CDAC NOIDA SEPT 2022 B\_BATCH

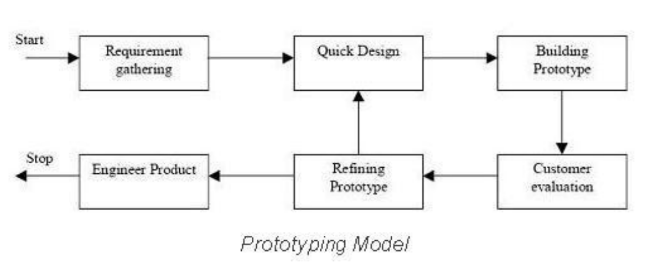
ASSIGNMENT 1: ASDM

SANDEEP K GADHE

**Q.1]** Discuss the prototyping model. What is the effect of designing a prototype on the overall cost of the project?

**ANS**. The basic idea in Prototype model is that instead of freezing the requirements before a design or coding can proceed, a throwaway prototype is built to understand the requirements. This prototype is developed based on the currently known requirements. Prototype model is a software development model. By using this prototype, the client can get an “actual feel” of the system, since the interactions with prototype can enable the client to better understand the requirements of the desired system. Prototyping is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. The prototype are usually not complete systems and many of the details are not built in the prototype. The goal is to provide a system with overall functionality.

**Diagram of Prototype model:**



**Advantages of Prototype model:**

1. Users are actively involved in the development

2.Since in this methodology a working model of the system is provided, the users get a

better understanding of the system being developed.

3.Errors can be detected much earlier.

4.Quicker user feedback is available leading to better solutions.

5.Missing functionality can be identified easily.

**Disadvantages of Prototype model:**

1. Leads to implementing and then repairing way of building systems.

2. Practically, this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.

**When to use Prototype model:**

Prototype model should be used when the desired system needs to have a lot of interaction with the end users.Typically, online systems, web interfaces have a very high amount of interaction with end users, are best suited for Prototype model. It might take a while for a system to be built that allows ease of use and needs minimal training for the end user.Prototyping ensures that the end users constantly work with the system and provide feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface system.

**What is the effect of designing a prototype on the overall cost of a software project?**

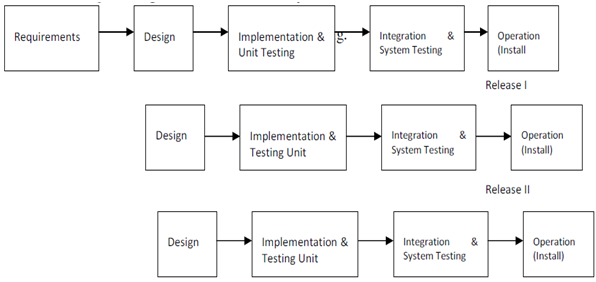
Prototyping may have some initial costs of developing, but it reduces the overall budget by helping your product to be free of the errors or glitches that could have occurred if the idea was made from scratch without any prior user testing. Furthermore, prototyping also helps to understand the intrinsic 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.

**Q2. Compare iterative enhancement model and evolutionary process model.**

**ANS.**

* **Iterative Enhancement Model:**

This model has the similar phases as the waterfall model, but with fewer restrictions. In general, the phases occur in the same order as in the waterfall model but these may be conducted in several cycles. A utilizable product is released at the end of each cycle with each release providing additional functionality.



* **Evolutionary Development Model:**

Evolutionary development model bears a resemblance to iterative enhancement model. The similar phases as defined for the waterfall model occur here in a cyclical fashion. This model is different from iterative enhancement model in the sense that this doesn't require a useable product at the end of each cycle. In evolutionary development requirements are implemented by category rather than by priority.

* Evolutionary models are iterative type models.
* They allow to develop more complete versions of the software.

**Following are the evolutionary process models.**  
  
 1. The prototyping model  
 2. The spiral model  
 3. Concurrent development model

**The prototyping model**  
 Prototype is defined as first or preliminary form using which other forms are copied or

derived.

 Prototype model is a set of general objectives for software.

