Perspective and Specialized Process Models

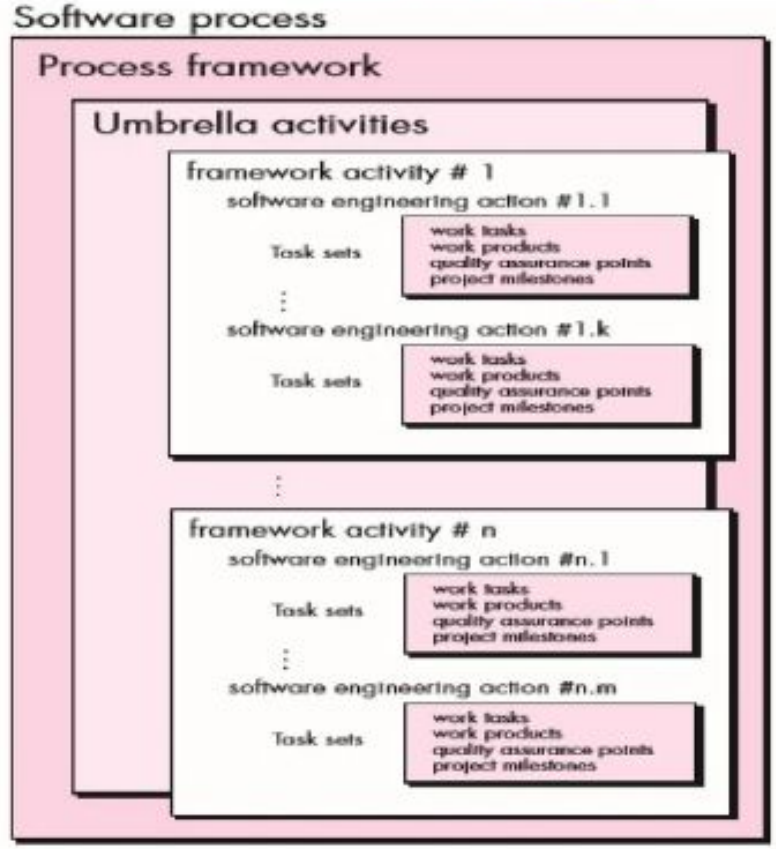
A GENERIC PROCESS MODEL

- A process was defined as a collection of work activities, actions, and tasks that are performed when some work product is to be created. Each of these activities, actions, and tasks reside within a framework or model that defines their relationship with the process and with one another.

- The software process is represented schematically in Figure 2.1. Referring to the figure, each framework activity is populated by a set of software engineering actions.

- Each software engineering action is defined by a task set that identifies the work tasks that are to be completed, the work products that will be produced, the quality assurance points that will be required, and the milestones that will be used to indicate progress.

FIGURE 2.1
A software process framework

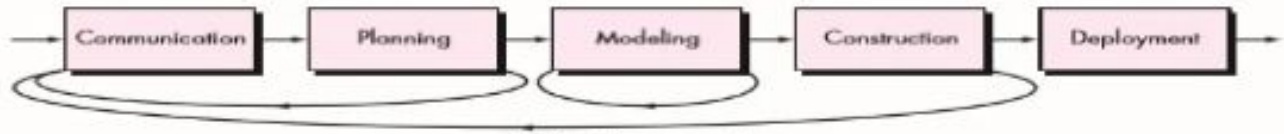


- A generic process framework for software engineering defines five framework activities—communication, planning, modeling, construction, and deployment. In addition, a set of umbrella activities—project tracking and control, risk management, quality assurance, configuration management, technical reviews, and others—are applied throughout the process.

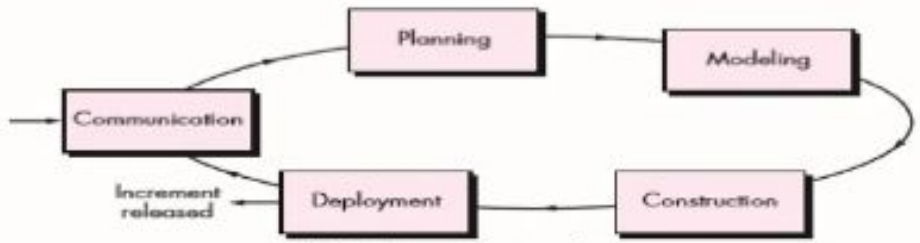
- A linear process flow executes each of the five framework activities in sequence, beginning with communication and culminating with deployment (Figure 2.2a).
- An iterative process flow repeats one or more of the activities before proceeding to the next (Figure 2.2b).
- An evolutionary process flow executes the activities in a "circular" manner. (Figure 2.2c).
- A parallel process flow (Figure 2.2d) executes one or more activities in parallel with other activities.

