Assignment

Post Graduate Diploma in Advanced Computing

AGILE SOFTWARE DEVELOPMENT METHODOLOGIES

PGDAC March 2023 (A)

Submitted To:
Ms. SARUTI GUPTA
PGDAI Coordinator

Submitted By:
NAMAN VERMA
(230109372)

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

Ans.

Prototyping Model:

The Prototype model is one of the software development life cycle models in which a prototype is built with minimal requirements.

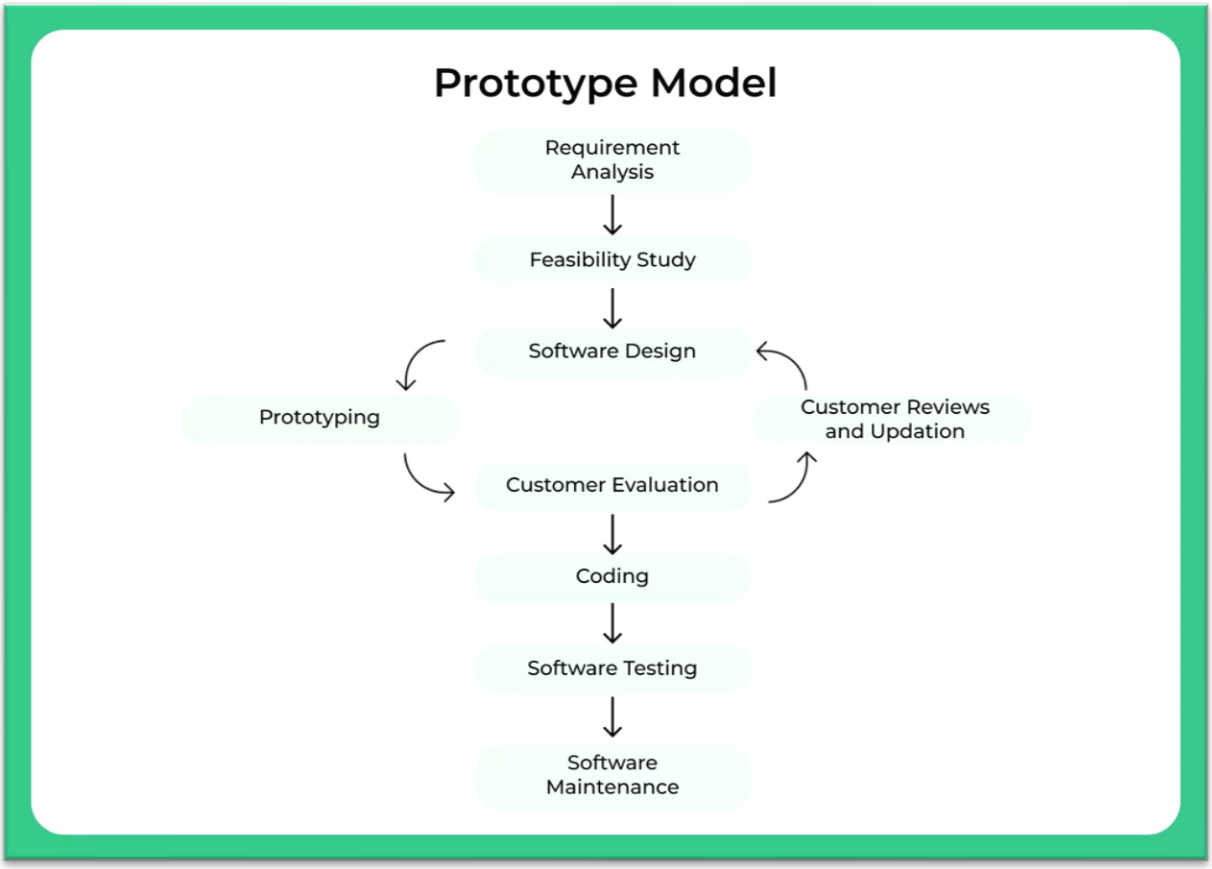
This prototype is then tested and modified based on the feedback received from the client until a final prototype with desired functionalities gets created.

This final prototype also acts as a base for the final product.

Advantages:

- Errors can be detected much earlier.
- Quick client feedback is received which speeds up the development process. Also, it helps the development team to understand the client's needs.
- It is useful when requirements are not clear from the client's end.

Diagram:



Disadvantages:

- In the beginning, it is a bit difficult to predict the exact amount of time needed to reach the final product.
- It is a time-consuming process as multiple prototypes might be needed until the client reaches the final requirements.
- Leads to implementing and then repairing way of building systems.