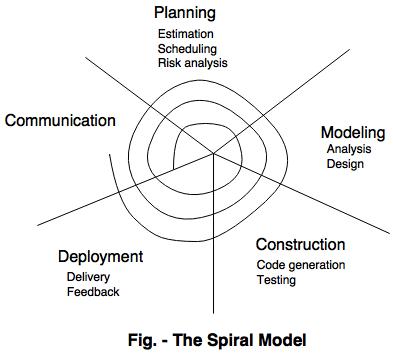
 It does not identify the requirements like detailed input, output.

 It is software working model of limited functionality.

 In this model, working programs are quickly produced.

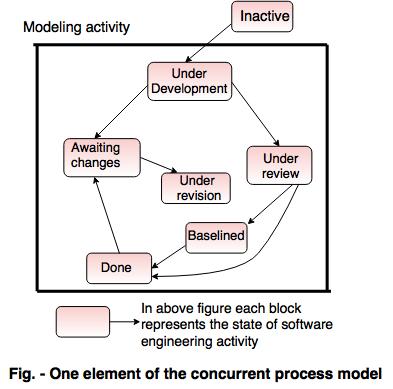
**The spiral model**

* Spiral model is a risk driven process model.
* It is used for generating the software projects.
* In spiral model, an alternate solution is provided if the risk is found in the risk analysis, then alternate solutions are suggested and implemented.
* It is a combination of prototype and sequential model or waterfall model.
* In one iteration all activities are done, for large project's the output is small.

**The framework activities of the spiral model are as shown in the following figure.**  
  


**The concurrent development model**

* The concurrent development model is called as concurrent model.
* The communication activity has completed in the first iteration and exits in the awaiting changes state.
* The modeling activity completed its initial communication and then go to the underdevelopment state.
* If the customer specifies the change in the requirement, then the modelling activity moves from the under-development state into the awaiting change state.
* The concurrent process model activities moving from one state to another state.



|  |  |
| --- | --- |
| Iterative enhancement model | Evolutionary process model |
| 1) This model has similar phases as the water fall model, but with few restrictions.  2) the phases occur in the same order as in the waterfall model but these may be conducted in several cycles.  3) A utilizable product is released at the end of each cycle with each release providing additional functionality.  4)SRS document is prepared and then prioritize their requirement.  5)Developer implement the specific requirement in one or more cycles of design.  6) The product is divided into releases.  7) The first release may be available with in few weeks or month.  8) Advantages:   * This module is flexible. * Lower the initial delivery cost.   9) Disadvantages:   * Need good planning and design. * Total cost is higher than waterfall module. * Needs clear and complete definition of the whole system. | 1) Evolutionary development model bears a resemblance to iterative enhancement model.  2) This model is different from iterative enhancement model in the sense that this doesn’t require a useable product at the end of each cycle.  3) In this model requirements are implemented by category rather than by priority.  4) Evolutionary model is the combination of iterative and incremental model of software development life cycle.  6) Incremental model firs implement a few basic features and deliver to the customer. Then build the next part and deliver it again and repeat this step until the desired system is realised. No long-term plans are made.  7) Advantages   * Risk analysis is better. * It supports changing environment. * Initial operating time is less. * Better suited for large mission-critical projects. * During the life cycle software is produced early which facilitates customer evaluation and feedback.  8) Disadvantage of the Model:  * Management complexity is more. * Not suitable for smaller projects. * Can be costly to use. * Highly skilled resources are required for risk analysis. |

**3. 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 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.

**The below diagram shows the different phases of the Spiral Model:**



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-

1. Objectives determination and identify alternative solutions: Requirements are gathered from the customers and the objectives are identified, elaborated, and analysed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
2. 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.
3. 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.
4. 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.

**Risk** handling in spiral model

A risk is any adverse situation that might affect the successful completion of a software project. The most important feature of the spiral model is handling these unknown risks after the project has started. Such risk resolutions are easier done by developing a prototype. The spiral model supports coping up with risks by providing the scope to build a prototype at every phase of the software development.

Advantages of Spiral Model:

1) Risk Handling: The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.