FIGURE 2.2 Process flow

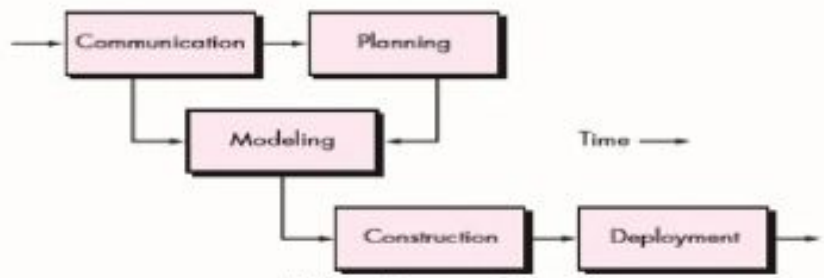
(a) Linear process flow



(b) Iterative process flow



(c) Evolutionary process flow



(d) Parallel process flow

Typical umbrella activities include:

Software project tracking and control—allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.

Risk management—assesses risks that may affect the outcome of the project or the quality of the product.

Software quality assurance—defines and conducts the activities required to ensure software quality

Technical reviews—assesses software engineering work products in an effort to uncover and remove errors before they are propagated to the next activity.

Measurement—defines and collects process, project, and product measures that assist the team in delivering software that meets stakeholders' needs; can be used in conjunction with all other framework and umbrella activities.

Software configuration management—manages the effects of change throughout the software process.

Reusability management—defines criteria for work product reuse and establishes mechanisms to achieve reusable components.

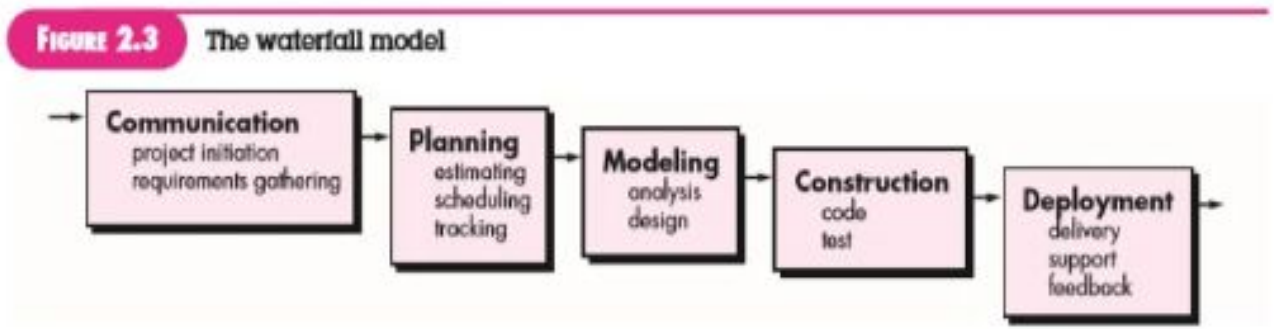
Work product preparation and production—encompasses the activities required to create work products such as models, documents, logs, forms, and lists.

