SDLC MODEL – ASSIGNMENT 3

NAME- Gaurav Chhotu Patil

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

What does prototyping mean?
Prototyping basically means to practically implement the findings of a discovery phase in design development. 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.

On the road to bringing a new product to market, without having an iota of doubt whether the product will be successful or not– one of the major steps is to build a prototype of your product. It is a very essential method in design thinking– for making your ideas come to life, and before spending your entire resources on its development, you actually know whether the product is worthy or will even work for your target audience. A person who wants their design process to be collaborative, productive and eventually successful, needs to build a product prototype. Developing a prototype will not only help to give a grasp of how the website or app will look like, but also help to learn about the features to expect and match all the set of specifications prior to the full-scale development process being launched.

Effect of designing a prototype on the overall cost of a software 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.

Note :-

Under what circumstances is it beneficial to construct a prototype?
Prototyping helps a user to–

- Shaping your ideas : Prototyping helps in improving the creative process by starting with the idea and iterating that idea over and over until it satisfies your needs.
- Communicate your ideas for meeting the client’s business objectives : It helps to communicate the idea if it does not exactly meet the client’s business objectives, and you can avoid the complete restructuring of the developed product by just revising the product prototype.
- Coming up with solutions quickly : You and your team can easily collaborate and use the cloud space for designing the prototype together and leave comments and suggestions for fixing the loopholes in the prototype.
- Building a bridge between the designers and customers : Prototypes help the designers to understand the customer’s tastes and preferences, according to which they slowly realize which product can lure which customer, and improve communication between them

Q2. Compare iterative enhancement model and evolutionary process model.

Evolutionary Process Model Evolutionary process model : (e.g. Prototyping model) resembles Iterative enhancement model, but this differs from iterative enhancement model in the sense that this does not release product at the end of each cycle. This model 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.

Prototyping model : In this process model, firstly develop a working prototype (i.e., incomplete versions of the software program being developed.) Of the software instead of developing the actual software. The working prototype is developed as per current requirements. The developers use this prototype to refine the requirements and prepare the final specification document. When the prototype is created, it is reviewed by the customer. Typically this review gives feedback to the developer that helps to remove uncertainties in the requirements of the software.

Incremental Process Model Increment process model : (e.g. Iterative Enhancement model) are effective in the situations where requirements are defined precisely and there is no confusion about the functionality of the final product and functionality can be delivered in phases as per desired priorities. After every cycle, a useable product is given to the customer.

