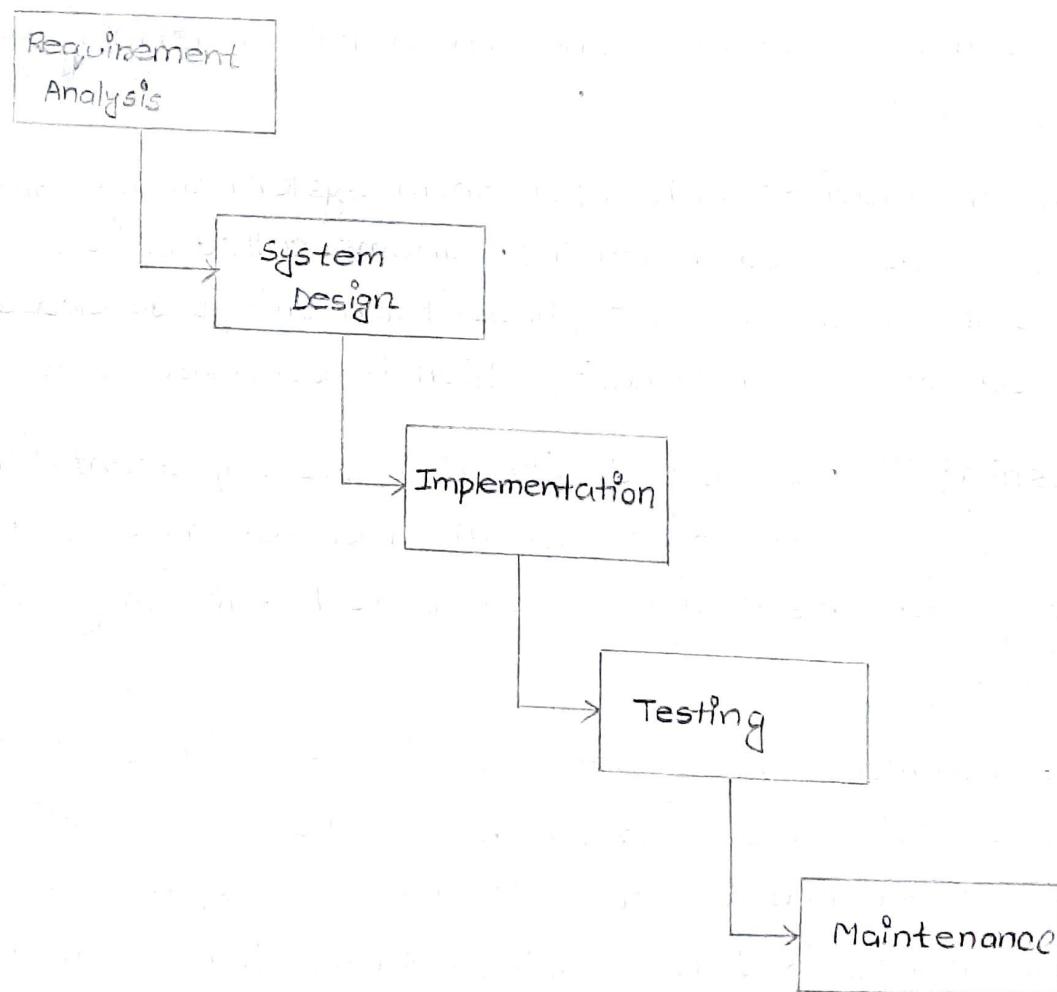


Describe waterfall model.

The Waterfall Model is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

In the waterfall approach the whole process of software development is divided into separate phases. In this model, the outcome of one phase acts as the input for the next phase sequentially.



Requirement Analysis: The complete service documentation will be prepared in this stage by the developers who will be consulting with the client. The clients will provide their necessities along with constraints and flow of work. The developers will analyze the given information to create the documentation both for developers and for the clients. The next stage starts with this approved document.

System Design: The requirement specification document from first phase are studied in this phase and system design is prepared. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

Implementation: With inputs from system design, the system is first developed 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.

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 failure.

Maintenance: There are some issues which came 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 environment.