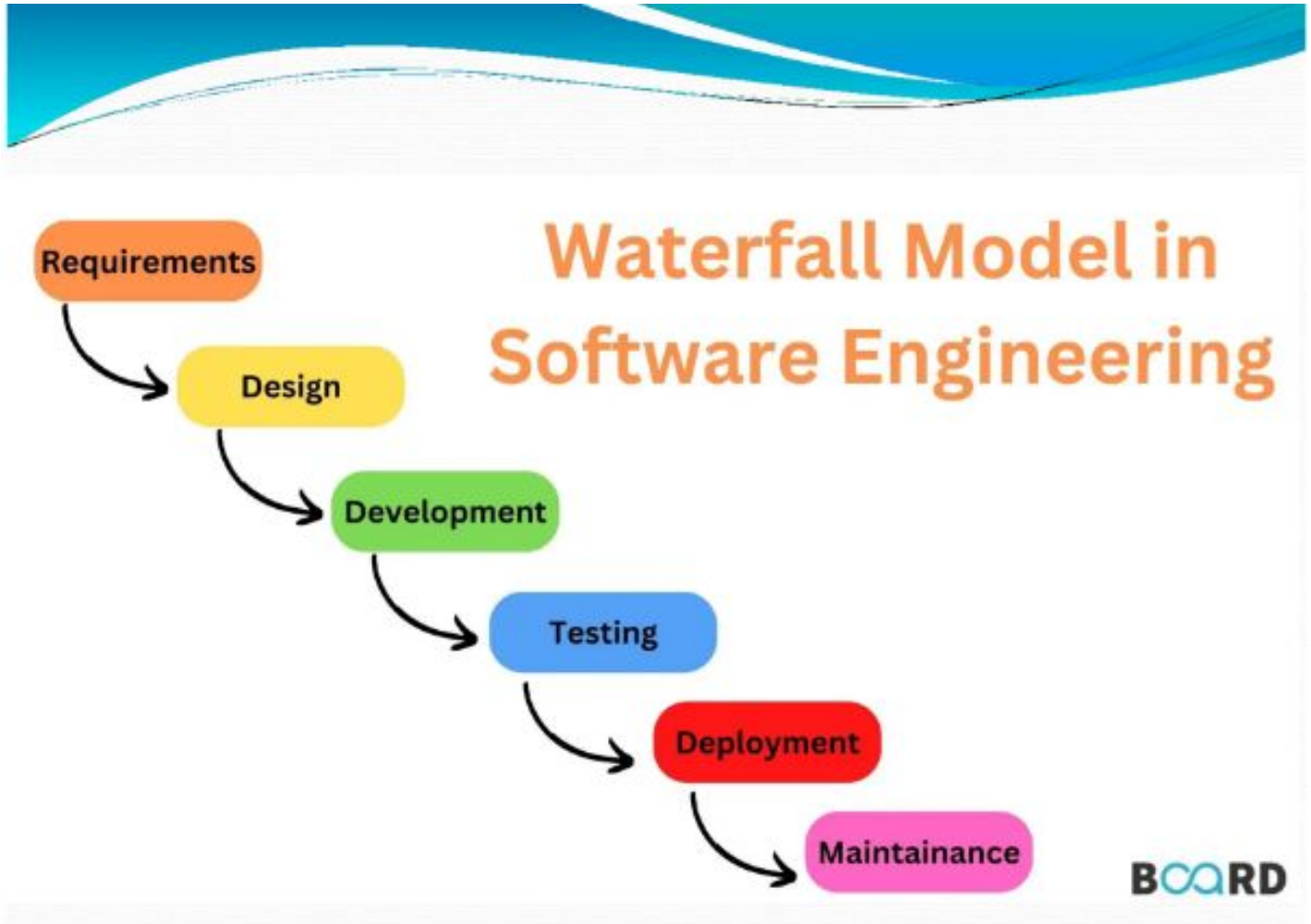
PRESCRIPTIVE PROCESS MODELS

- The Waterfall Model
- Spiral Model
- V Shaped Model
- RAD Model
- Iterative Model
- Prototype Model

The Waterfall Model

- The waterfall model, sometimes called the classic life cycle, suggests a systematic, sequential approach to software development that begins with customer specification of requirements and progresses through planning, modeling, construction, and deployment. Linear-sequential model or classic life cycle model.




Waterfall Model in Software Engineering
Requirements
Design
Development
Testing
Deployment
Maintainance
BOARD

Features of waterfall model:

- It is very simple to understand and use.
- In a waterfall model, each phase must be completed fully before the next phase can begin.
- This type of model is basically used for the for the project which is small and there are no uncertain requirements.
- At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project.
- Testing phase starts only after the development is complete.
- In waterfall model phases do not overlap.
- Documentation is produced in each and every step

Advantages of waterfall model-

- This model works for small projects because the requirements are understood very well.
- The waterfall model is simple and easy to understand, implement, and use.
- All the requirements are known at the beginning of the project, hence it is easy to manage.
- Documentation helps for future reference

