

**ASSIGNMENT 1**

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

The prototype model requires that before carrying out the development of actual software, a working prototype of the system should be built. A prototype is a toy implementation of the system. Prototyping Model is a software development model in which prototype is built, tested, and reworked until an acceptable prototype is achieved. It also creates base to produce the final system or software. It works best in scenarios where the project's requirements are not known in detail. It is an iterative, trial and error method which takes place between developer and client.

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. A prototype usually turns out to be a very crude version of the actual system, possibly exhibiting limited functional capabilities, low reliability, and inefficient performance as compared to actual software. In many instances, the client only has a general view of what is expected from the software product. In such a scenario where there is an absence of detailed information regarding the input to the system, the processing needs, and the output requirement, the prototyping model may be employed.

Q2. Compare iterative enhancement model and evolutionary process model.

**Iterative Enhancement Model:** This model has the similar phases as the waterfall model, but with fewer restrictions. In general the phases occur in the same order as in the waterfall model but these may be conducted in several cycles. A utilizable product is released at the end of the each cycle with each release providing additional functionality.

**Evolutionary Development Model:** Evolutionary development model bears a resemblance to iterative enhancement model. The similar phases as defined for the waterfall model occur here in a cyclical fashion. This model is different from iterative enhancement model in the sense that this doesn't require a useable product at the end of each cycle. In evolutionary development requirements are implemented by category rather than by prior

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 (implementation specific work accelerates as we move further from the origin).

Q4. Explain the Scrum Agile methodology.

Scrum is an agile project management framework that helps teams structure and manage their work through a set of values, principles, and practices. Much like a rugby team (where it gets its name) training for the big game, scrum encourages teams to learn through experiences, self-organize while working on a problem, and reflect on their wins and losses to continuously improve.

While the scrum I'm talking about is most frequently used by software development teams, its principles and lessons can be applied to all kinds of teamwork. This is one of the reasons scrum is so popular. Often thought of as an agile project management framework, scrum describes a set of meetings, tools, and roles that work in concert to help teams structure and manage their work. Agile scrum methodology is a sprint-based project management system with the goal of delivering the highest value to stakeholders.

Q5. Explain the utility of Kanban CFD reports.

A Cumulative Flow Diagram is a graphical representation of work as it flows through your Kanban system. It is a time-based plot, with the time interval in the x-axis and the number of cards in the y-axis. The graph is divided into different coloured bands, with the bands representing a state or column in your Kanban board.

In the Kanban methodology, Kanban boards are used to divide the workflow of a given project into three columns: "To Do" tasks, tasks that are "Work in Progress" (WIP), and tasks that are "Done." Cumulative flow diagrams collect every task that has gone through your workflow to visualize three critical metrics:

**Cycle time:** This is the total time it takes your team to complete each task from the beginning to the end. One of the benefits of CFDs is that you can see where you can optimize your workflow to reduce cycle times.

**Work in progress:** This is the number of tasks your team is actively handling at a certain time. Cumulative flow diagrams will visualize inefficiencies in your project timeline when your team has too much or too little work in progress at any given point.

**Throughput:** This is the number of tasks your team can complete over a given period. As this is the ultimate measure of your team's productivity, cumulative flow diagrams should show where you can align your efforts and resources so that throughput increases over time.