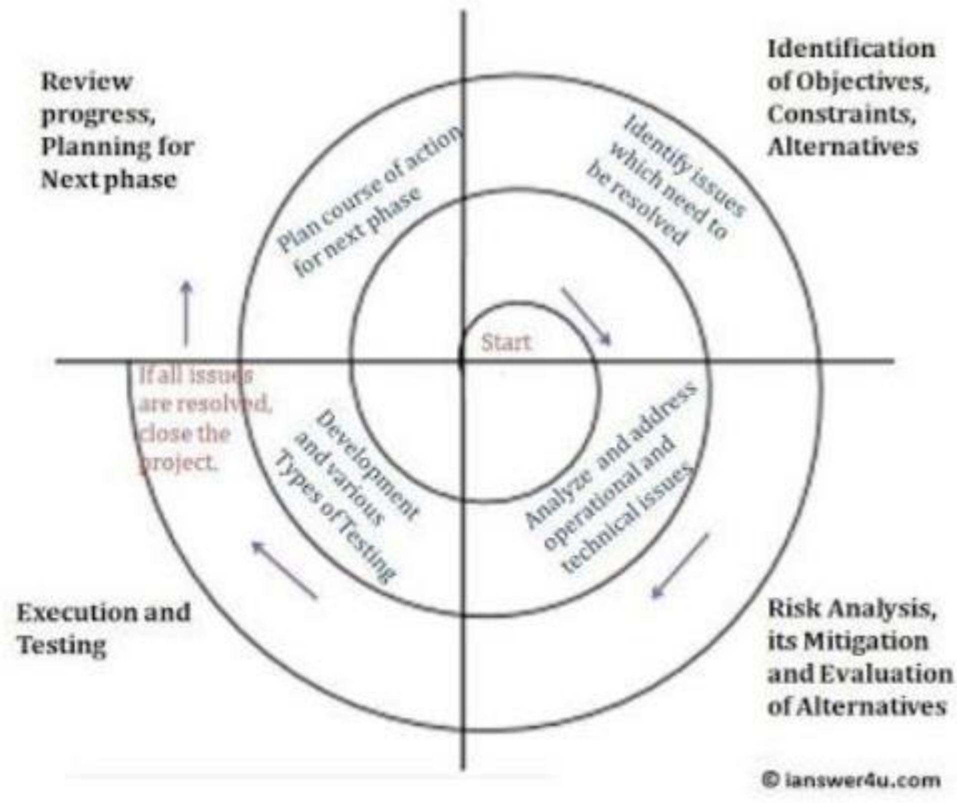
Disadvantages of Agile model

- More risk of sustainability, maintainability and extensibility.
- An overall plan, an agile leader and agile project manager practice is a must without which it will not work.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is a very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.
- There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted by the customer.
- After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.
- Developers in a hurry to build prototypes may end up with sub-optimal solutions.
- The customer might lose interest in the product if he/she is not satisfied with the initial

prototype.

Spiral Model

Spiral Model is a risk-driven software development process model. It is a combination of waterfall model and iterative model. Spiral Model helps to adopt software development elements of multiple process models for the software project based on unique risk patterns ensuring efficient development process. Each phase of spiral model in software engineering begins with a design goal and ends with the client review the progress.


Review progress, Planning for Next phase
Identification of Objectives, Constraints, Alternatives
Plan course of action for next phase
Identify issues which need to be resolved
Start
If all issues are resolved, close the project.
Development and various Types of Testing
Analyze and address operational and technical issues
Execution and Testing
Risk Analysis, its Mitigation and Evaluation of Alternatives
© ianswer4u.com

Spiral Model Diagram

Spiral Model Phases

Spiral Model Phases Activities performed during phase

Planning - It includes estimating the cost, schedule and resources for the iteration. It also involves understanding the system requirements for continuous communication between the system analyst and the customer.

GROUP 1 PRESENTATION

Risk Analysis - Identification of potential risk is done while risk mitigation strategy is planned and finalized.

Engineering - It includes testing, coding and deploying software at the customer site

Evaluation -Evaluation of software by the customer. Also, includes identifying and monitoring risks such as schedule slippage and cost overrun

When to use Spiral Model?

a) A Spiral model in software engineering is used when project is large.

b) When releases are required to be frequent, spiral methodology is used.

c) When creation of a prototype is applicable.

d) When risk and costs evaluation is important.

e) Spiral methodology is useful for medium to high-risk projects.

f) When requirements are unclear and complex, Spiral model in SDLC is useful.

g) When changes may require at any time.

h) When long term project commitment is not feasible due to changes in economic priorities.

Spiral Model Advantages

- Additional functionality or changes can be done at a later stage.
- Cost estimation becomes easy as the prototype building is done in small fragments.
- Continuous or repeated development helps in risk management.
- Development is fast and features are added in a systematic way in Spiral development.