2) Good for large projects: It is recommended to use the Spiral Model in large and complex projects.

3) Flexibility in Requirements: Change requests in the Requirements at later phase can be incorporated accurately by using this model.

4) Customer Satisfaction: Customer can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

Disadvantages of Spiral Model:

1) Complex: The Spiral Model is much more complex than other SDLC models.

2) Expensive: Spiral Model is not suitable for small projects as it is expensive.

3) Too much dependability on Risk Analysis: The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced experts, it is going to be a failure to develop a project using this model.

4) Difficulty in time management: As the number of phases is unknown at the start of the project, so time estimation is very difficult.

**Q 4) Explain the Scrum Agile methodology.**

**Ans.**

* **How does agile scrum work?**

Agile scrum methodology is the combination of the agile philosophy and the scrum framework. Agile means “incremental, allowing teams to develop projects in small increments. Scrum is one of the many types of agile methodology, known for breaking projects down into sizable chunks called “sprints.” Agile scrum methodology is good for businesses that need to finish specific projects quickly.

Agile scrum methodology is a [project management system](https://www.businessnewsdaily.com/9977-best-online-project-management-software.html) 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.

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.

Agile scrum methodology has several benefits. First, it encourages products to be built faster, since each [set of goals](https://www.businessnewsdaily.com/11225-set-achievable-business-goals.html) must be completed within each sprint’s time frame. It also requires frequent planning and goal setting, which helps the scrum team focus on the current sprint’s objectives and increase productivity.

* **What is agile?**

Agile is a process that allows a team to more efficiently manage a project by breaking it down into several stages, each of which allows for consistent collaboration with stakeholders to promote steady improvements at every stage.

**Key takeaway:**Agile lets a team manage a project more efficiently by breaking it down into several stages.

**What are the values of agile?**

Agile was first described in the Agile Manifesto in 2000 by a group of developers who sought out a new method of writing software. The manifesto cites four values:

1. Individuals and interactions over processes and tools
2. Working software over comprehensive documentation
3. Customer collaboration over contract negotiation
4. Responding to change over following a plan

**What are the 12 principles of agile?**

The Agile Manifesto also enacted 12 principles in reference to software development and was later reconfigured to fit a wider perspective of users:

1. Customer satisfaction
2. Early and continuous delivery
3. Embrace change
4. Frequent delivery
5. Collaboration of businesses and developers
6. Motivated individuals
7. Face-to-face conversation
8. Functional products
9. Technical excellence
10. Simplicity
11. Self-organized teams
12. Regulation, reflection and adjustment

* What are the benefits of agile scrum methodology?
* [Flexibility and adaptability](https://www.businessnewsdaily.com/5696-pivot-business-strategy-shift.html)
* Creativity and innovation
* Lower costs
* Quality improvement
* Organizational synergy
* Employee satisfaction
* Customer satisfaction