Conclusion:

The prototype model is an iterative method which has its advantages and disadvantages. It is particularly useful when the client does not have clarity on what all features, they need in the product.

Effect of design on Overall Cost

- At initial stage, the prototyping may not seem cost effective but it reduces the overall budget by helping your product to be free from all errors that could appear if the prior testing would not have been taken place.

- The major effect of designing a prototype is to minimize the cost of restructuring a project when its already completed.

- The complete time to reframe the entire project also gets reduced via prototyping.

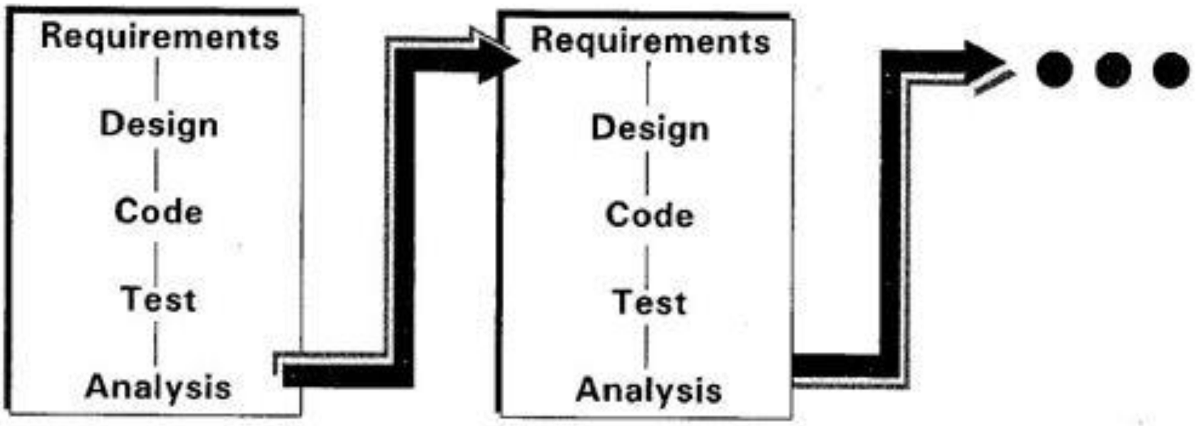
Q2. Compare iterative enhancement model and evolutionary process model.

Ans.

Iterative Enhancement Model:

The iterative enhancement model comprises the features of waterfall model in an iterative manner. The waterfall model performs each phase for developing complete software whereas the iterative model has phases similar to the linear sequential model and has an iterative nature. During the implementation phase, the project is divided into small subsets that
are increments which are implemented individually. This model comprises several phases where each phase produces an increment.

Diagram:



Iterative Enhancement Model

Evolutionary Process Model:

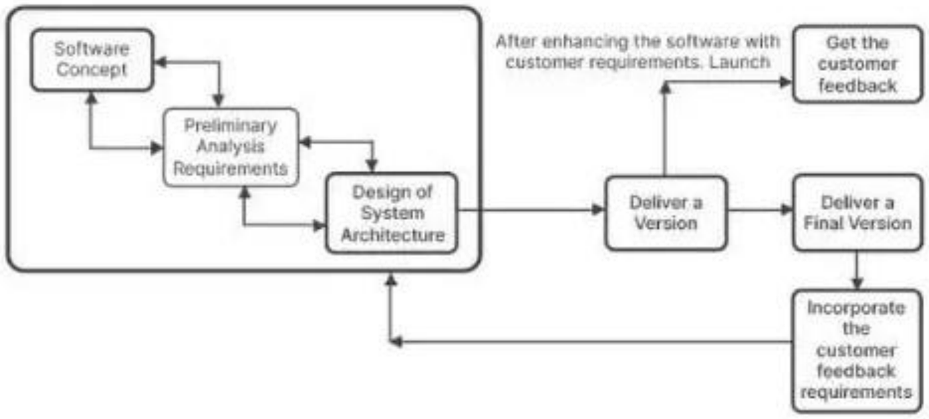
The evolutionary process model comprises the combination of characteristics of iteration and incremental model. In this model the software requirements are fragmented into several chunks that can be built and transferred incrementally.

In this model services are not required for the first stage of the system. The iterative waterfall model is used for developing an evolutionary model.

In simple words, “Iterative” + “Incremental model” = Evolutionary model.

Diagram:

Evolutionary Model in Software Engineering


Software Concept
Preliminary Analysis Requirements
Design of System Architecture
After enhancing the software with customer requirements. Launch
Get the customer feedback
Deliver a Version
Deliver a Final Version
Incorporate the customer feedback requirements

Advantages:

Iterative Enhancement Model	Evolutionary Process Model:
The feedback from early increment improves the later stages.	Risk analysis is better.
User gets benefits earlier than the conventional approach.	Initial operating time is less.
Smaller sub projects are easy to manage.	It supports changing environment.