Disadvantages

- Risk of not meeting the schedule or budget.
- Spiral development works best for large projects only also demands risk assessment expertise.
- For its smooth operation spiral model protocol needs to be followed strictly.

- Documentation is more as it has intermediate phases.

V-model

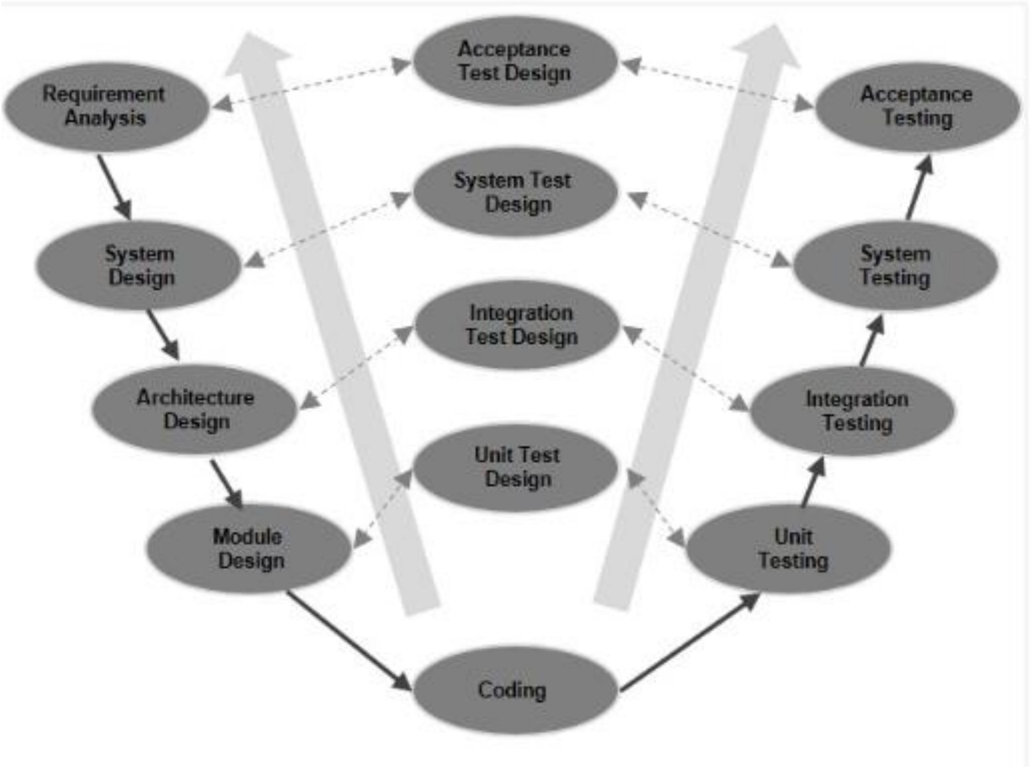
The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

V-Model - Design

Under the V-Model, the corresponding testing phase of the development phase is planned in parallel. So, there are Verification phases on one side of the ‘V’ and Validation phases on the other side. The Coding Phase joins the two sides of the V-Model.

The following illustration depicts the different phases in a V-Model of the SDLC.



Business Requirement Analysis

GROUP 1 PRESENTATION

This is the first phase in the development cycle where the product requirements are understood from the customer's perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement. This is a very important activity and needs to be managed well, as most of the customers are not sure about what exactly they need.
The **acceptance test design planning** is done at this stage as business requirements can be used as an input for acceptance testing.

System Design

Once you have the clear and detailed product requirements, it is time to design the complete system. The system design will have the understanding and detailing the complete hardware and communication setup for the product under development. The system test plan is developed based on the system design. Doing this at an earlier stage leaves more time for the actual test execution later.

Architectural Design

Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. The system design is broken down further into modules taking up different functionality. This is also referred to as **High Level Design (HLD)**.

The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage. With this information, integration tests can be designed and documented during this stage.

Module Design

In this phase, the detailed internal design for all the system modules is specified, referred to as **Low Level Design (LLD)**. It is important that the design is compatible with the other modules in the system architecture and the other external systems. The unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. These unit tests can be designed at this stage based on the internal module designs.

Coding Phase

The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements.

The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

Validation Phase

In this phase different system testing are done; They include unit testing, integration testing, system testing and acceptance testing.

The following areas are some of the most suitable scenarios to use the V-Model application.

- Requirements are well defined, clearly documented and fixed.
- Product definition is stable.
- Technology is not dynamic and is well understood by the project team.
- There are no ambiguous or undefined requirements.
- The project is short.

The advantages of the V-Model method are as follows −

- This is a highly-disciplined model and Phases are completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.

The disadvantages of the V-Model method are as follows :

- 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.

GROUP 1 PRESENTATION