Software Development Life Cycle Assignment 1

Name- Swapnil Ramchandra Kadam

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

1) Prototyping basically means to practically implement the findings of a discovery phase in design development.

2) It means to develop a model or a dummy website for a product that is partially functional and gives a great tangible idea of how the product will look and feel after its development.

Step of prototype model

- Start with approximate requirements.
- 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.

3) 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.

4) 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.

Q2 Compare iterative enhancement model and evolutionary process model.

Iterative Enhancement Model

The incremental model (also known as 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 incremental model has phases similar to the linear sequential model arid has an iterative nature of prototyping. During the implementation phase, the project is divided into small subsets known as increments that are implemented individually. This model comprises several phases where each phase produces an increment. These increments are identified in the beginning of the development process and the entire process from requirements gathering to delivery of the product is carried out for each increment.

Features of Iterative Enhancement model

- Avoids the problems resulting in risk driven approach in the software
- Understanding increases through successive refinements.
- Performs cost-benefit analysis before enhancing software with capabilities
- Incrementally grows in effective solution after every iteration
- Does not involve high complexity rate
- Early feedback is generated because implementation occurs rapidly for a small subset of the software.

Evolutionary model

Evolutionary model is also referred to as the successive versions model and sometimes as the incremental model. In Evolutionary model, the software requirement is first broken down into several modules (or functional units) that can be incrementally constructed and delivered.

The development first develops the core modules of the system. The core modules are those that do not need services from the other modules. The initial product skeleton is refined into increasing levels of capability by adding new functionalities in successive versions. Each evolutionary model may be developed using an iterative waterfall model of development.

Benefits of Evolutionary model

- Large project: Evolutionary model is normally useful for very large products.
- User gets a chance to experiment with a partially developed software much before the complete version of the system is released.

- Evolutionary model helps to accurately elicit user requirements during the delivery of different versions of the software.
- The core modules get tested thoroughly, thereby reducing the chances of errors in the core modules of the final products.
- Evolutionary model avoids the need to commit large resources in one go for development of the system.

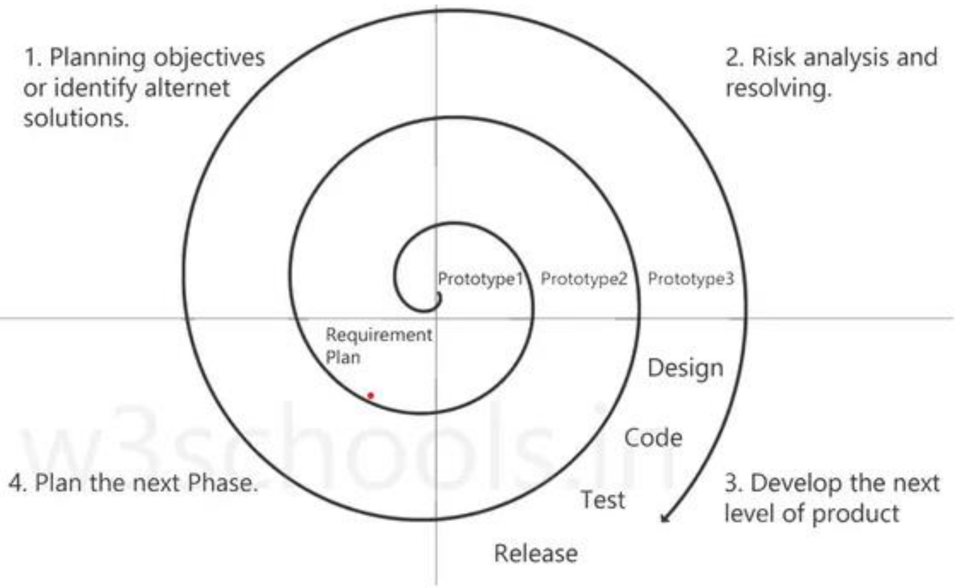
Q3 As we move outward along with process flow path of the spiral model, what can we say about software that is being developed or maintained.

1) Spiral model is one of the most important Software Development Life Cycle models, which provides support for Risk Handling. In its diagrammatic representation, it looks like a spiral with many loops.

2) The exact number of loops of the spiral is unknown and can vary from project to project. Each loop of the spiral is called a Phase of the software development process.

3) The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using the spiral model.

4) The Radius of the spiral at any point represents the expenses (cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.


1. Planning objectives or identify alternet solutions.
2. Risk analysis and resolving.
Prototype1
Prototype2
Prototype3
Requirement Plan
Design
Code
Test
Release
4. Plan the next Phase.
3. Develop the next level of product
w3schools.in

The phase of the spiral model has four quadrants, and each of them represents some specific stage of software development. The functions of these four quadrants are listed below:

1. Planning objectives or identify alternative solutions: In this stage, requirements are collected from customers and then the aims are recognized, elaborated as well as analyzed at the beginning of developing the project. If the iterative round is more than one, then an alternative solution is proposed in the same quadrant.

2. Risk analysis and resolving: As the process goes to the second quadrant, all likely solutions are sketched, and then the best solution among them gets select. Then the different types of risks linked with the chosen solution are recognized and resolved through the best possible approach. As the spiral goes to the end of this quadrant, a project prototype is put up for the most excellent and likely solution.

3. Develop the next level of product: As the development progress goes to the third quadrant, the well-known and mostly required features are developed as well as verified with the testing methodologies. As this stage proceeds to the end of this third quadrant, new software or the next version of existing software is ready to deliver.