**Q5. Explain the utility of Kanban CFD reports.**

**What is a Cumulative Flow Diagram?**

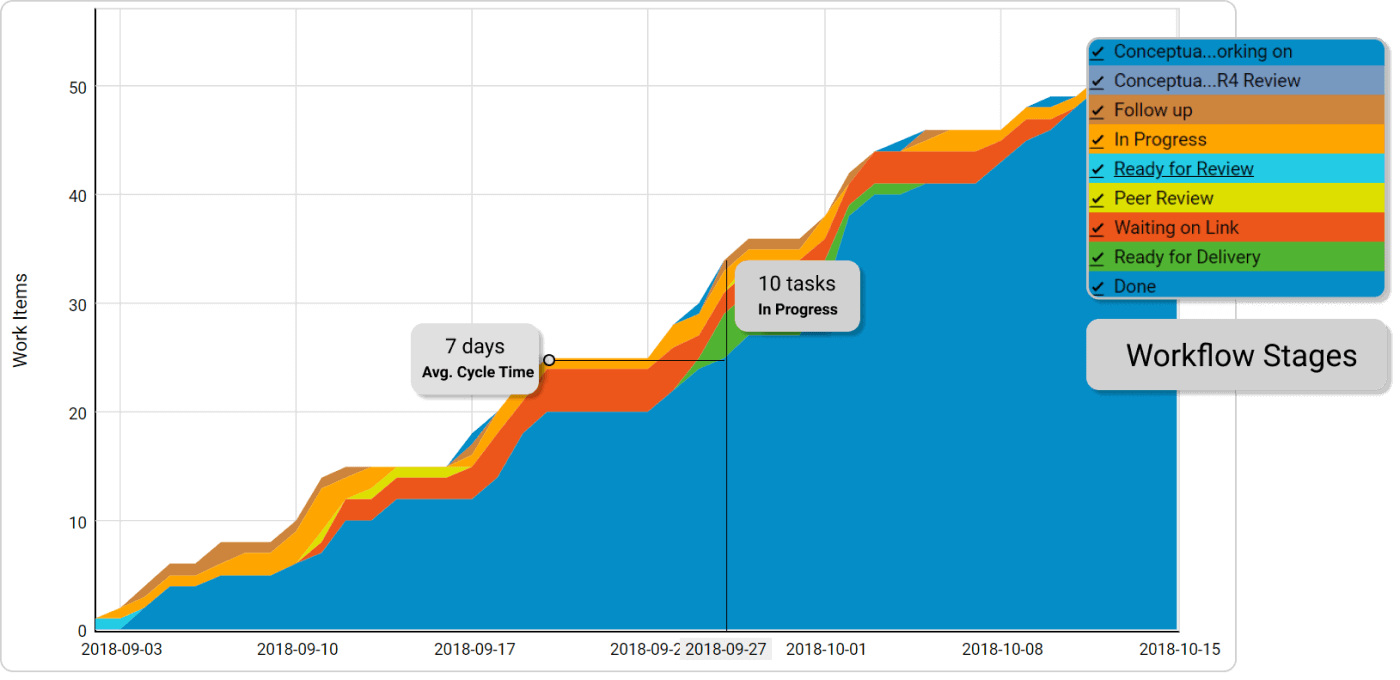
The cumulative flow diagram (also known as CFD) is one of the most advanced Kanban and Agile analytics charts. It provides a concise visualization of the three most [important metrics of your flow](https://kanbanize.com/agile/project-management/agile-metrics):

* 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.

## How To Read a Cumulative Flow Diagram

The chart tracks the total number of work items in the columns of the "In Progress" section on your Kanban board each day.



The horizontal axis of the CFD represents the time frame for which the chart is visualizing data. The vertical axis shows the cumulative number of cards in the workflow at various points in time.

The differently coloured bands that divide sections of the upward flow are the different stages of your workflow as they appear on the Kanban board itself. The bands always go up or sideways in accordance with the number of assignments that go through your process.