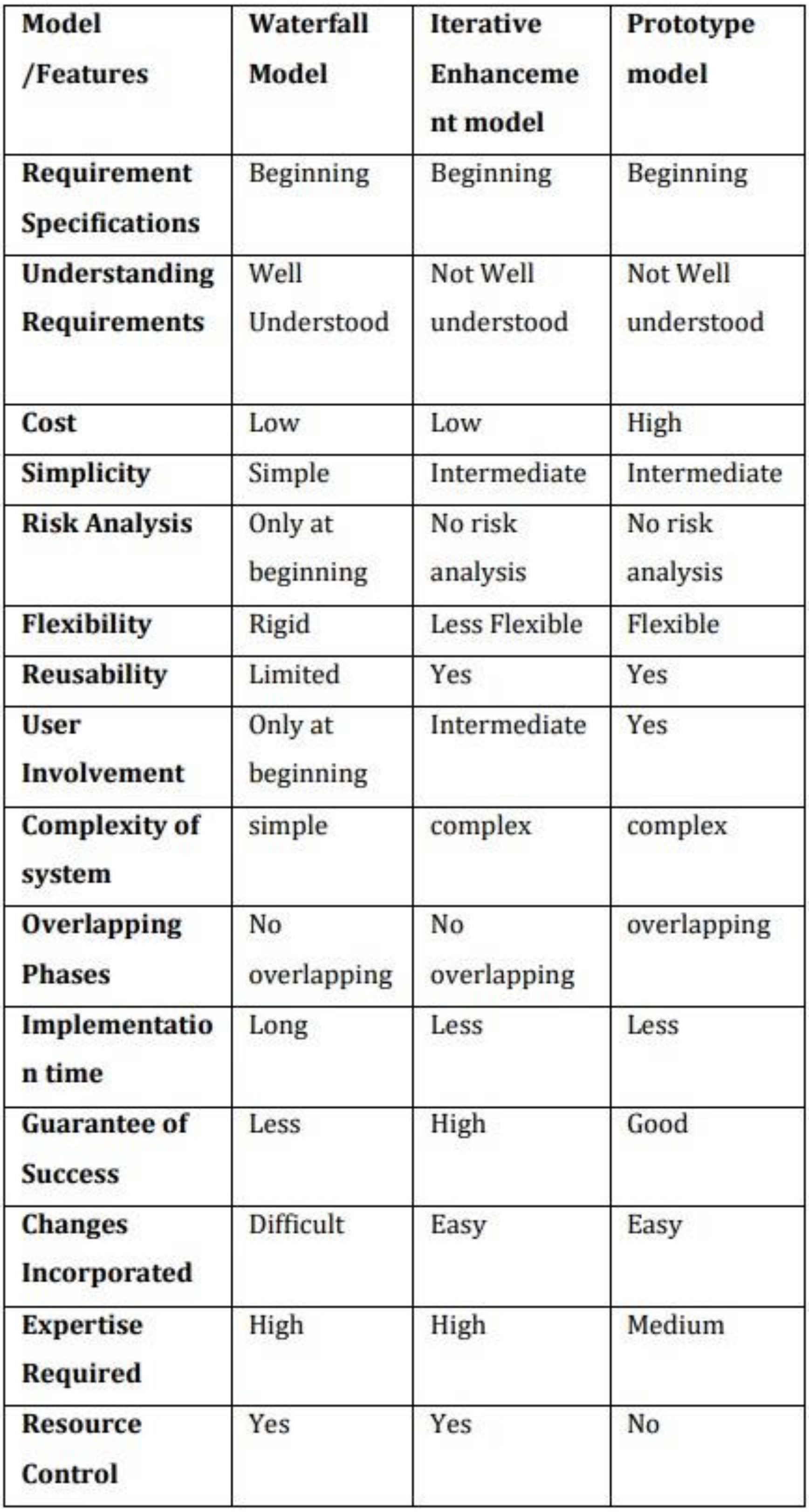
Iterative Enhancement model:

This model contains the following phases: Requirement Analysis & Specification Design Implementation and Unit Testing Integration and System Testing Operation and Maintenance

- Requirement analysis and specification
- Design
- Implementation and unit testing
- Integration and System Testing
- Operation and Maintenance

These phases are same as the waterfall model, but these may be conduct in several cycles in Iterative enhancement model. A useable product is released at the end of the each cycle, with each release providing additional functionality. This model does deliver an operational quality product at each release. The complete product is divided into releases and the developer delivers the product release by release. A typical product will usually have many releases. With this model, first release may be available within few weeks or months whereas the customer generally waits months or years to receive a product using other model.

Fig. Comparison between iterative enhancement model and evolutionary process model(Prototype).

