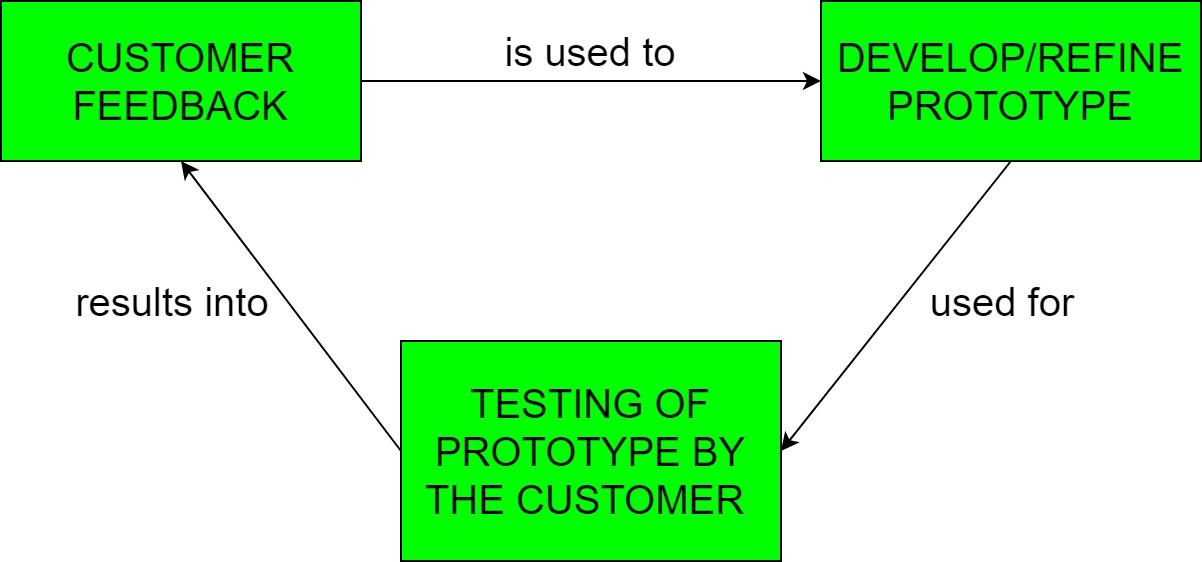
**ASSIGNMENT ON SDLC MODEL**

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

Ans: Prototyping is defined as the process of developing a working replication of a product or system that has to be engineered. It offers a small scale facsimile of the end product and is used for obtaining customer feedback as described below:

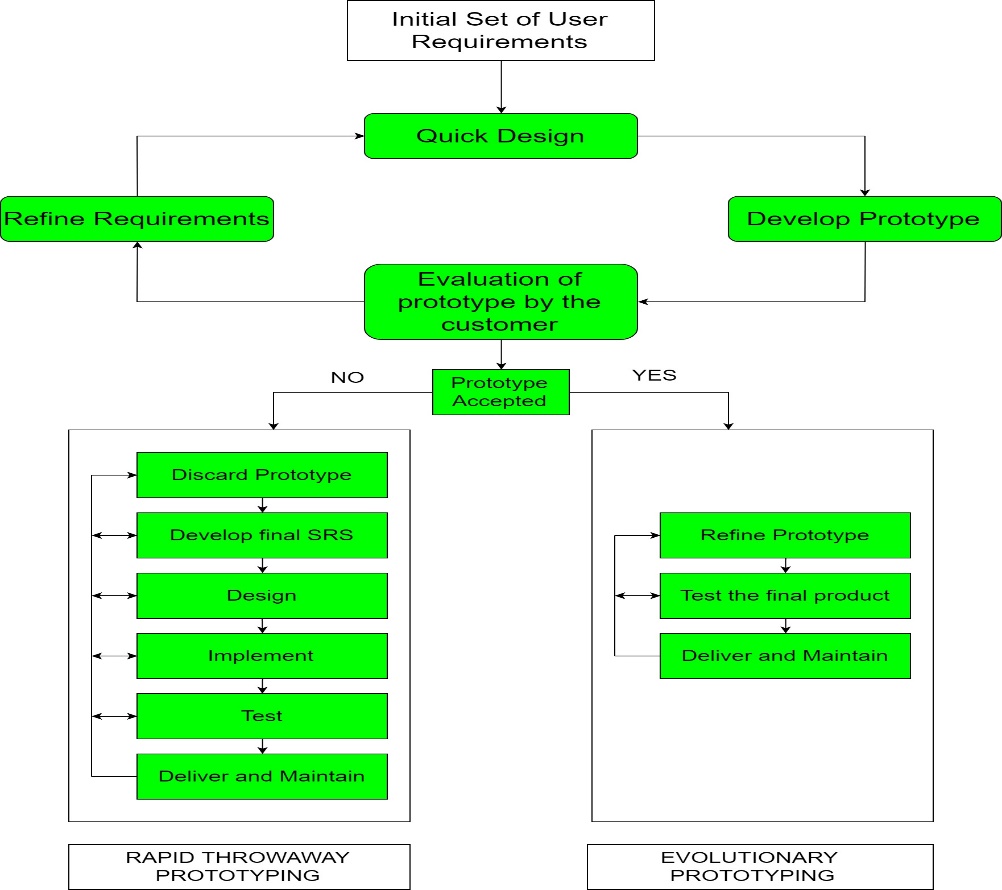


The Prototyping Model is one of the most popularly used Software Development Life Cycle Models (SDLC models). This model is used when the customers do not know the exact project requirements beforehand. In this model, a prototype of the end product is first developed, tested and refined as per customer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

