**ASSIGNMENT ON SDLC MODEL**

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

It is based on approximate requirements and carry out a quick design.

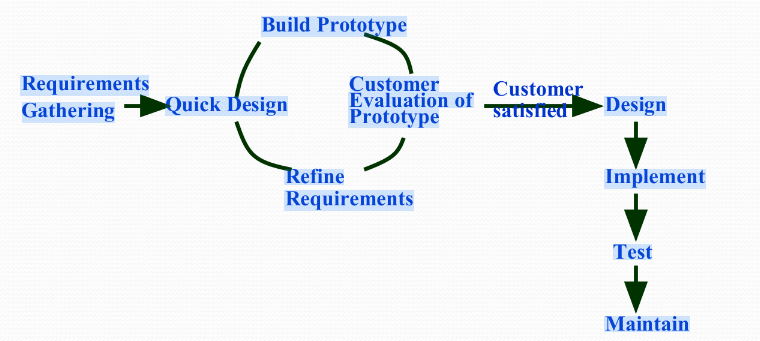
Prototype model is built using several short-cuts .Short-cuts might involve using inefficient, inaccurate, or dummy functions.

A function may use a table look-up rather than performing the actual computations.

The developed prototype is submitted to the customer for his evaluation. Based on the user feedback, requirements are refined.

This cycle continues until the user approves the prototype.

The actual system is developed using the classical waterfall approach.



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.

2. Compare iterative enhancement model and evolutionary process model.

**Evolutionary process model** :Objective is to work with customers and to evolve a final system from an initial outline specification. Should start with well-understood requirements and add new features as proposed by the customer. Throw-away prototyping.

Modern development processes take evolution as fundamental, and try to provide ways of managing, rather than ignoring, the risk.

Specification is evolved in conjunction with the software – No complete specification in the system

development contract. Difficult for large customers.

**Iterative enhancement 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.

3. 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.

Process is represented as a spiral rather than as a sequence of activities with backtracking.

Each loop in the spiral represents a phase in the process.No fixed phases such as specification or design - loops in the spiral are chosen depending on what is required.

Risks are explicitly assessed and resolved throughout the process.

Spiral model strengths: Provides early indication of insurmountable risks, without much cost.

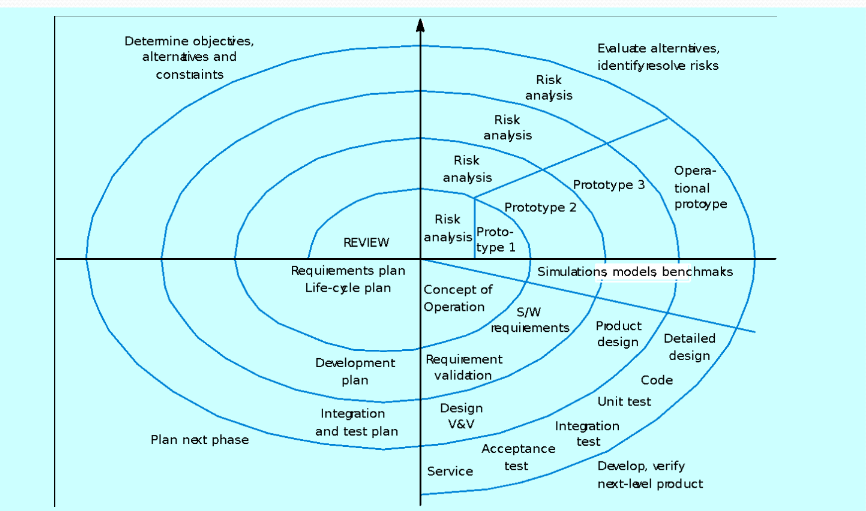
Users see the system early because of rapid prototyping tools.

Critical high-risk functions are developed first.

Users can be closely tied to all lifecycle steps.

Early and frequent feedback from users.

Cumulative costs assessed frequently.



When to use Spiral Model: When creation of a prototype is appropriate.

When costs and risk evaluation is important

For medium to high-risk projects

Users are unsure of their needs

Requirements are complex

New product line

Significant changes are expected (research and exploration).