Application :

Every software developed is different and requires a suitable model to be followed based on the internal and external factors. Some situations where the use of waterfall model is most appropriate are :

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

What are the advantages and disadvantages of waterfall model?

Advantages / Pros :

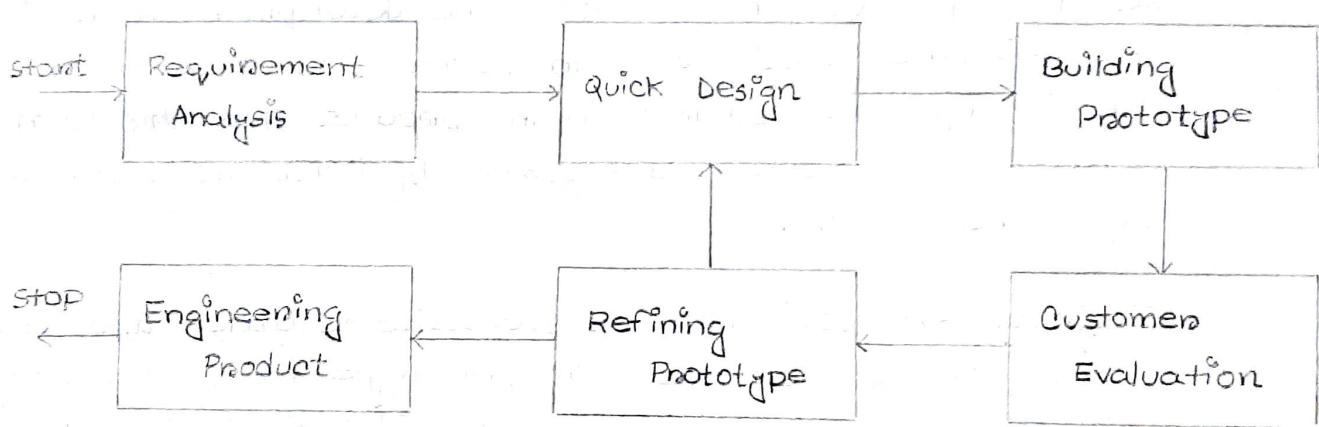
1. Simple and easy to understand and use.
2. Easy to manage due to the rigidity of the model - each phase has specific deliverables and a review process.
3. Phases are processed and completed one at a time.
4. Works well for smaller projects where requirements are very well understood.
5. Clearly defined stages.
6. Well understood milestones.
7. Easy to arrange tasks.
8. Process and results are well documented.

Disadvantages / Cons:

1. No working software is produced until late during the life cycle.
2. High amounts of risk and uncertainty.
3. Not a good model for complex and object-oriented projects.
4. Poor model for long and ongoing projects.
5. Not suitable for the projects where requirements are at a moderate to high risk of changing. So risk and uncertainty is high with this process model.
6. It is difficult to measure progress within stages.
7. Cannot accommodate changing requirements.
8. Adjusting scope during the life cycle can end a project.

Describe the prototyping model.

Often, a customer defines a set of general objectives for software but does not identify detailed input, processing or output requirements. In other cases, the developer may be unsure of the efficiency of an algorithm, the adaptability of an operating system or the form that human/machine interaction should take. In these and many other situations a prototyping paradigm may offer the best approach.



Requirement Analysis: Firstly, developers and customers meet and define the overall objectives, requirements and outline areas where further definition is mandatory.

Quick Design: In this phase the very basic requirements are showed and user interfaces are provided. These features used in quick design may not exactly work in the same manner as the actual software developed.

Building Prototypes: In this phase it takes the output from the quick design phase and build that system design locally. This

local building prototype gives a look and feel similar as the final product has to be designed.

Customer Evaluation: The customer checks the working functionality of the build prototype and gives a feedback to the development team. These feedback from customer and stakeholders of the product is collected in an organized manner and used for the further enhancements in the product under development phase.

Refining Product: In this phase the development team discusses the customers feedback and reviews about the build prototype and if any issue occurred in the product then the team tries to refine the project and successively fulfill the customers desire about the system.

Engineer Product: After all successive feedback and positive review from the customers the build prototype take over by the software design and development team. The actual product is designed and developed in this phase.