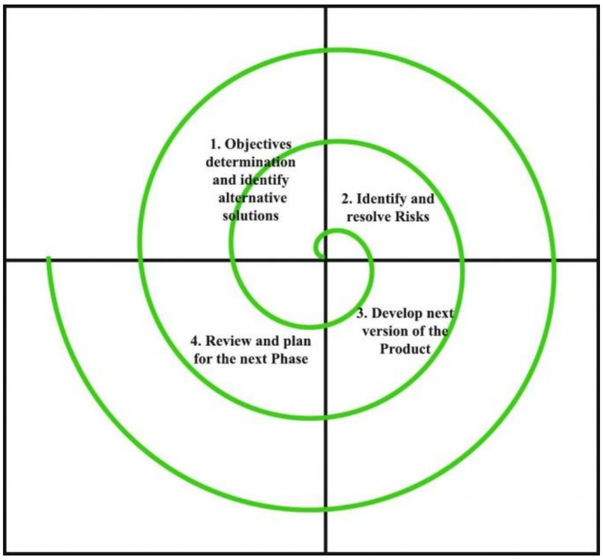


Model /Features	Waterfall Model	Iterative Enhancement model	Prototype model
Requirement Specifications	Beginning	Beginning	Beginning
Understanding Requirements	Well Understood	Not Well understood	Not Well understood
Cost	Low	Low	High
Simplicity	Simple	Intermediate	Intermediate
Risk Analysis	Only at beginning	No risk analysis	No risk analysis
Flexibility	Rigid	Less Flexible	Flexible
Reusability	Limited	Yes	Yes
User Involvement	Only at beginning	Intermediate	Yes
Complexity of system	simple	complex	complex
Overlapping Phases	No overlapping	No overlapping	overlapping
Implementation time	Long	Less	Less
Guarantee of Success	Less	High	Good
Changes Incorporated	Difficult	Easy	Easy
Expertise Required	High	High	Medium
Resource Control	Yes	Yes	No

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?

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. 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. 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.

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. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

Each phase of the Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants are discussed below -

Objectives determination and identify alternative solutions : Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
Identify and resolve Risks : During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.
Develop next version of the Product : During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
Review and plan for the next Phase : In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

Q4. Explain the Scrum Agile methodology.

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

Benefits of Scrum Methodology: Easily Scalable: Scrum processes are iterative and are handled within specific work periods, which makes it easier for the team to focus on definite functionalities for each period.
Compliance of expectations: The client establishes their expectations indicating the value that each requirement/ history of the project brings, the team estimates them and with this information the Product Owner establishes its priority

Compliance of expectations: The client establishes their expectations indicating the value that each requirement/ history of the project brings

Flexible to changes: Quick reaction to changes in requirements generated by customer needs or market developments.

Time to Market reduction: The client can start using the most important functionalities of the project before the product is completely ready.

Higher software quality: The working method and the need to obtain a functional version after each iteration, helps to obtain a higher quality software.

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

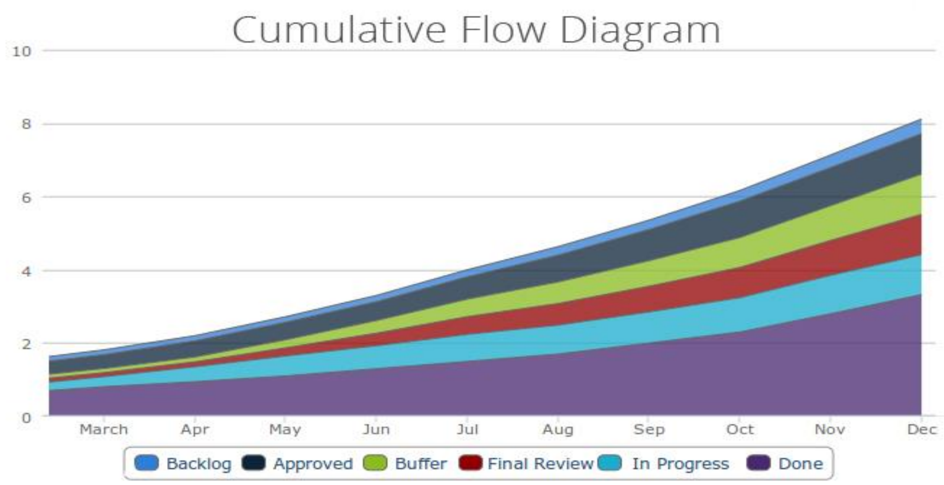
Sprint Backlog (SB): It is a subset of items of the product backlog, which are selected by the team to perform during the sprint on which they are going to work.

Q5. Explain the utility of Kanban CFD reports.

