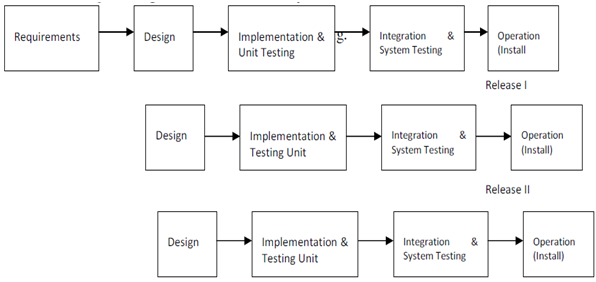
**Q1. Discuss the prototyping model. What is the effect of designing a prototype on the overall cost of the project?**

Ans: Prototype model Is the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed. It is an activity that can occur in software development and is comparable to prototyping as known from other fields, such as mechanical engineering or manufacturing. A prototype typically simulates only a few aspects of, and may be completely different from, the final product. Prototyping has several benefits: The software designer and implementer can get valuable feedback from the users early in the project. The client and the contractor can compare if the software made matches the software specification, according to which the software program is built. It also allows the software engineer some insight into the accuracy of initial project estimates and whether the deadlines and milestones proposed can be successfully met. The degree of completeness and the techniques used in the prototyping have been in development and debate since its proposal in the early 1970s.

Prototype on the overall cost of software the most obvious reason for using Throwaway Prototyping is that it can be done quickly. If the users can get quick feedback on their requirements, they may be able to refine them early in the development of the software. Making changes early in the development lifecycle is extremely cost effective since there is nothing at that point to redo. If a project is changed after a considerable amount of work has been done then small changes could require large efforts to implement since software systems have many dependencies. Speed is crucial in implementing a throwaway prototype, since with a limited budget of time and money little can be expended on a prototype that will be discarded.

**Q2. Compare iterative enhancement model and evolutionary process model.**

Ans: 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 each cycle with each release providing additional functionality.



Evolutionary Development Model: Evolutionary development model bear 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 priority.

**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: 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 (i.e., implementation specific work accelerates as we move further from the origin)

**Q4. Explain the Scrum Agile methodology.**

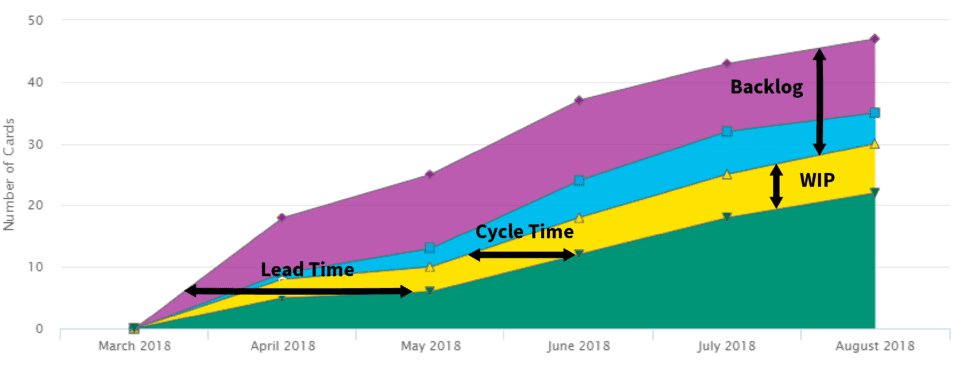
Ans: Agile scrum methodology is a[project management system](https://www.businessnewsdaily.com/9977-best-online-project-management-software.html) that relies on incremental development. Each iteration consists of two- to four-week sprints, where each sprint's goal is to build the most important features 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.

Whereas other project management methods emphasize building an entire product in one iteration from start to finish, agile scrum methodology focuses on delivering several iterations of a product to provide stakeholders with the highest business value in the least amount of time.

[Agile scrum methodology](https://www.businessnewsdaily.com/15892-project-management-styles.html) 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.

**Q.5 Explain the utility of Kanban CFD reports.**

Ans: 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. 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. 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 CFD chart can also indicate your Lead and Cycle times, as shown in the image below. 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.



The cumulative flow diagram provides insights that may not be apparent with just basing progress on a Kanban board. As the CFD chart includes historical data, it shows trends and patterns that depict the performance of the team through a given period. With a single view, one can be able to spot if a bottleneck is becoming a recurring problem within a process state. If there are deviations to what should be the trajectory or form of the chart, the team can apply corrective action immediately.

Since the CFD chart also shows performance metrics such as [WIP](https://kanbanzone.com/knowledge-base/wip-limits/), lead time, and cycle time, the team can use these data to estimate when they are likely to complete a project. If given a target release date, the team can use their historical performance on assessing whether they can meet it or not. The chart can also be used by management as a basis for capacity planning needs. Overall, the cumulative flow diagram is a very useful tool to [measure the performance](https://kanbanzone.com/kanban-resources/kanban-metrics/) of a team and create a stable and predictable flow.