When to use prototype model:

- Whenever the customer not clear about the requirement the developers go for prototype model.
- If it is complex project then prototype model makes clear understand about the requirement.
- Prototyping makes sure that the customer constantly works with the system and provide a feedback about the system.

What are the advantages and disadvantages of prototype model?

Advantages:

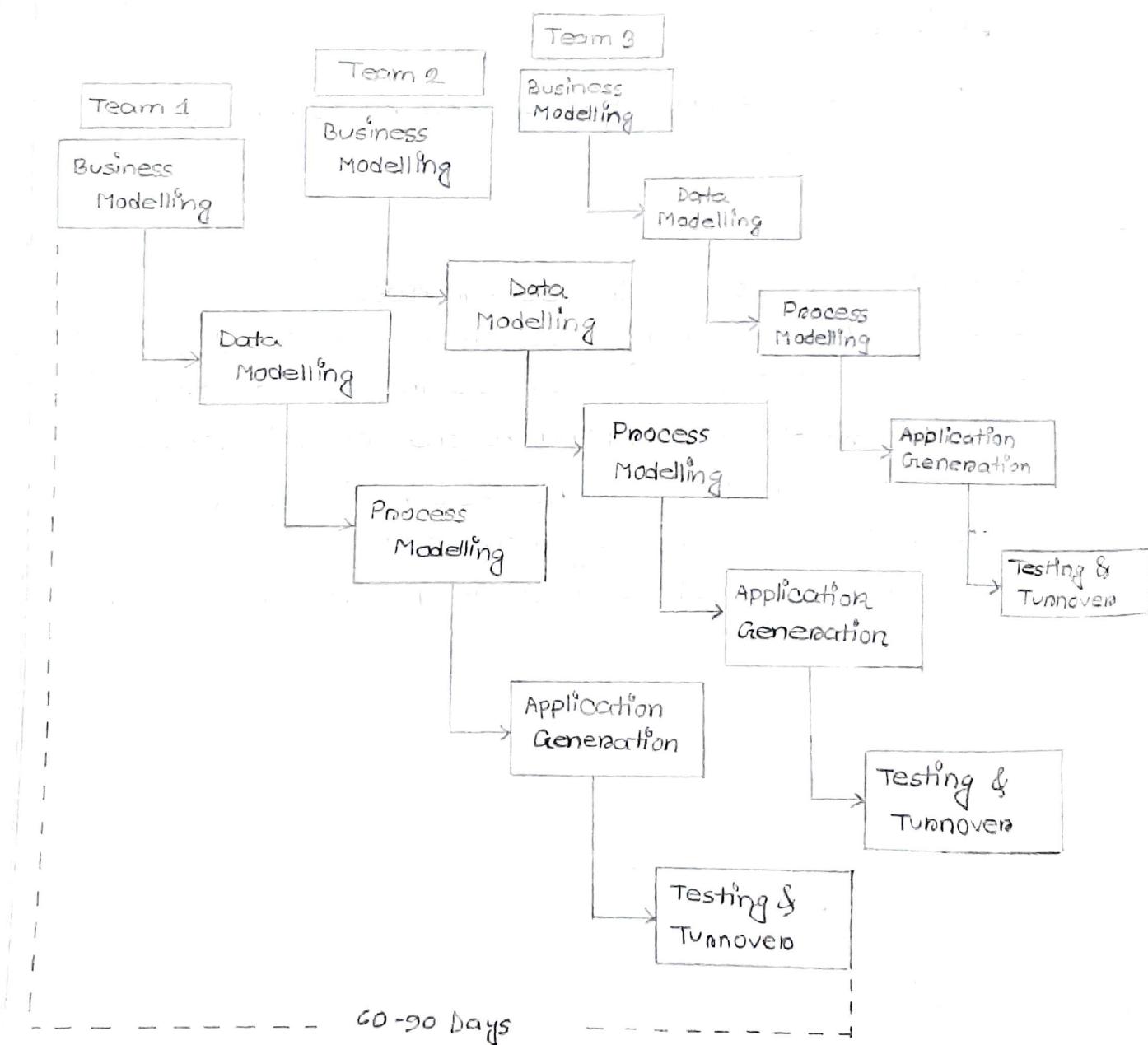
1. Customers satisfaction exists because customers can feel the product at very early stage.
2. If there is missing functionality, can be identified easily.
3. There will be less chance of software rejection.
4. Requirement changes are allowed.
5. Due to customers approval the errors can be found at early stage.
6. Customers involvement will be there in the development where it leads to better solutions for any confusion/complexity/difficult functions.
7. The developed prototype can be reused by developers and test engineers.