There are four types of models available:

1. **Rapid Throwaway Prototyping –**   
   This technique offers a useful method of exploring ideas and getting customer feedback for each of them. In this method, a developed prototype need not necessarily be a part of the ultimately accepted prototype. Customer feedback helps in preventing unnecessary design faults and hence, the final prototype developed is of better quality.



**B) Evolutionary Prototyping –** In this method, the prototype developed initially is incrementally refined on the basis of customer feedback till it finally gets accepted. In comparison to Rapid Throwaway Prototyping, it offers a better approach which saves time as well as effort. This is because developing a prototype from scratch for every iteration of the process can sometimes be very frustrating for the developers. 

**C) Incremental Prototyping –** In this type of incremental Prototyping, the final expected product is broken into different small pieces of prototypes and being developed individually. In the end, when all individual pieces are properly developed, then the different prototypes are collectively merged into a single final product in their predefined order. It’s a very efficient approach that reduces the complexity of the development process, where the goal is divided into sub-parts and each sub-part is developed individually. The time interval between the projects’s beginning and final delivery is substantially reduced because all parts of the system are prototyped and tested simultaneously. Of course, there might be the possibility that the pieces just do not fit together due to some lack of ness in the development phase – this can only be fixed by careful and complete plotting of the entire system before prototyping starts.

**D) Extreme Prototyping –**This method is mainly used for web development. It is consists of three sequential independent phases:

**D.1)**In this phase a basic prototype with all the existing static pages are presented in the HTML format.

**D.2)**  In the 2nd phase, Functional screens are made with a simulated data process using a prototype services layer.

**D.3)** This is the final step where all the services are implemented and associated with the final prototype.

This Extreme Prototyping method makes the project cycling and delivery robust and fast, and keeps the entire developer team focus centralized on products deliveries rather than discovering all possible needs and specifications and adding necessitated features.

**Advantages –** 

* The customers get to see the partial product early in the life cycle. This ensures a greater level of customer satisfaction and comfort.
* New requirements can be easily accommodated as there is scope for refinement.
* Missing functionalities can be easily figured out.
* Errors can be detected much earlier thereby saving a lot of effort and cost, besides enhancing the quality of the software.
* The developed prototype can be reused by the developer for more complicated projects in thefuture.
* Flexibility in design.

**Disadvantages –** 

* Costly w.r.t time as well as money.
* There may be too much variation in requirements each time the prototype is evaluated by the customer.
* Poor Documentation due to continuously changing customer requirements.
* It is very difficult for developers to accommodate all the changes demanded by the customer.
* There is uncertainty in determining the number of iterations that would be required before the prototype is finally accepted by the customer.
* After seeing an early prototype, the customers sometimes demand the actual product to be delivered soon.
* Developers in a hurry to build prototypes may end up with sub-optimal solutions.
* The customer might lose interest in the product if he/she is not satisfied with the initial prototype.

**Use –**   
The Prototyping Model should be used when the requirements of the product are not clearly understood or are unstable. It can also be used if requirements are changing quickly. This model can be successfully used for developing user interfaces, high technology software-intensive systems, and systems with complex algorithms and interfaces. It is also a very good choice to demonstrate the technical feasibility of the product.

**Effect of designing a prototype on the overall cost of the project**

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.

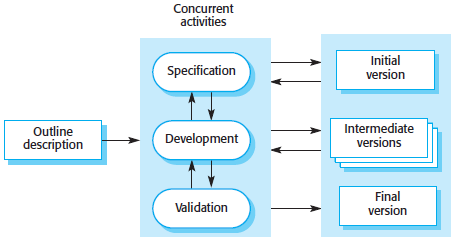
**2. Compare iterative enhancement model and evolutionary process model.**

**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.

In this Model, you can start with some of the software specifications and develop the first version of the software. After the first version if there is a need to change the software, then a new version of the software is created with a new iteration. Every release of the Iterative Model finishes in an exact and fixed period that is called iteration.



**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. **It** is useful for projects using new technology that is not well understood. This is also used for complex projects where all functionality must be delivered at one time, but the requirements are unstable or not well understood at the beginning. Evolutionary development is based on the idea of developing an initial implementation, exposing this to user comment and refining it through many versions until an adequate system has been developed (Figure ev.01). Specification, development and validation activities are interleaved rather than separate, with rapid feedback across activities.