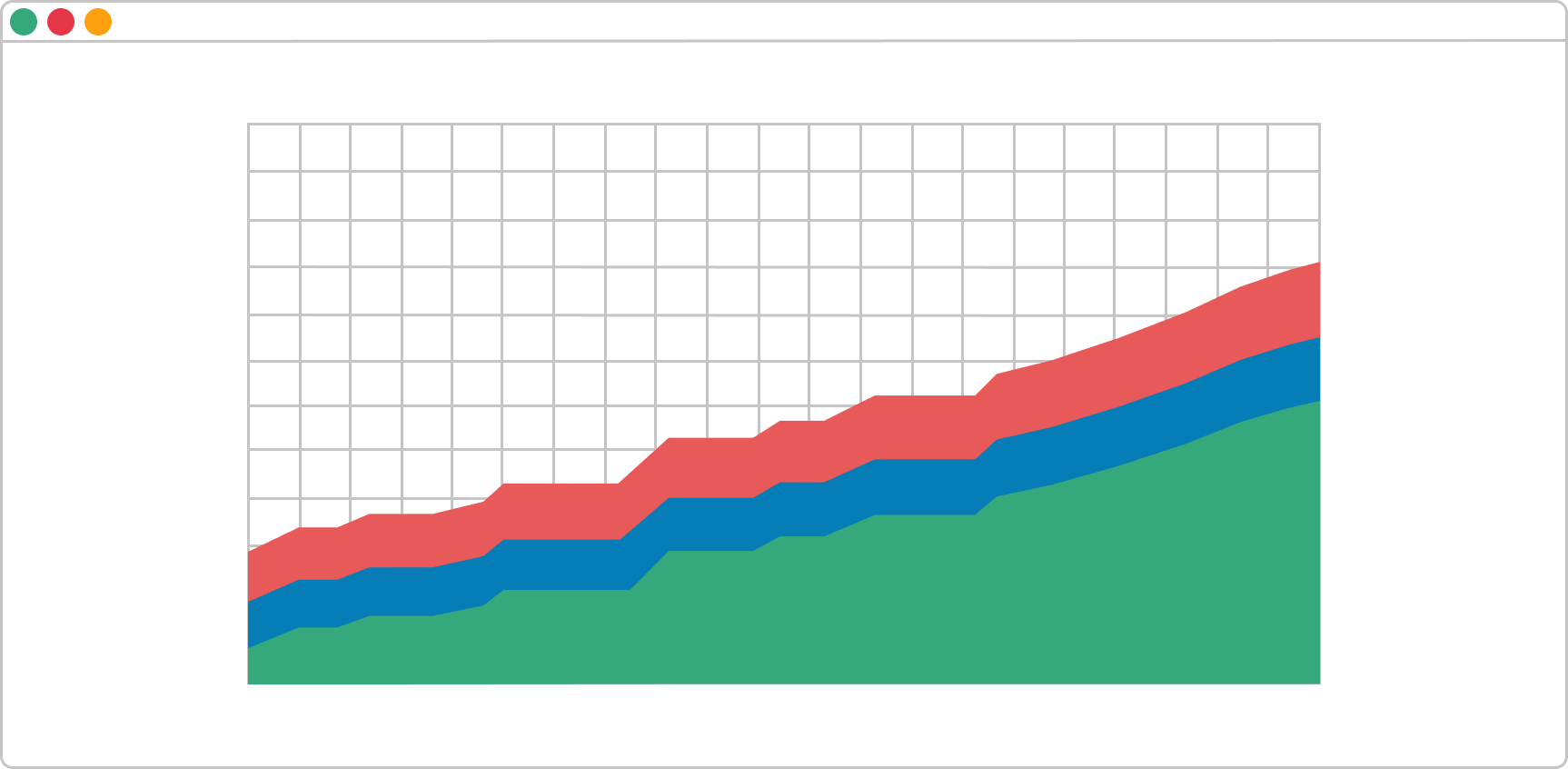
The top line of each band on the cumulative flow chart represents the entry point of tasks in the respective stage of your Kanban board, while the bottom one shows when it leaves it. If a line becomes flat, nothing arrives in the corresponding stage, or nothing is leaving it. Using a CFD, you can get an idea of how long your tasks' approximate cycle time is. This is possible by measuring the horizontal distance between the top line of the first stage on the cumulative flow diagram and the bottom line of the last "in progress" stage. The number of days/weeks/months that have passed is the approximate average cycle time of your team’s assignments for the time frame.

**The distance between the lines of a CFD will show you the problems of your workflow.**

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 band are progressing 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.

**The band are rapidly narrowing:**

### Unused capacity on a CFD

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 graph is rapidly widening:**

### Bottleneck on a CFD

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. There are many possible actions to resolve this issue. However, if this is not generated by a dependency on external stakeholders, you should reconsider the existing [WIP limits](https://kanbanize.com/kanban-resources/getting-started/what-is-wip/) on your Kanban board and focus on finishing tasks that are in progress before starting new ones. You should be aware of a possible 4th scenario as well – the bands are going down**. If any band on your cumulative flow diagram goes down, the diagram is incorrect.**A task should never disappear from your workflow. In conclusion, the cumulative flow diagram is an advanced analytic tool that will give you an accurate picture of how stable your process is and how efficient your team is. You can get a piece of actionable advice on where you need to focus your efforts in order to improve your process in a single glance.