Disadvantages:

1. There are no parallel deliverables.
2. It is time consuming if customer asks for changes in prototype.
3. This methodology may increase the system complexity as scope of the system may expand beyond original plans.
4. The invested effort in the preparation of prototypes may be too much if not properly monitored.
5. Customers may get confused in the prototypes and real system.

Describe the RAD model.

Rapid Application Development (RAD) is an incremental software development process model which is a "high speed" adaption of the linear sequential model in which rapid development is achieved by using component based construction. If requirements are well understood and project scope is constrained, the RAD process enable a development team to create a "fully functional system" within very short time periods, such as in 60 to 90 days.



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

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

Process Modeling: Data objects defined in data modeling are converted to achieve the business information flow to achieve some specific business objective.

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

Testing & Turnovers: Test new components and all the interfaces.

When to use RAD model:

- RAD model 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 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).

What are the advantages and disadvantages of RAD model?

Advantages:

1. Requirement changes are allowed.
2. Initial reviews occur.
3. Reusability of components exists.
4. Customers can measure the progress.
5. Reduced development time.

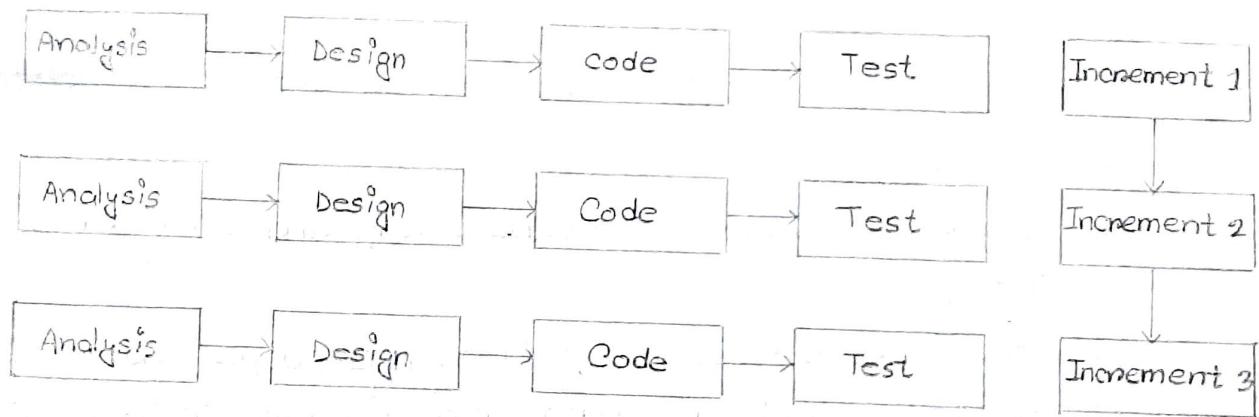
Disadvantages:

1. It requires skilled designers.
2. There will be dependency on strong technically team members for finding business requirements.
3. suitable for only short term project.
4. Dependency is high on modeling skills.

Describe the incremental model.

Incremental model are broken down into multiple standalone modules of software development cycle. These cycles are further divided into smaller and more manageable iterations.

Each iteration passes through the requirements, design, coding and testing phases. And each subsequent release of the system adds function to the previous release until all the designed functionality has been implemented.



The system is put into production when the first increment is delivered. The first increment is often a core product where the basic requirements are addressed. and supplementary features are added in the next increments. Once the core product is analyzed by the client, there is plan of development for the next increment.

Requirement Analysis: Requirement and specification of the software are collected.

Design: Some high-end function are designed during this stage.

Code: Coding of software is done during this stage.

Test: Once the system is deployed, it goes through the testing phase.

When to use Incremental model :

- Requirements of the system are clearly understood
- When demand for early release of product arises.
- When team resources are not very well skilled or trained.
- When high risk features and goals are involved.
- Major requirements must be defined; however some details can evolve with time.
- Such model is more in use for web application and product based companies.