Disadvantages:

Iterative Enhancement Model	Evolutionary Process Model:
The development cost of total product is higher.	Management complexity is more.

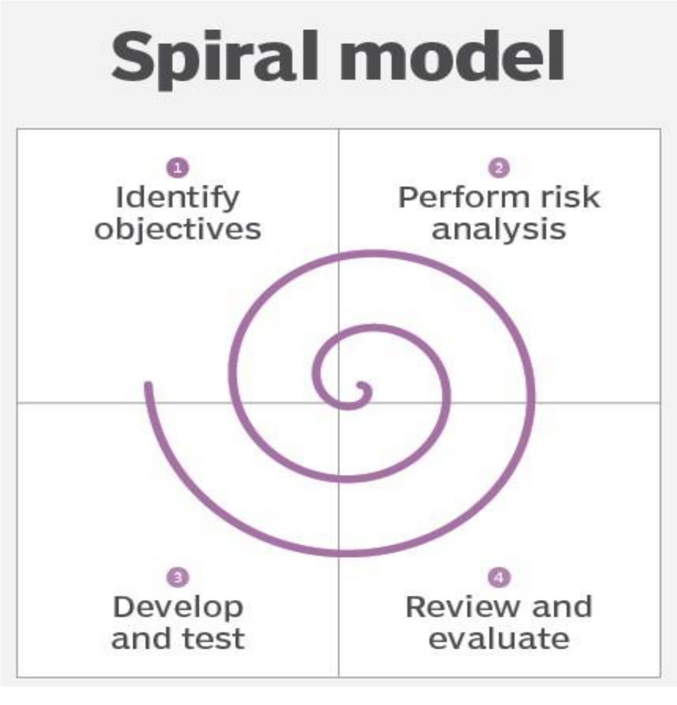
Q3. As we move outward along with process flow path of the spiral model, what can we say about software that is being developed or maintained?

Ans.

Spiral Model:

The spiral model is a method used for risk management that combines the iterative development process model with elements of the Waterfall model. The spiral model looks like a coil with many loops. The number of loops varies based on each project and is often designated by the project manager. Each loop of the spiral is a phase in the software development process.

Diagram:



Functioning of Spiral Model:

The path followed by the Spiral Model is basically divided into four phases. Every phase can be broken into four quadrants:

- Identifying and understanding requirements.

(The overall goal of this phase should be determined and all objectives should be analysed)

- Performing risk analysis.

(Risk analysis should be performed on all possible solutions in order to find any faults. Each risk should then be resolved using the most efficient strategy.)

- Building the prototype.

(In this phase the prototype is built and tested. This step mainly includes: architectural design, design of modules, physical product design and the final design. It takes the proposal that has been created in the first two quadrants and turns it into final software)

- Evaluation of the software's performance.

(Finally, in this fourth quadrant, the test results of the newest version are evaluated. This analysis allows the designer to stop and understand what worked and didn’t work before new process)

At the end of this quadrant, planning for the next phase begins and the cycle repeats. At last the end of the whole spiral, the software is finally deployed in its required form.

Conclusion:

The Spiral Model allows the product to be rolled out and refined in each phase of the spiral, with the ability to build prototypes in each stage.

A prototype is created at the beginning of each phase as a risk management technique.

The most important feature of the model is that once the project starts, it has the ability to manage unknown risks.

So the software that is being developed or maintained using this model is less complexed with no risks.

Q4. Explain the Scrum Agile methodology.

Ans.

