Q1 Discuss the prototyping model. What is the effect of designing a prototype on the overall cost of the project?

=>The prototyping model is a systems development method in which a prototype is built, tested and then reworked as necessary until an acceptable outcome is achieved from which the complete system or product can be developed. Prototype is the act of making the software applications prototypes which is basically an incomplete version of the software program that is being developed. It takes place in software development and is comparable to prototyping as known in other fields like that of manufacturing and mechanical engineering. However, it is completely different from that of the final product and stimulates only a few aspects.

Types of prototype models

- Rapid throwaway- This method involves exploring ideas by quickly developing a prototype based on preliminary requirements that is then revised through customer feedback. The name rapid throwaway refers to the fact that each prototype is completely discarded and may not be a part of the final product.
- Evolutionary- This approach uses a continuous, working prototype that is refined after each iteration of customer feedback. Because each prototype is not started from scratch, this method saves time and effort.
- Incremental- This technique breaks the concept for the final product into smaller pieces, and prototypes are created for each one. In the end, these prototypes are merged into the final product.

1. I believe prototyping usually begins with sketching. It's the most intuitive thing to do.

2. It is for the client or business executives, decision makers who need to approve the design, layout, content and functionality before coding and implementation
3. Developers and coders who need to clearly understand, to the finest detail possible, what to build before they jump into coding.
4. This affect to app development cost directly or indirectly in terms of stockholders or developers.

Developing a prototype will not only **help to give a grasp of how the website or app will look like**, but also help to learn about the features to expect and match all the set of specifications prior to the full-scale development process being launched.

Q2. Compare iterative enhancement model and evolutionary process model.

=> Iterative and evolutionary development is a foundation not only of modern software methods, but as the history section of the "Evidence" chapter shows of methods used as far back as the 1960s. Agile methods are a subset of iterative and evolutionary methods. This chapter summarizes key practices:

iterative development	*evolutionary development*
risk-driven and client-driven	*evolutionary requirements*
timeboxing	*adaptive planning*

The various phases of Iterative model are as follows:

1. Requirement gathering & analysis: 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.

2. Design: In the design phase, team design the software by the different diagrams like Data Flow diagram, activity diagram, class diagram, state transition diagram, etc.

3. Implementation: In the implementation, requirements are written in the coding language and transformed into computer which are called Software.

4. Testing: After completing the coding phase, software testing starts using different test methods. There are many test methods, but the most common are white box, black box, and grey box test methods.

5. Deployment: After completing all the phases, software is deployed to its work environment.

Evolutionary Process Models

- Evolutionary models are iterative type models.
- They allow to develop more complete versions of the software.

Following are the evolutionary process models.

The prototyping model
The spiral model

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 is software working model of limited functionality.
- In this model, working programs are quickly produced.

The Spiral model

- Spiral model is a risk driven process model.
- It is used for generating the software projects.
- In spiral model, an alternate solution is provided if the risk is found in the risk analysis, then alternate solutions are suggested and implemented.

Q3. As we move outward along with process flow path of the spiral model, what can we say about software that is being developed or maintained.

=> As work moves outward on the spiral, the product moves toward a more complete state and the level of abstraction at which work is performed is reduced. **Spiral model** is one of the most important Software Development Life Cycle models, which provides support for **Risk Handling**. In its diagrammatic representation, it looks like a spiral with many loops.

The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process**.** The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using the spiral model.

The Spiral Model is a risk-driven model, meaning that the focus is on managing risk through multiple iterations of the software development process. It consists of the following phases:

1. Planning: The first phase of the Spiral Model is the planning phase, where the scope of the project is determined and a plan is created for the next iteration of the spiral.
2. Risk Analysis: In the risk analysis phase, the risks associated with the project are identified and evaluated.
3. Engineering: In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.

Q4. Explain the Scrum Agile methodology.

=> Agile scrum methodology is a project management system that relies on incremental development. Each iteration consists of two- to four-week sprints, where the goal of each sprint is to build the most important features first and come out with a Potentially Shippable Product. Scrum is **a framework of rules, roles, events, and artifacts used to implement Agile projects**. It is an iterative approach, consisting of sprints that typically only last one to four weeks, with the objective of continuously improving a product.

- Agile and scrum are two similar project management systems with a few key differences.
- Agile is more flexible and promotes leadership teams, while scrum is more rigid and promotes cross-functional teams.
- Agile lets teams develop projects in small increments called “sprints” and allows for more effective collaborations among teams working on complex projects.
- This article is for business owners and project managers who want to learn more about agile scrum methodology and how to implement it as a management process.

Agile scrum methodology is used by companies of all sizes for its ability to provide high-end collaboration and efficiency for project-based work. Agile and scrum are two different methods and can be used separately; however, their combined benefits make the agile scrum methodology the most popular use of agile. Here’s the complete guide to agile scrum methodology.

Q5. Explain the utility of Kanban CFD reports.

=> The cumulative flow diagram (also known as CFD) is one of the most advanced Kanban and Agile analytics charts. It provides a concise visualization of the three most important metrics of your Agile flow:

Cycle time
Throughput
Work in progress

Its main purpose is to show you how stable your flow is and help you understand where you need to focus on making your process more predictable. It gives you quantitative and qualitative insight into past and existing problems and can visualize massive amounts of data.

The fundamental purpose of the cumulative flow diagram is to demonstrate the stability of your workflow. Analysis of the CFD should tell you what areas need your focus, in order to maintain continuous process improvement. It enables you to improve your overall productivity and efficiency.

CFD charts are a powerful tool that Kanban teams can use to measure flow and analyze trends about a team's performance. Think of a CFD chart as a storyteller. It paints a picture of how workflows through your Kanban system within a period. Kanban Reports, such as Cumulative Flow Diagram (CFD), Throughput Run report, Cycle Time scatterplot, Cycle Time Variation, Aging work in progress, Processing Time per State, and Process Control chart, are calculated for your Kanban system as a whole.