ASSIGNMENT NO : 1

NAME: - SHANTANU SANJAY MATE

SUBJECT: - ASDM

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

ANS: -

The Prototyping Model is one of the most popularly used Software Development Life Cycle Models (SDLC models). This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.

In this process model, the system is partially implemented before or during the analysis phase thereby giving the customers an opportunity to see the product early in the life cycle. The process starts by interviewing the customers and developing the incomplete high-level paper model. This document is used to build the initial prototype supporting only the basic functionality as desired by the customer. Once the customer figures out the problems, the prototype is further refined to eliminate them. The process continues until the user approves the prototype and finds the working model to be satisfactory and it offers a small-scale facsimile of the end product and is used for obtaining customer feedback.

Types of prototypes:

Rapid Throwaway Prototyping. Evolutionary Prototyping. Incremental Prototyping. Extreme Prototyping.

Effects:

Prototyping may have some initial costs of developing, but it reduces the overall budget by helping your product to be free of the errors or glitches that could have occurred if the idea was made from scratch without any prior user testing. Furthermore, prototyping also helps to understand the intrinsic flaws, shortcomings and drawbacks that can be improved during the product development process. If the prototyping process is ignored completely, it might result in the restructuring and redesigning of the entire product after spending all your resources on its development. So, the effect of designing a prototype on the overall cost of a software project is to actually reduce the additional costs of restructuring and reframing it after its full-fledged development- which might cost a fortune.

Q2. Compare iterative enhancement model and evolutionary process model.

Ans: -

Sr no	**Iterative enhancement model**	**Evolutionary Development Model**
1.	Iterative model that is developed in multiple cycles of iterations. Project is started with a comparatively small task or component and increments are made in each cycle of the iterations until desired product is reached. During this process,	Evolutionary model is a combination of Iterative and Incremental model of software development life cycle. Delivering your system in a big bang release, delivering it in incremental process over time is the action done in this model. Some initial requirements

	developers have advantage of evaluating and testing components at each stage and gaining information that would be useful at end stages as well. This knowledge is used to also improve design of final product. Iteration includes update and execution of cycles are to be basic, direct, and particular, supporting overhaul at that stage or as an errand added to task control list. Degree of configuration detail isn't directed by iterative methodology.	and architecture envisioning need to be done. It is better for software products that have their feature sets redefined during development because of user feedback and other factors. The Evolutionary development model divides the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle. Feedback is provided by the users on the product for the planning stage of the next cycle and the development team responds, often by changing the product, plan or process. Evolutionary model solves this problem in a different approach.
2	This model iterates requirements, design, build and test phases again and again for each requirement and builds up a system iteratively till the system is completely build.	It is more customer focused model. In this model the software is divided in small units which is delivered earlier to the customers.

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

Ans:

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 Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

Each phase of the Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants are discussed below-

1. **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated, and analse at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
2. **Identify and resolve Risks:** During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.
3. **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.

4. **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

Q4. Explain the Scrum Agile methodology.

Ans:

Scrum is the type of Agile framework. It is a framework within which people can address complex adaptive problem while productivity and creativity of delivering product is at highest possible values. Scrum uses Iterative process.

Silent features of Scrum are:

• Scrum is light-weighted framework

• Scrum emphasizes self-organization

• Scrum is simple to understand

• Scrum framework help the team to work together

Sprint:

A Sprint is a time-box of one month or less. A new Sprint starts immediately after the completion of the previous Sprint.

Release:

When the product is completed then it goes to the Release stage.

Sprint Review:

If the product still has some non-achievable features, then it will be checked in this stage and then the product is passed to the Sprint Retrospective stage.

Sprint Retrospective:

In this stage quality or status of the product is checked. Product Backlog: According to the prioritize features the product is organized.

Sprint Backlog:

Sprint Backlog is divided into two parts Product assigned features to sprint and Sprint planning meeting.

Advantages:

• Scrum framework is fast moving and money efficient. • Scrum framework works by dividing the large product into small sub-products. It's like a divide and conquer strategy • In Scrum customer satisfaction is very important.

Dis-advantages:

• Scrum framework do not allow changes into their sprint. • Scrum framework is not fully described model. If you want to adopt it you need to fill in the framework with your own details like Extreme Programming (XP), Kanban, DSDM. • It can be difficult for the Scrum to plan, structure and organize a project that lacks a clear definition.

Q5. Explain the utility of Kanban CFD reports.

ANS:

The cumulative flow diagram (also known as CFD) is one of the most advanced Kanban and Agile analytics charts. CFD charts are a powerful tool that Kanban teams can use to measure flow and analyse 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. With this information handy, teams can diagnose problems and improve their process to create a more stable and predictable flow. It provides a concise visualization of the three most important metrics of your 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.