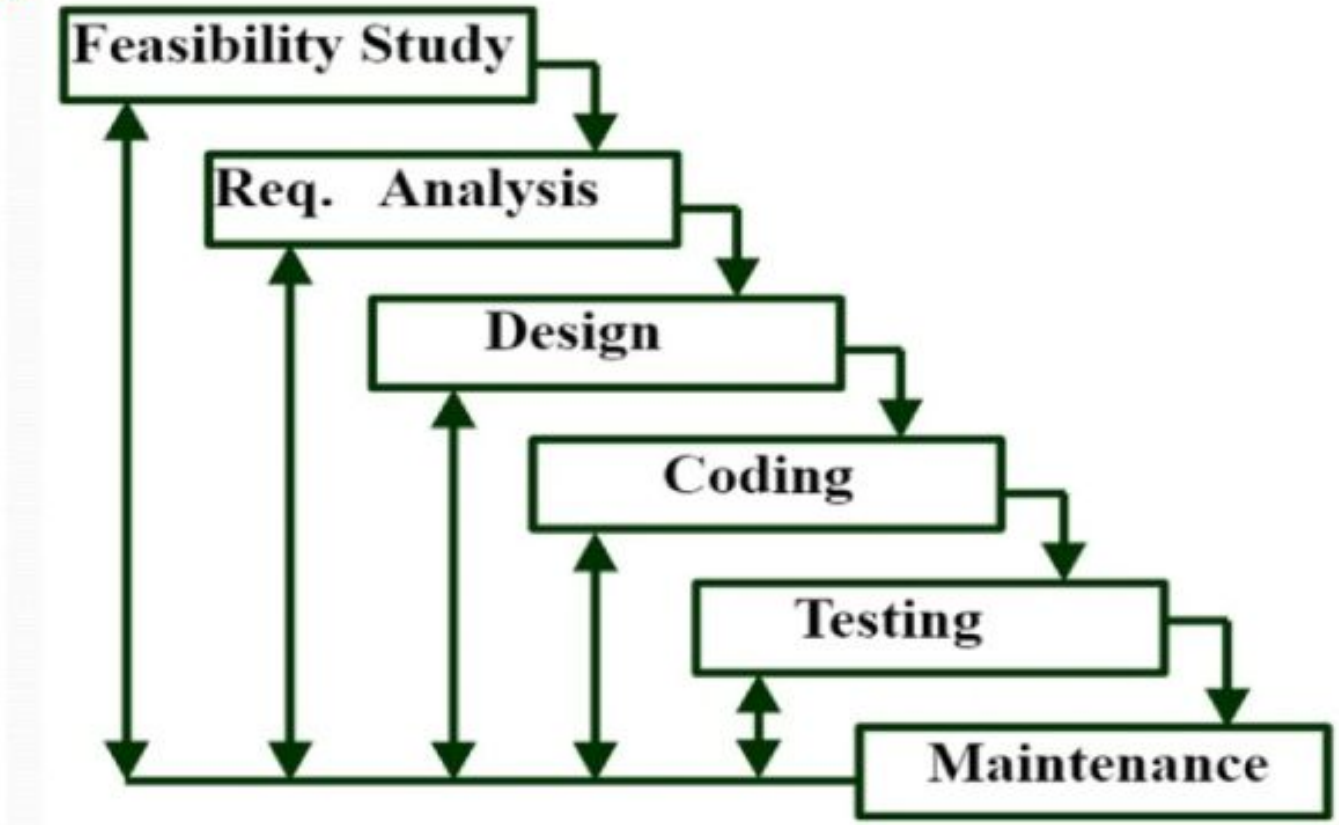
Disadvantages of the waterfall model

- The problems with this model are uncovered, until the software testing.
- Requirement analysis is done initially, sometime its nit possible to state all requirements initially.
- The amount of risk is high.
- This model is not good for complex and object oriented projects.
- Customer can see the model only at the end

When to use the waterfall model:

- This model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable.
- Technology is understood.
- There are no ambiguous requirements
- The project is short.

Iterative Waterfall Model



The Spiral model

- Spiral model is a risk driven process model.
- Spiral model is iterative in nature
- In spiral model, an alternate solution is provided if the risk is found in the risk analysis, then alternate solutions are suggested and implemented.
- It is a combination of prototype and sequential model or waterfall model.
- In one iteration all activities are done, for large project's the output is small.
- The framework activities of the spiral model are as shown in the following figure.

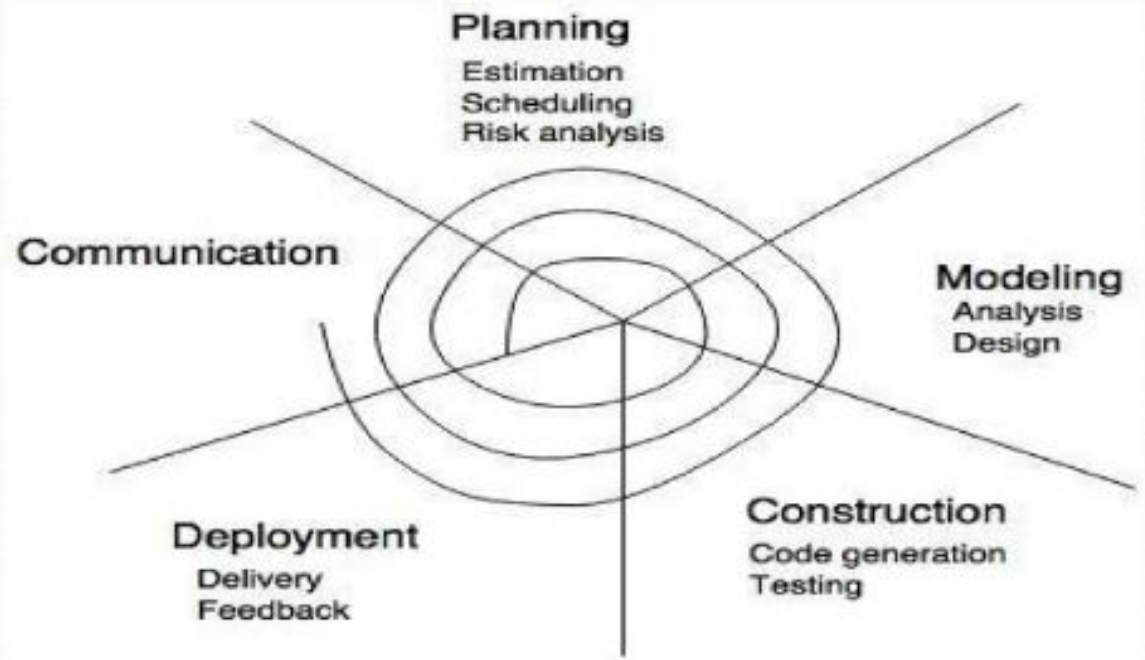


Fig. - The Spiral Model

- The spiral model is similar to the incremental model, with more emphasis placed on risk analysis.
- The spiral model has **four phases: Planning, Risk Analysis, Engineering and Evaluation.**
- A software project repeatedly passes through these phases in iterations (called Spirals in this model).
- The baseline spiral, starting in the planning phase, requirements is gathered and risk is assessed.
- Each subsequent spiral builds on the baseline spiral.