**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?**

Ans: The Product advances to a more complete state as work spirals outward and the level of abstraction at which work is conducted decreases (i.e., implementation specific work accelerates as we move further from the origin.

**Explanation**

One of the most significant models for the software Development Life cycle that supports risk handling is the spiral is the spiral model.

In diagrammatic form, it resembles the spiral with several loops is unclear and varies from project to project. A phase of the software development process is referred to as each spiral loop

The project manager might alter the precise number of phases required to build the product depending on the project’s risk. The Project manager plays a crucial role in the spiral model of product development since they dynamically set the number of phases.

The waterfall model’s methodical, managed element are combined with idea of iterative development in the spiral model. Iterative and sequential linear development models. Iterative and sequential linear development models, or the waterfall model, are combined to create the spiral model, which places a strong emphasis on risk analysis.



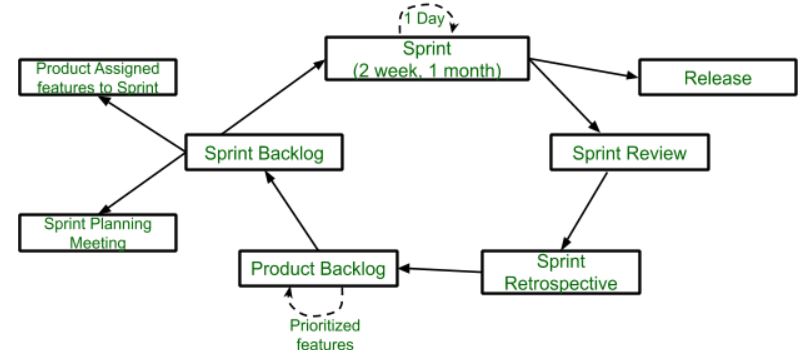
**4. Explain the Scrum Agile methodology.**

**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

**Lifecycle of Scrum:**



**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 have 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.

**Advantage of using Scrum framework:**

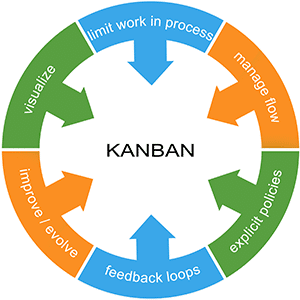
* Scrum framework is fast moving and money efficient.
* Scrum framework works by dividing the large product into small sub-products. It’s like a divide and conquer strategy
* In Scrum customer satisfaction is very important.
* Scrum is adaptive in nature because it have short sprint.
* As Scrum framework rely on constant feedback therefore the quality of product increases in less amount of time

**Disadvantage of using Scrum framework:**

* Scrum framework do not allow changes into their sprint.
* Scrum framework is not fully described model. If we want to adopt it we 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.
* The daily Scrum meetings and frequent reviews require substantial resources.

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

Ans: Kanban is one such Agile popular framework based on the idea of continuous releases to implement Agile and DevOps Software Development. **Jira Kanban Reports**helps you to track and analyze your team’s work during a project. Kanban is a Japanese word meaning ‘Visual Signal’, and hence work items are represented visually on a Kanban Board, allowing team members (Kanban Teams) to see the state of every piece of work at any time.



The following 4 important pillars in Kanban help teams to deliver products and also play a key role in Jira Kanban Reports:

* Continuous Releases
* Work-in-Progress(WIP) Limits
* List of Work
* Columns or Lanes

The work in the Kanban Board is tracked by displaying the status of work in Columns and Lanes. A Kanban Board is designed to maximize efficiency, decide Work in Progress (WIP) and visualize the same. To understand Jira Kanban Reports, the following elements of a Kanban Board are of great importance:

* **Visual Signal:** The first thing a user observes in a Kanban Board is Visual Cards (usually stickies, tickets). Kanban Teams write the project and tasks onto Cards, and Agile Teams receive Cards having a requirement of a project called**User Story**. These Visual Signals help Stakeholders understand the operational status of teams.
* **Columns:** A Column represents the workflow of a project. It displays various stages and comprises a bunch of Visual Signals. A typical workflow may consist of “To Do,” “In Progress,” “Complete,” and others.
* **WIP limits:**The maximum number of Cards in one Column at any given time is called **Work-in-Progress Limits**. These WIP limits expose bottlenecks in workflows and are an early sign to warn a team of the committed work.
* **Commitment Point:** Kanban Teams often have a backlog of their Board, where customers and teammates brainstorm for a project. The Commitment Point is a moment when an idea is picked by a team and work begins for the project.
* **Delivery Point:**When a Kanban Team reaches the end of project workflow, they are at the Delivery Point. Delivery Point is a state where a product is prepared and delivered to the customer. For any given workflow, the team’s goal is to drift the Cards from Commitment to Delivery Point within a short span elapsed called **Lead Time**. Kanban Teams thrive on decreasing their Lead Time. This process improves the delivery and recognition of the organization’s work.