Cumulative Flow Diagram is an analytical tool, fundamental to Kanban method. It allows teams to visualize their effort and project progress. When there's an impediment about to occur within the process - the CFD is where you'll see it first. Instead of the graph staying smooth and rising gently, there will be a bump, a sudden ascend or descend. So, where being able to predict problems is concerned, this is the very graph you need.

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



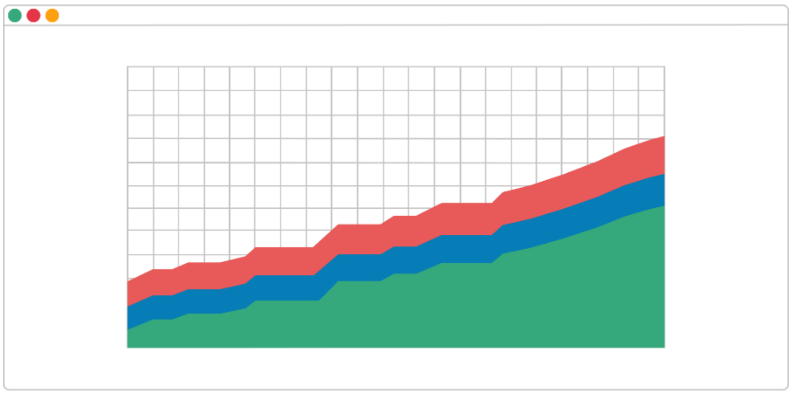
benefit from it?

The CFD only requires 3 basic things from the process - a Backlog, an In Progress column and a Done section - using this type of division allows you to read valid and usable information from the diagram. Therefore, any team, that utilizes this kind of workflow division, can benefit from Cumulative Flow. Whether you use Scrum, Kanban or any other custom project management method, for as long as you organize it in task groups, the CFD will be of great help.

Understanding the data on a CFD Chart
You can spot whether your process is stable in just a single glance by looking at how the top and the bottom line of each band in your cumulative flow diagram are progressing.
There are three common scenarios:

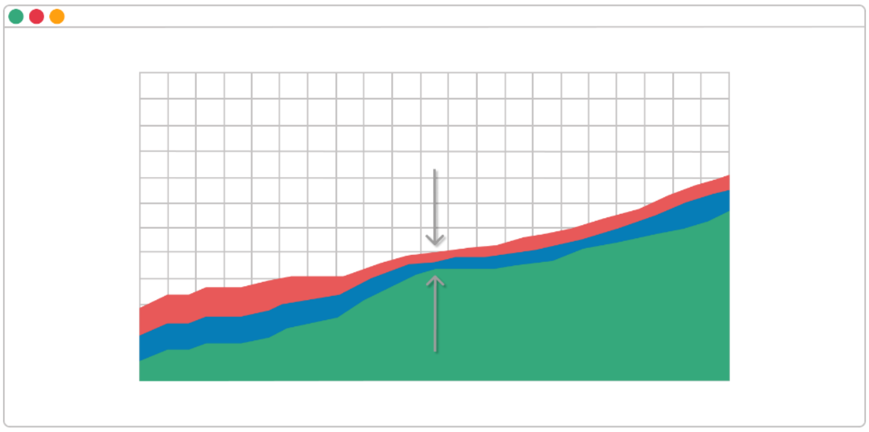
The Bands are Progressing in Parallel

This means that your throughput is stable, and new tasks are entering your workflow in parallel to those that are leaving it. This is the ideal outcome and shows that you can focus your efforts on shortening your assignments' cycle times.



A Band is Rapidly Narrowing

If a band on your CFD is continuously narrowing, that means that the throughput of the stage it represents is higher than the entry rate. This is a sign that you've got more capacity than you really need at this stage, and you should relocate it to optimize the flow.



A Band is Rapidly Widening

Whenever this happens on a cumulative flow diagram, the number of cards that enter the corresponding stage on the Kanban board is higher than the number of assignments leaving it. It is a common problem caused by multitasking and other waste activities that don't generate value.