Planning Phase:

Requirements are gathered during the planning phase.
Requirements like 'BRS' that is **'Business Requirement Specifications'** and SRS' that is **System Requirement specifications'.**

Risk Analysis:

In the risk analysis phase, a process is undertaken to identify risk and alternate solutions.
A prototype is produced at the end of the risk analysis phase.
If any risk is found during the risk analysis then alternate solutions are suggested and implemented.

Engineering Phase:

In this phase software is developed, along with testing at the end of the phase.
Hence in this phase the development and testing is done.

Evaluation phase:

This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

Advantages of Spiral Model

- It reduces high amount of risk.
- It is good for large and critical projects.
- It gives strong approval and documentation control.
- Requirement changes can be made easily at every stage.
- Additional Functionality can be added at a later date.
- In spiral model, the software is produced early in the life cycle process.

Disadvantages of Spiral Model

- It can be costly to develop a software model.
- It is not used for small projects.
- Risk analysis requires highly specific expertise.
- Project‘s success is highly dependent on the risk analysis phase

When to use Spiral model:

- When costs and risk evaluation is important
- When project is not expected within a specific limited time span
- For medium to high-risk projects
- Users are unsure of their needs
- Requirements are complex.
- Significant changes are expected (research and exploration)

V-Shaped Model

V- Shaped Model means Verification and Validation model. A variation in the representation of the waterfall model is called the V-Shaped Model.

Just like the waterfall model, the V-Shaped life cycle is a sequential path of execution of processes.

Each phase must be completed before the next phase begins.

Testing of the product is planned in parallel with a corresponding phase of development

- As a software team moves down the left side of the V, basic problem requirements are refined into progressively more detailed and technical representations of the problem and its solution.

- Once code has been generated, the team moves up the right side of the V, essentially performing a series of tests (quality assurance actions) that validate each of the models created as the team moved down the left side.

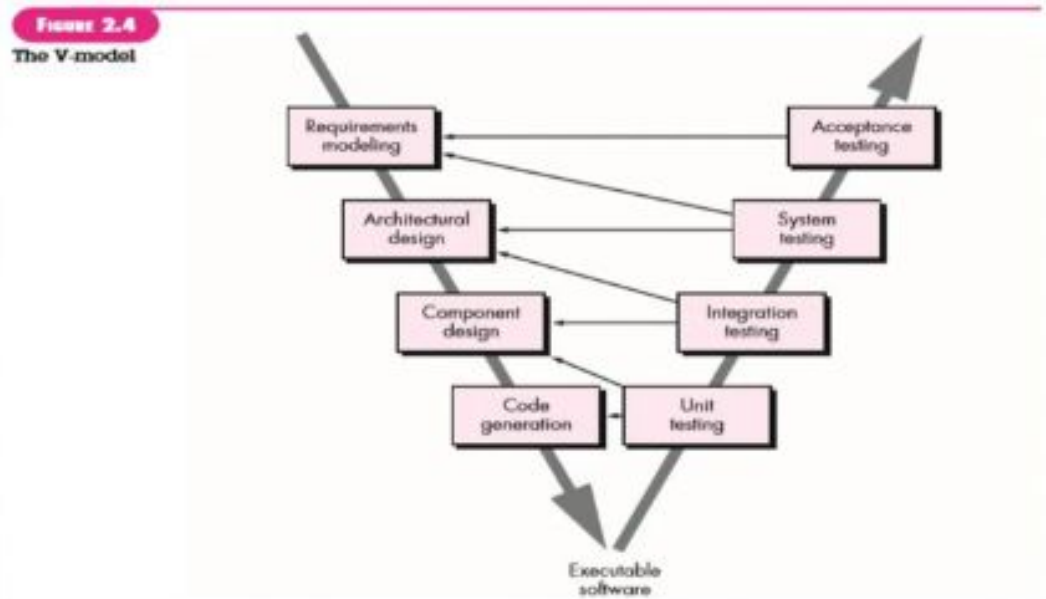


FIGURE 2.4
The V-model

Phases of V-Model

Requirements

- Requirements like BRS and SRS begin the life cycle model just like the waterfall model.
- But, in this model before development is started, a system test plan is created.
- The test plan focuses on meeting the functionality specified in the requirements gathering.

The high-level design (HLD)

- This phase focuses on system architecture and design.
- It provides overview of solution, platform, system, product and service/process.
- An integration test plan is created in this phase as well in order to test the pieces of the software systems ability to work together.

Phases of V-Model

The low-level design (LLD)

- This phase is where the actual software components are designed.
- It defines the actual logic for each and every component of the system.
- Class diagram with all the methods and relation between classes comes under LLD.
- Component tests are created in this phase as well.

Implementation

- In this phase , all coding takes place.
- Once coding is complete, the path of execution continues up the right side of the V where the test plans developed earlier are now put to use.

Coding

- This is at the bottom of the V-Shape model.
- Module design is converted into code by developers.