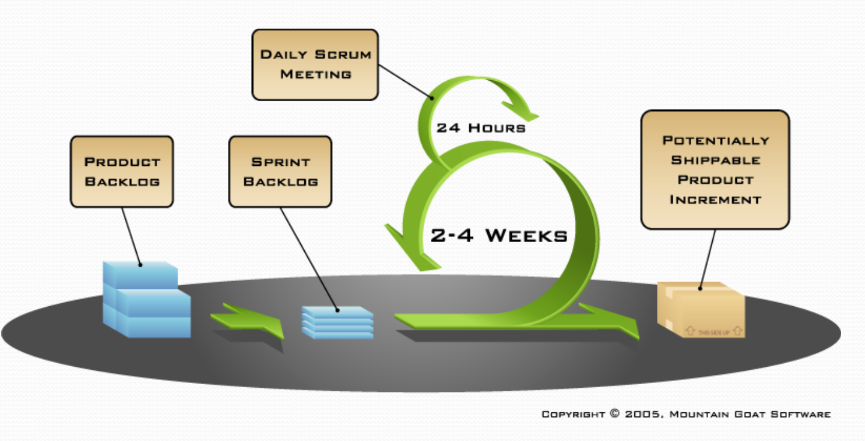
4. Explain the Scrum Agile methodology.

Focus on the code rather than the design.Based on an iterative approach to softwaredevelopment.

Intended to deliver working software quickly.

There are claims that agile methods are probably best suited to small/medium-sized business systems or PC products.

Scrum is a framework for effective collaborations among teams working on complex products. Scrum is a type of agile technology that consists of meetings, roles, and tools to help teams working on complex projects collaborate and better structure and manage their workload..



Agile scrum methodology 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 deliverable product. More features are built into the product in subsequent sprints and are adjusted based on stakeholder and customer feedback between sprints.

Agile scrum methodology has several benefits. First, it encourages products to be built faster, since each set of goals must be completed within each sprint’s time frame. It also requires frequent planning and goal setting, which helps the scrum team focus on the current sprint’s objectives and increase productivity

Benefits of scrum agile methodology:

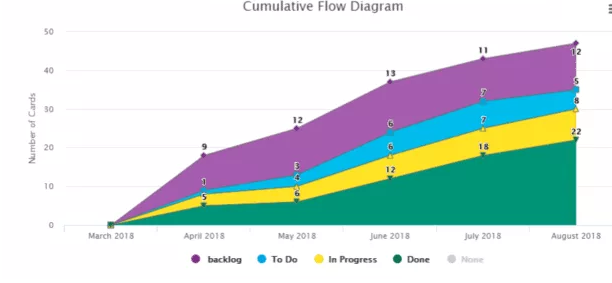
* Flexibility and adaptability
* Creativity and innovation
* Lower costs
* Quality improvement
* Organizational synergy
* Employee satisfaction
* Customer satisfaction

5. Explain the utility of Kanban CFD reports.

One of the common reasons why teams use Kanban is to visualize their workflow. Through a Kanban board teams can track their progress and spot any bottlenecks in their system. One of these tools is called the Cumulative Flow Diagram or a **CFD chart**.

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. With this information handy, teams can diagnose problems and improve their process to create a more stable and predictable flow.

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 colored bands, with the bands representing a state or column in your Kanban board.



The topmost band in a cumulative flow diagram represents the items that your team needs to do, while the bottom band represents the items that your team has done. The bands in between represent the items that your team is currently working on or the *in progress* items. In our example above, the *Backlog* and *To Do* bands represent the To Do states. The graph generated can vary depending on the configuration of your CFD chart tool if you are using an application.

The Cumulative Flow Diagram shows how many items have moved from one state to another in a given period. Depending on the tool you’re using, it should be able to give you the actual count of items per band per time interval. You can determine how many items are still waiting to be done and how many items you’re currently working on.

The trajectory of the chart should be consistently upwards, with the bands staying more or less parallel and even in width. This means that you are delivering work at a stable pace. The exception is the Done band since it should widen over time indicating more items have been completed. But not all projects are perfect and there are times when an upwards slope is not achieved. Those times can be indications of possible issues within your workstream. The width of your bands can also indicate problems in your flow.