What are the advantages and disadvantages of incremental mode

Advantages:

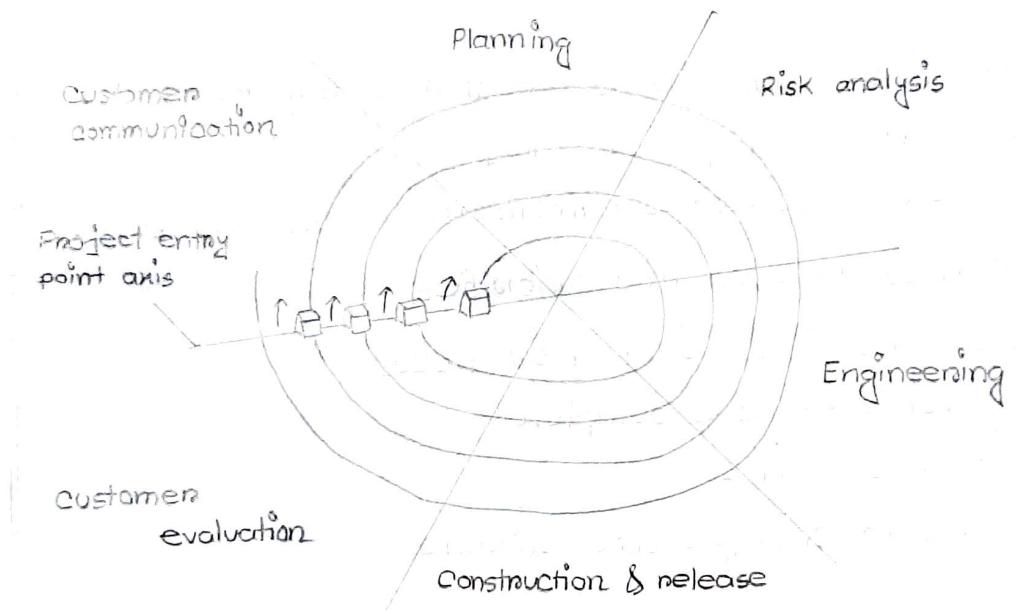
1. Generates working software quickly and early during the software life cycle.
2. More flexible - less costly to change scope and requirements.
3. Easier to test and debug during a smaller iteration.
4. Easier to manage risk because risky pieces are identified and handled during its iteration.
5. Each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

Disadvantages:

1. It is not suitable for smaller projects.
2. Progress of project mainly depends upon the risk analysis.
3. For risk analysis, experienced and skilled resources are required.
4. Requires more numbers of resources.
5. An iteration of each phase is rigid.

Describe the spiral model.

The spiral model is a type of iterative software development model which is generally implemented in high risks projects. In this method, the features of both waterfall model and prototype model are combined. In spiral model we can arrange all the activities in the form of a spiral.



The spiral model has six phases. A software project repeatedly passes through these phases in iterations.

Customer communication— tasks required to establish effective communication between developers and customers.

Planning— tasks required to define resources, timeliness, and other project related information.

Risk analysis— tasks required to assess both technical and management risks.

Engineering— tasks required to build one or more representations of the application.

Construction and release - tasks required to construct, test,

install and provide user support.

Customer evaluation - tasks required to obtain customer feedback based on evaluation of the software representations created during the engineering stage and implemented during the installation stage.

When to use spiral model:

- When costs and risk evaluation is important.
- For medium to high-risk projects.
- Long term project commitment is unwise because of potential changes to economic priorities.
- Users are unsure of their needs.
- Requirements are complex.
- New product line
- Significant changes are expected

What are the advantages and disadvantages of spiral model.

Advantages:

1. High amount of risk analysis hence avoidance of risk is enhanced.
2. Good for large and mission-critical projects.
3. Strong approval and documentation control.
4. Additional functionality can be added at a later date.
5. Software is produced early in the software life cycle.

Disadvantages:

1. Can be a costly model to use.
2. Risk analysis requires highly specific expertise.
3. Project's success is highly dependent on the risk analysis phase.
4. Doesn't work well for smaller projects.