Advantages of V-model

- Simple and easy to use.
- Testing activities like planning, test designing happens well before coding. This saves a lot of time. Hence higher chance of success over the waterfall model.
- Proactive defect tracking – that is defects are found at early stage.
- Avoids the downward flow of the defects.
- Works well for small projects where requirements are easily understood.

Disadvantages of V-model

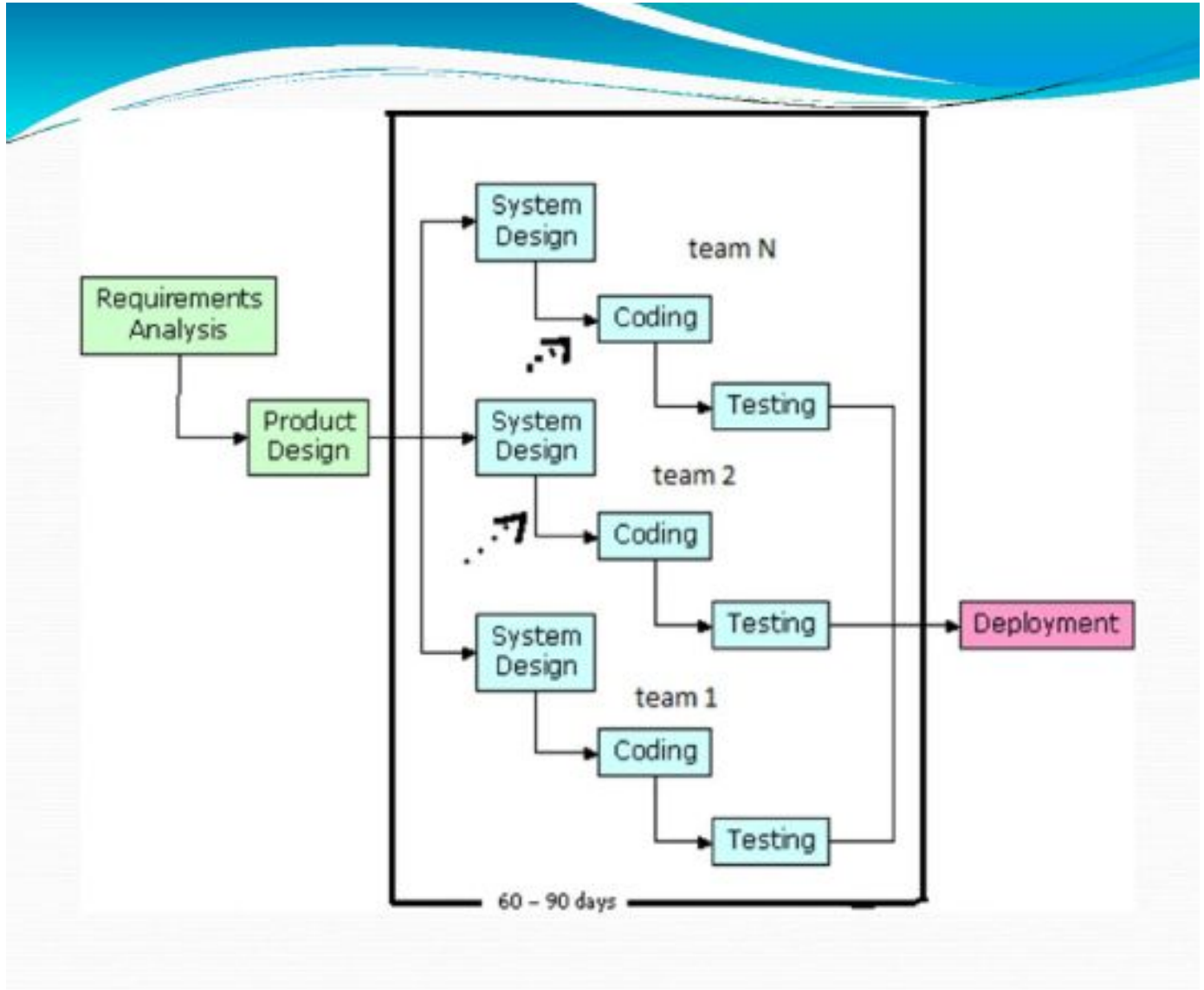
- Very rigid and least flexible.
- Software is developed during the implementation phase, so no early prototypes of the software are produced.
- If any changes happen in midway, then the test documents along with requirement documents has to be updated

When to use the V-model

- The V-shaped model should be used for small to medium sized projects where requirements are clearly defined and fixed.
- The V-Shaped model should be chosen when ample technical resources are available with needed technical expertise.
- High confidence of customer is required for choosing the V-Shaped model approach. Since, no prototypes are produced, there is a very high risk involved in meeting customer expectations.

RAD Model/(RAPID APPLICATION DEVELOPMENT MODEL)

- RAD model is a type of incremental process model in which there is extremely short development cycle.
- 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.
- This can quickly give the customer something to see and use and to provide feedback regarding the delivery and their requirements.