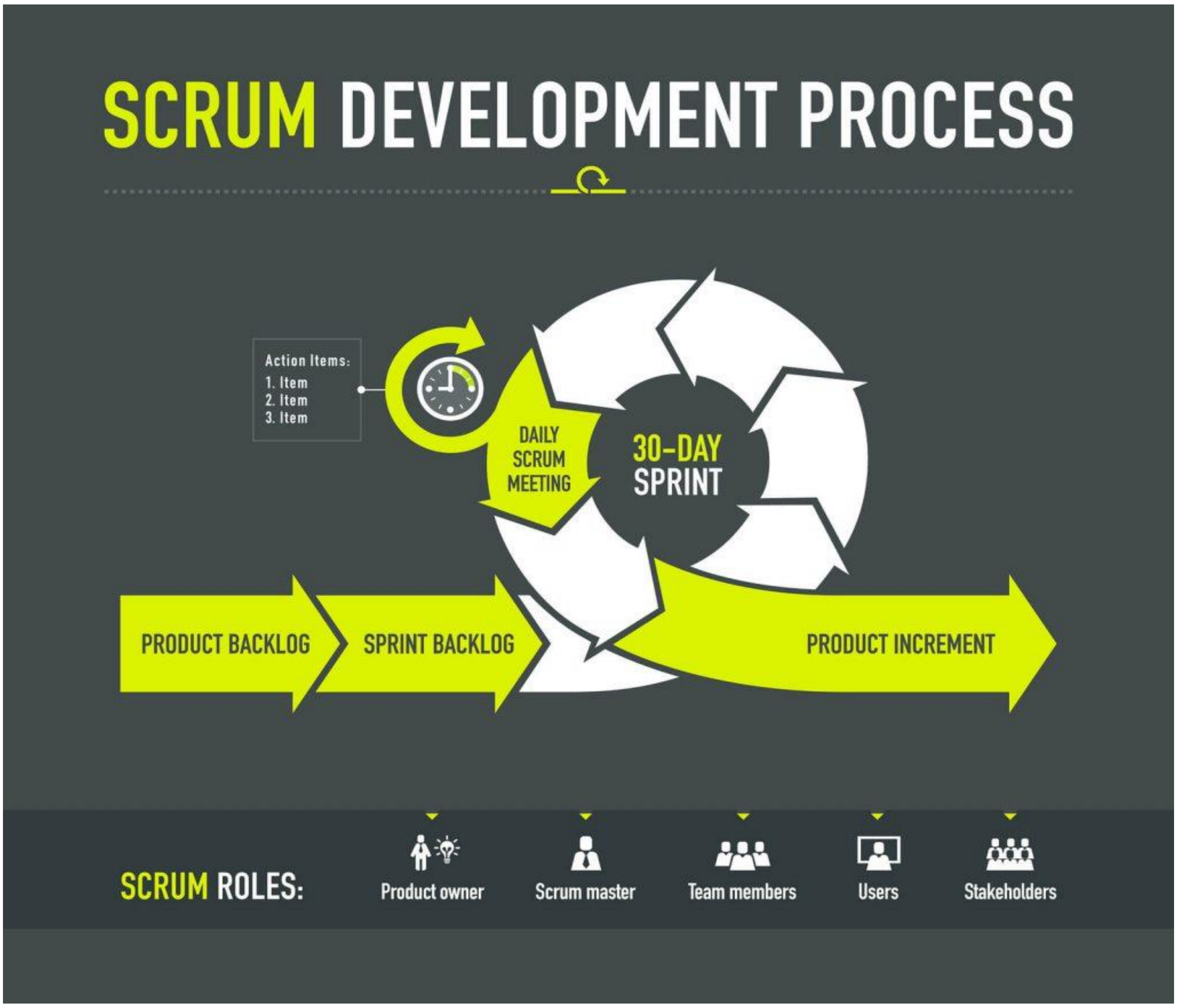
Scrum Agile Method:

Scrum: A scrum is an agile development methodology used in the development of software based on an iterative and incremental processes. Scrum is adaptable, fast, flexible and effective agile framework that is designed to deliver value to the customer throughout the development of the project.

Objective of Scrum: The primary objective of Scrum is to satisfy the customer's need through an environment of transparency in collective responsibility and continuous progress.

How it starts: The development starts from a general idea of what needs to be built, elaborating a list of characteristics ordered by priority that is product backlog that the owner of the product wants to obtain.

Diagram:



Scrum Methodology & Process:

Scrum is precisely an evolution of Agile Management. Scrum methodology is based on a set of very defined practices and roles that must be involved during the software development process.

A scrum is executed in temporary blocks that are short and periodic, called Sprints, which usually range from 2 to 4 weeks, which is the term for feedback and reflection. Each Sprint is an entity in itself, that is it provides a complete result, a variation of the final product that must be able to be delivered to the client.

The process has as a starting point, a list of requirements that make up the project plan. It is the client of the project that prioritizes these objectives considering a balance of the value and the cost that is how the iterations and consequent deliveries are determined.

Scrum methodology is used mainly for software development, but other sectors are also taking advantage of its benefits by implementing this methodology in their organizational models such as sales, marketing, & HR teams etc.

Scrum Artifacts:

Scrum Artifacts are designed to guarantee the transparency of key information in decision making.

Product Backlog (PB): The product backlog is a list that collects everything the product needs to satisfy the potential customers. It is prepared by the product owner and the functions are prioritized according to what is more and less important for the business. The goal is for the product owner to answer the question “What should be done”.

