Assignment 01 (ASDM)

Submitted By: Mrityunjai Kumar Anand **Email ID: jai4uanand@gmail.com**

Question 01: -Discuss the prototyping model. What is the effect of designing a prototype on the overall cost of the project?

Ans: -The prototyping model is a popular approach to software development that involves creating a preliminary version of the software system to test its functionality, design, and features before building the final product. This model is often used for complex or innovative projects where requirements and specifications are not fully defined at the beginning of the development process.

This model involves the following steps

1. Identify the user requirements and design a preliminary version of the software system.
2. Develop a prototype that demonstrates the basic functionality and design of the system.
3. Evaluate the prototype and gather feedback from stakeholders.
4. Use the feedback to refine the prototype and create a final version of the system.

One of the main benefits of the prototyping model is that it allows developers to identify and address issues with the software system early in the development process. This can lead to cost savings because it is usually less expensive to fix problems during the design phase than during later stages of development or after the software has been released.

Designing a prototype can also help to reduce the overall cost of the project by:

A. **Identifying requirements:**
   Prototyping can help to clarify and refine user requirements, which can prevent costly changes and delays later in the development process.
B. **Reducing development time:**
   By identifying issues early on, the prototyping model can help to reduce the time required for development and testing.
C. **Enhancing collaboration:**
   The feedback gathered from stakeholders during the prototyping phase can help to improve communication and collaboration among team members, which can lead to faster and more efficient development.
D. **Improving user satisfaction:**
   Prototyping allows users to provide feedback on the design and functionality of the software system before it is released, which can help to ensure that the final product meets their needs and expectations.

Overall, the prototyping model can help to improve the quality and effectiveness of software development while reducing the overall cost of the project.

Submitted By: Mrityunjai Kumar Anand

Email ID: jai4uanand@gmail.com

Question 02: - Compare iterative enhancement model and evolutionary process model.

Ans: -Both the iterative enhancement model and the evolutionary process model are software development models that focus on incremental development and continuous feedback from users. However, they have some differences:

	Iterative enhancement model	Evolutionary process model
Approach	The iterative enhancement model is an incremental approach that builds on a basic version of the software in multiple iterations.	The evolutionary process model is an iterative and incremental approach that involves continuously evolving the software based on feedback from stakeholders.
Planning	The iterative enhancement model emphasizes planning for each iteration before implementation begins.	The evolutionary process model emphasizes continuous planning and adaptation throughout the development process.
Feedback	The iterative enhancement model emphasizes feedback from users at the end of each iteration,	The evolutionary process model emphasizes continuous feedback throughout the development process.
Requirements	The iterative enhancement model assumes that the requirements are known at the beginning of the development process and are refined during each iteration.	The evolutionary process model assumes that requirements are not well understood at the beginning and will evolve as the software is developed.
Deliverables	The iterative enhancement model delivers a working version of the software at the end of each iteration.	The evolutionary process model delivers a working version of the software at the end of each evolutionary cycle.
Risk management	The iterative enhancement model emphasizes risk management at each iteration.	The iterative enhancement model emphasizes risk management at each iteration.

Submitted By: Mrityunjai Kumar Anand

Email ID: jai4uanand@gmail.com

Question 03: - 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 spiral model is a software development process model that emphasizes risk management and iterative development. As we move outward along the process flow path of the spiral model, we can say the following about the software being developed or maintained:

Increasing complexity:

The spiral model emphasizes the importance of understanding and managing project risks, and as we move outward along the process flow path, the complexity of the software being developed or maintained increases. This is because risks become more complex and difficult to manage as the project progresses

.

A. **Increasing functionality:**

The spiral model emphasizes iterative development, and as we move outward along the process flow path, the software being developed or maintained becomes more functional. Each iteration adds new functionality and refines existing functionality.

B. **Increasing quality:**

The spiral model emphasizes the importance of testing and quality assurance, and as we move outward along the process flow path, the quality of the software being developed or maintained improves. Testing is done at each iteration, and defects are identified and fixed.

C. **Increasing cost:**

The spiral model emphasizes the importance of risk management, and as we move outward along the process flow path, the cost of developing or maintaining the software increases. This is because more resources are required to manage risks and ensure that the software meets the needs of the stakeholders.

Submitted By: Mrityunjai Kumar Anand

Email ID: jai4uanand@gmail.com

Question 04: - Explain the Scrum Agile methodology.

Ans: - Scrum is an Agile methodology for software development that emphasizes teamwork, collaboration, and iterative development. It is an empirical process framework in which the development process is optimized through feedback from the team and stakeholders.

The Scrum process involves following key roles.

Roles:

A. **Product Owner:**
Represents the stakeholders and is responsible for prioritizing and maintaining the product backlog

.

B. Scrum Master:
Facilitates the Scrum process, helps the team to achieve its goals, and ensures that the team follows the Scrum principles and practices.

C. Development Team:
Self-organizing and cross-functional team that is responsible for delivering the product increment.

D. Sprint Planning:
A meeting at the beginning of each sprint where the team collaborates to define a sprint goal and selects the items from the product backlog that they will work on during the sprint.

E. Daily Scrum:
A short daily meeting where the team reviews their progress and plans for the next 24 hours

Ceremonies

A. Sprint Planning:
A meeting at the beginning of each sprint where the team collaborates to define a sprint goal and selects the items from the product backlog that they will work on during the sprint

.

B. Daily Scrum:
A short daily meeting where the team reviews their progress and plans for the next 24 hours.

C. Sprint Review:
A meeting at the end of each sprint where the team presents the product increment to the stakeholders and receives feedback.

D. Sprint Retrospective:

A meeting at the end of each sprint where the team reflects on their performance and identifies areas for improvement.

Artifacts:

A. Product Backlog:
A prioritized list of features and requirements that the product owner maintains

B. Sprint Backlog:
A list of items from the product backlog that the team commits to completing during the sprint.

C. Product Increment:
A working and usable version of the product that is delivered at the end of each sprint

Question 05: - Explain the utility of Kanban CFD reports.

Ans: - Kanban CFD (**Cumulative Flow Diagram**) reports are a powerful tool for teams using the Kanban methodology. By visualizing and tracking the flow of work, teams can identify and resolve bottlenecks, track progress, and make data-driven decisions, ultimately leading to improved efficiency and performance.

Some of the specific ways in which Kanban CFD reports can be useful:

A. Identifying bottlenecks:
The CFD report allows the team to see where work items are getting stuck or delayed, which can help them identify and resolve bottlenecks in the workflow.

B. Tracking progress:
The CFD report shows the number of work items at each stage of the workflow over time, providing a visual representation of progress. This can help the team stay on track and ensure that work is moving through the system efficiently.

C. Predicting delivery dates:
By analyzing the CFD report, the team can estimate how long it will take to complete work items and predict when they will be delivered.

D. Facilitating data-driven decisions:
The CFD report provides a data-driven approach to decision-making, allowing the team to make informed decisions based on the actual performance of the system rather than intuition or guesswork.

E. Continuous improvement:
The CFD report can help the team identify areas for improvement and make changes to the workflow to optimize performance