Requirements Analysis
Product Design
System Design
Coding
Testing
team N
System Design
Coding
Testing
team 2
System Design
Coding
Testing
team 1
Deployment
60 – 90 days

Features of RAD model

- Using a RAD Model, a software product can be developed within 60 to 90 days of time.

- Various phases of RAD model are

  - Requirement gathering,
  - Planning,
  - Analysis,
  - Design and
  - Deployment

When to use RAD model

- When there is a need to create a system 2-3 months of time.
- Only when the resources with high business knowledge are available

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.

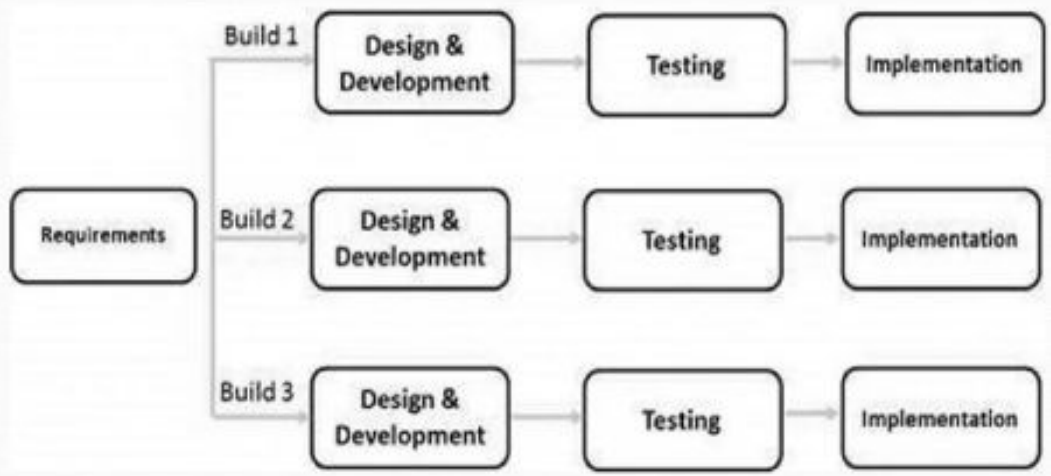
Disadvantages of RAD Model:

- Requires highly skilled developers/designers.
- Requires multiple teams and large number of people to work on a project.
- Need Heavy resources
- If there is no proper modularization, then RAD project would fail.
- Adopting new technologies is difficult.

Iterative Process Models

- The iterative development model develops a system by **building small portions** of all the features. This helps to meet the initial scope quickly and release it for feedback.
- In the iterative model, you start off by implementing a small set of software requirements. These are then **enhanced iteratively** in the evolving versions until the system is completed. This process model starts with part of the software, which is then implemented and reviewed to identify further requirements.

In this Iterative model, the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.



This model is most often used in the following scenarios −

- Requirements of the complete system are clearly defined and understood.
- Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
- There is a time to the market constraint.
- A new technology is being used and is being learnt by the development team while working on the project.
- Resources with needed skill sets are not available and are planned to be used on contract basis for specific iterations.
- There are some high-risk features and goals which may change in the future.

Advantages of Iterative model

- The advantages of the and Incremental SDLC Model are as follows −
- Some working functionality can be developed quickly and early in the life cycle.
- Results are obtained early and periodically.
- Parallel development can be planned.
- Progress can be measured.
- Less costly to change the scope/requirements.
- Testing and debugging during smaller iteration is easy.
- Risks are identified and resolved during iteration; and each iteration is an easily managed milestone.
- Easier to manage risk - High risk part is done first.
- With every increment, operational product is delivered.

Advantages of Iterative model

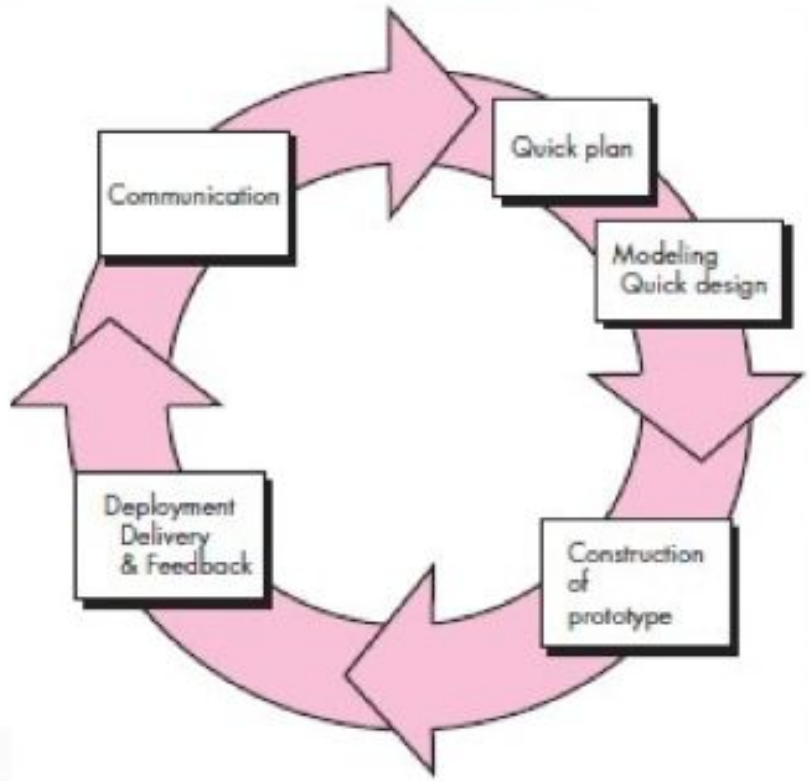
- Issues, challenges and risks identified from each increment can be utilized/applied to the next increment.
- Risk analysis is better.
- It supports changing requirements.
- Initial Operating time is less.
- Better suited for large and mission-critical projects.
- During the life cycle, software is produced early which facilitates customer evaluation and feedback.

