Assignment 1

1. Question 1
   1. Discuss the prototyping model.

When building new Software many a times the product owner may just have a vague idea of the requirement and does not want to invest a lot of resources. To do feasibility studies with limited resources the product owner can quickly build a model for testing. During testing all results are taken and compared with initial intended vision of the products to find if prototype meets the requirement. Also many such prototype can be made by tweaking the original vision to test it in a real world environment.

There are four types of prototyping models:

1. Rapid Throwaway model: This prototype is used for exploring ideas and customer feedback and the developed prototype need not be part of the accepted prototype. The customer feedback is used to design faults.
2. Evolutionary Prototyping: Prototyping is incrementally improved on the basis of customer feedback so we don’t build a new prototype from scratch when customer feedback is taken to fic the problem.
3. Incremental Prototyping: In this the prototype is broken into many subcomponent and then are individually tested. If any component is faulty then that component is either improved or rejected. This means we don’t have to build everything from scratch but downside is that there is chance all the components may not integrate into a fully functioning prototype.
4. Extreme Prototyping: This method is mainly used for web development. Initially we create a basic prototype consisting of static webpages in HTML format. Then we create one with a functional screens made with a simulated data process using prototype service layer. Finally all services are implemented and integrated to create a final prototype.
   1. What is the effect of designing a prototype on the overall cost of the project?

It is costly as lot of money and time will be spent on building the prototype. Customers demand might change during prototype phase which will delay the process. It might lead to increase in no. of iterations of the prototype before the final model is accepted. The documentation process may be poorly executed due to changing requirements. Developers may end up with a sub optimal solution as they are developing many models. The process can be long drawn out one which might lead to customer losing interest in the project.

1. Compare iterative enhancement model and evolutionary process model.

|  |  |
| --- | --- |
| Evolutionary Model | Incremental Enhancement Model |
| The entire software is developed together and customers keep giving insight for improving the product. | System is broken into small components which is developed independently with each increment delivery part of the required functionality. |
| If any component isn’t functioning properly then entire project may need to start again. Either it is discarded or the project goes back to the point where the problem occurred. | If any component is not functioning properly it is discarded the only that portion is redeveloped. |
| As all component are developed together so all components are in sync with one another. | As all components are developed together they may not be in sync with one another. Scalability can also be an issue. |
| It is costly because during faults the projects may need to go back where last time it was functioning properly. | Cost is less as only faulty components need to be fixed. |

1. 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?

There is no fixed path to follow. Many different steps of the project may happen simultaneously. This give lot of freedom to developers as they are not bound to a process when a problem occurs.

1. Explain the Scrum Agile methodology.

Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

The philosophy behind scrum is that there is a Product Owner orders the work for a complex problem into a Product Backlog. Team turns work does in incremental pattern called sprints. The results from each sprint are analysed and adjusted before starting the next sprint and this goes on.

A typical scrum team consists of:

Developers: - Role of developers is to

* Create a plan for the Sprint, the Sprint Backlog;
* Instill quality by adhering to a Definition of Done;
* Adapt their plan each day toward the Sprint Goal; and,
* Holdeach other accountable as professionals.

Product Owners: The Product Owner main purpose is to manage Product Backlog, which includes:

* Developing and explicitly communicating the Product Goal;
* Creating and clearly communicating Product Backlog items;
* Ordering Product Backlog items;
* Ensuring that the Product Backlog is transparent, visible and understood.

Product owner is single point authority.

Scrum Master: The Scrum Master is the one serves as leader of the Scrum Team and serves it in several ways, including:

* Coaching the team members in self-management and cross-functionality;
* Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done;
* Causing the removal of impediments to the Scrum Team’s progress; and,
* Ensuring that all Scrum events take place and are positive, productive, and kept within the time box.

The Scrum Master serves the Product Owner in several ways, including:

* Helping find techniques for effective Product Goal definition and Product Backlog management;
* Helping the Scrum Team understand the need for clear and concise Product Backlog items;
* Helping establish empirical product planning for a complex environment; and,
* Facilitating stakeholder collaboration as requested or needed.

The Scrum Master serves the organization in several ways, including:

* Leading, training, and coaching the organization in its Scrum adoption;
* Planning and advising Scrum implementations within the organization;
* Helping employees and stakeholders understand and enact an empirical approach for complex work; and,
* Removing barriers between stakeholders and Scrum Teams.

Scrum Events:-

Sprint: It is recurring event for a fixed time period where all events which includes Sprint Planning, Daily Scrums, Sprint Review, and Sprint Retrospective are worked on.

Sprint Planning means how the events within sprints need to be organized to run the sprint smoothly.

Daily Scrums are short meetings where all the things regarding the sprint are reviewed.

Sprint Review are the analysis of the events of sprints before its completion and thses reviews are taken into consideration when moving onto the next sprint.

Sprint Retrospective is the final event that Scrum Team inspects how the last Sprint went with regards to individuals, interactions, processes, tools, and their Definition of Done.

Scrum Artefacts:

It is the final work done by the team where end product and team performance is reviewed.

Product Backlog: It is the emergent, ordered list of what is needed to improve the product for further development.

Sprint Backlog: It is the emergent, ordered list of what is needed to improve the performance of the scrum team for how to work in the future. It consists of what has been achieved and how to achieve.

Increments: They are small targets that are meant to be achieved to reach the final goal. Each Increment is necessary step be achieved to move onto next increment and it must be verified that all Increments must work together. Multiple Increments may be created within a Sprint.

1. Explain the utility of Kanban CFD reports.

Kanban CFD reports are cumulative graphical representation of different work metrics use to evaluate work performance.

To study these graphs we look at these metrics:

Cycle time: The total time taken by the team to complete each assignment from start to finish.

Work in Progress: This is the number of tasks that your team is currently working on.

Throughput: The number of tasks your team can finish in a particular period.

Lead Time:The time span between the time a task enters the work system and the time it is completed is known as lead time.

By looking at the graph are aim to prioritize task in such a way or redistribute the workforce over different task in such way that we can minimize lead time which means all given task spend least amount of time in idle when then enter the system for execution.

The graph consist of amount of work done, to be done, and being worked on the y axis and time spent on the x-axis. The amount of work to be done and work in progress multiplied by time spent gives us the lead time.