Sprint Backlog (SB)**:** It is a subset of items of the product backlog, which are selected by the team to perform during the sprint on which they are going to work. The team establishes the duration of each Sprint.

Increment: The Increment is the sum of all the tasks, use cases, user stories, product backlogs and any element that was developed during the sprint and that will be made available to the end user in the form of Software.

Advantages:

- Scrum helps to deliver the product the highest possible value.
- Scrum can help teams carry out project deliveries in a fast and effective way.
- Scrum makes sure that money and time are used efficiently.
- Large and complex projects can be separated into practically manageable parts.
- Scrum works well for dynamic and fast-moving project improvement.

Q5. Explain the utility of Kanban CFD reports.

Ans.

Cumulative Flow Diagram:

The cumulative flow diagram also known as CFD is one of the most advanced Kanban and Agile analytics charts. It provides a concise visualization of the three most important factors of your Agile 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.

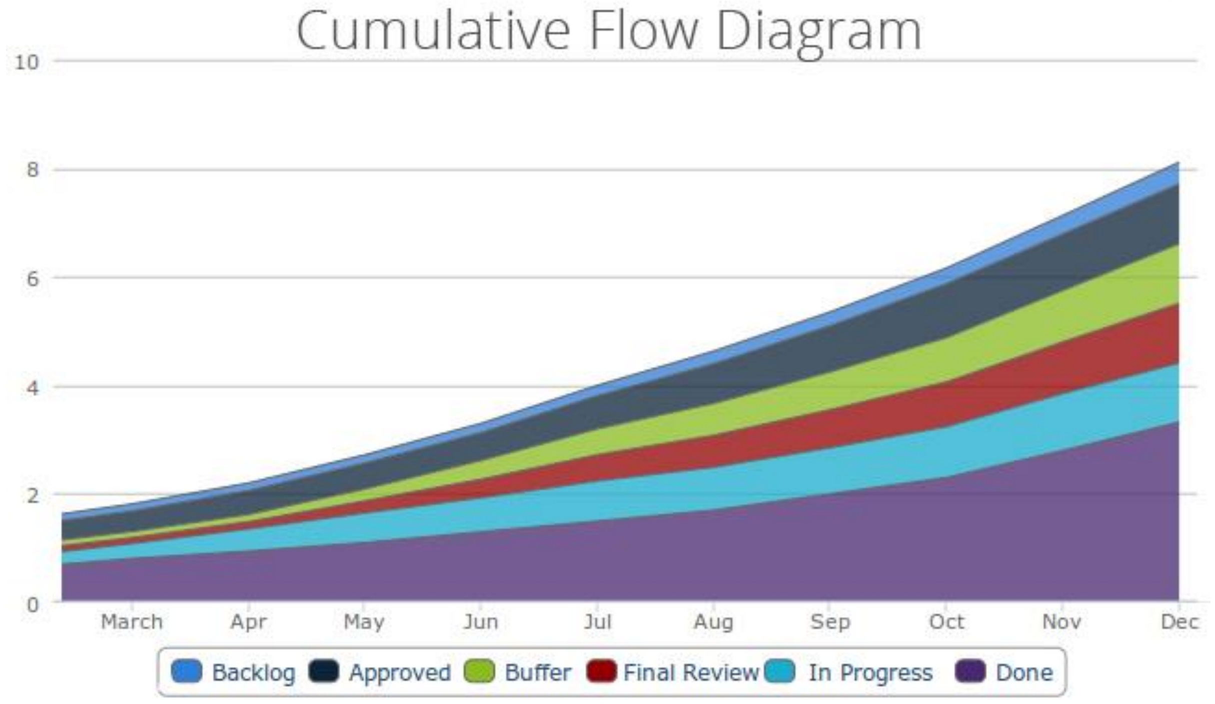
What does it show?

CFD visualises how tasks mount up over time, together with their distribution along the process stages. The graph is built from different coloured bands of tasks gathered in various columns.

One colour represents one column so that each band shows how many tasks sit at what stage of the process, in a given time (the horizontal value).

How should it look?

The ideal diagram you want to see is an evenly rising one, with bands staying more-less even, except for the "completed tasks" band, which should continuously be getting taller, just as the number of done tasks is hopefully always getting higher.



Benefits:

CFD charts are a powerful tool that Kanban teams can use to measure flow and analyse trends about a team's performance. A CFD chart acts as a storyteller. It paints a picture of how workflows through your Kanban system within a period.

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 minor error 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 factors as lead time, and cycle time, the team can use these data to estimate when they are likely to complete a project.

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 of a team and create a stable and predictable flow.

****