The Prototyping model

- Prototype is defined as first or preliminary form using which other forms are copied or derived.
- Prototype model is a set of general objectives for software.
- It does not identify the requirements like detailed input, output.
- It is software working model of limited functionality.
- In this model, working programs are quickly produced.
- It is difficult to do modifications in earlier development phases of linear model. In such cases Iterative approach is adopted.
- The evolutionary process model is iterative model

Features of Prototyping model

- The basic idea here is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements.
- This prototype is developed based on the currently known requirements.
- By using this prototype, the client can get an actual feel of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system.
- Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements.
- The prototype is usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.


Communication
Quick plan
Modeling
Quick design
Construction
of
prototype
Deployment
Delivery
& Feedback

The different phases of Prototyping model are:

1.Communication

In this phase, developer and customer meet and discuss the overall objectives of the software.

2.Quick design

Quick design is implemented when requirements are known. It includes only the important aspects like input and output format of the software. It focuses on those aspects which are visible to the user rather than the detailed plan. It helps to construct a prototype.

3.Modeling quick design

This phase gives the clear idea about the development of software because the software is now built. It allows the developer to better understand the exact requirements.

The different phases of Prototyping model are:

4.Construction of prototype

The prototype is evaluated by the customer itself.

5.Deployment, delivery, feedback

If the user is not satisfied with current prototype then it refines according to the requirements of the user.The process of refining the prototype is repeated until all the requirements of users are met. When the users are satisfied with the developed prototype then the system is developed on the basis of final prototype.

Advantages of Prototyping Model

- Prototype model need not know the detailed input, output, processes, adaptability of operating system and full machine interaction.
- In the development process of this model users are actively involved.
- The development process is the best platform to understand the system by the user.
- Errors are detected much earlier.
- Gives quick user feedback for better solutions.
- It identifies the missing functionality easily. It also identifies the confusing or difficult functions.

Disadvantages of Prototyping Model:

- The client involvement is more and it is not always considered by the developer.
- It is a slow process because it takes more time for development.
- Many changes can disturb the rhythm of the development team.
- It is a thrown away prototype when the users are confused with it.

When to use Prototype model:

Prototype model should be used when the desired system needs to have a lot of interaction with the end users.

- Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model.
- It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.
- Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system.
- They are excellent for designing good human computer interface systems.

Disadvantages of Iterative model

- More resources may be required.
- Although cost of change is lesser, but it is not very suitable for changing requirements.
- More management attention is required.
- System architecture or design issues may arise because not all requirements are gathered in the beginning of the entire life cycle.
- Defining increments may require definition of the complete system.
- Not suitable for smaller projects.
- Management complexity is more.
- End of project may not be known which is a risk.
- Highly skilled resources are required for risk analysis.
- Projects progress is highly dependent upon the risk analysis phase.

Selection of a Life Cycle Model

Selection of a model is based on:

a) Requirements

b) Development team

c) Users

d) Project type and associated risk

Based On Characteristics Of Requirements

Requirements	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Are requirements easily understandable and defined?	Yes	No	No	No	No	Yes
Do we change requirements quite often?	No	Yes	No	No	Yes	No
Can we define requirements early in the cycle?	Yes	No	Yes	Yes	No	Yes
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes	Yes	No

Based On Status Of Development Team

Development team	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
Less experience on similar projects?	No	Yes	No	No	Yes	No
Less domain knowledge (new to the technology)	Yes	No	Yes	Yes	Yes	No
Less experience on tools to be used	Yes	No	No	No	Yes	No
Availability of training if required	No	No	Yes	Yes	No	Yes

Based On User's Participation

Involvement of Users	Waterfall	Prototype	Iterative enhancement	Evolutionary development	Spiral	RAD
User involvement in all phases	No	Yes	No	No	No	Yes
Limited user participation	Yes	No	Yes	Yes	Yes	No
User have no previous experience of participation in similar projects	No	Yes	Yes	Yes	Yes	No
Users are experts of problem domain	No	Yes	Yes	Yes	No	Yes