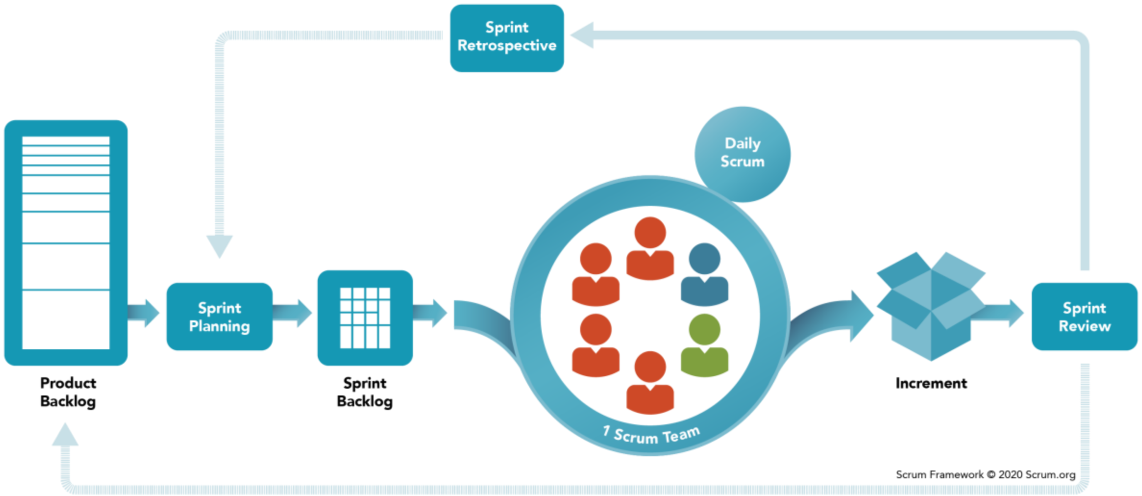
4. Plan the next Phase: As the development process proceeds in the fourth quadrant, the customers appraise the developed version of the project and reports if any further changes are required. At last, planning for the subsequent phase is initiated

Q4 Explain the Scrum Agile methodology.

1) Agile scrum methodology is used by companies of all sizes for its ability to provide high-end collaboration and efficiency for project-based work. Agile and scrum are two different methods and can be used separately; however, their combined benefits make the agile scrum methodology the most popular use of agile.

2) Agile scrum methodology is a project management system that 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.

3)Whereas other project management methods emphasize building an entire product in one operation 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.



In practice, Scrum consists of the following key steps:

- ***Product Backlog*** - A product backlog is a list of everything that needs to be achieved on a project, broken down into individual items. This is where the baseline requirements of every feature needed for the end product are prioritized by the product owner for the Scrum team.

  The product backlog is usually made up of several different types of items:

User stories, which are high-level descriptions of a feature, told from the perspective of the end-user of the product.

Bugs are problems that arise that the product owner wants to be fixed.

Tasks, which are assigned to the scrum team to complete.

- ***Sprint Backlog*** - The sprint backlog is further broken down into tasks for the team to execute. Every item on the sprint backlog needs to get developed, tested and documented. The product owner helps the scrum team come up with a sprint backlog during their sprint meeting. The sprint backlog is often represented as a task board, which is broken up into columns that represent the workflow. They tend to have the following titles:

- ***Product Increment***-This is the most important scrum artifact. The product increment is all the product backlog items that have been completed during a sprint. Each sprint is potentially creating shippable product increments, and so the product increment must fit into the team's definition of done and be acceptable to the product owner..

- ***Burndown Chart*** **-** Progress throughout the sprint is measured by the number of story points left to complete in that sprint and displayed using a 'burndown chart'.

There can many different roles in a Scrum project, but there are three core roles:

- Product Owner

- Scrum Master

- Scrum Team

All of these people are committed to the scrum project, and are augmented by various other specialist roles

Benefits of agile Scrum methodology

1)Flexibility and adaptability – since the requirements (consisting of features, user stories and other backlog items) are deliberately kept high-level at the start, with the ability to redefine and replan after each release and sprint, you have the flexibility to adapt the requirements from feedback from the customer and end users 'on the fly'.

2**)Creativity and innovation** – Scrum lets integrated teams of developers, designers, testers, and business/functional experts all work together rather than working in individual 'silos'. This cross-discipline collaboration fosters creativity and innovation.

3) Lower risks and costs – a study found that the majority of features delivered in traditional waterfall projects were never used or needed by customers, due to scope creep and feature bloat. With Scrum, you are ruthlessly pruning the release backlog after each sprint so such features (and their associated costs) are quickly weeded out. The short-time boxes reduce the likelihood of risks materializing during the sprint and thus addressing increment delivery.

4) Quality improvement – With engineering approaches such as continuous exploration, continuous integration, continuous delivery, test-driven-development, DevOps and relentless test automation, Scrum yields much higher-quality code, products and process.

Q5 Explain the utility of Kanban CFD reports.

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 metrics of your flow:

1) Cycle time
2) Throughput
3) Work in progress

The following are the utility of Kanban CFD report

1) A Cumulative Flow Diagram (CFD) is a visual tool used in Kanban to measure project health, examine workflow at different stages, identify any obstacles, and monitor the overall work progress.

2) Only take what is necessary: Only pulling what is required is essential for a successful Kanban implementation. By doing this, overproduction is avoided, expenses are reduced, and operations are better suited to meet market demands.

3) Create the Precise Quantity Needed: Taking only what you need results in only producing the precise amount of goods needed.

4) Quality improvement – With engineering approaches such as continuous exploration, continuous integration, continuous delivery, test-driven-development, DevOps and relentless test automation, Scrum yields much higher-quality code, products and process.

5) Organizational synergy – Scrum lets team members from different disciplines harness their different skills in a cross-cutting product development ethos, so that the finished product integrates the best skills in the organization not just what’s written on their job title.