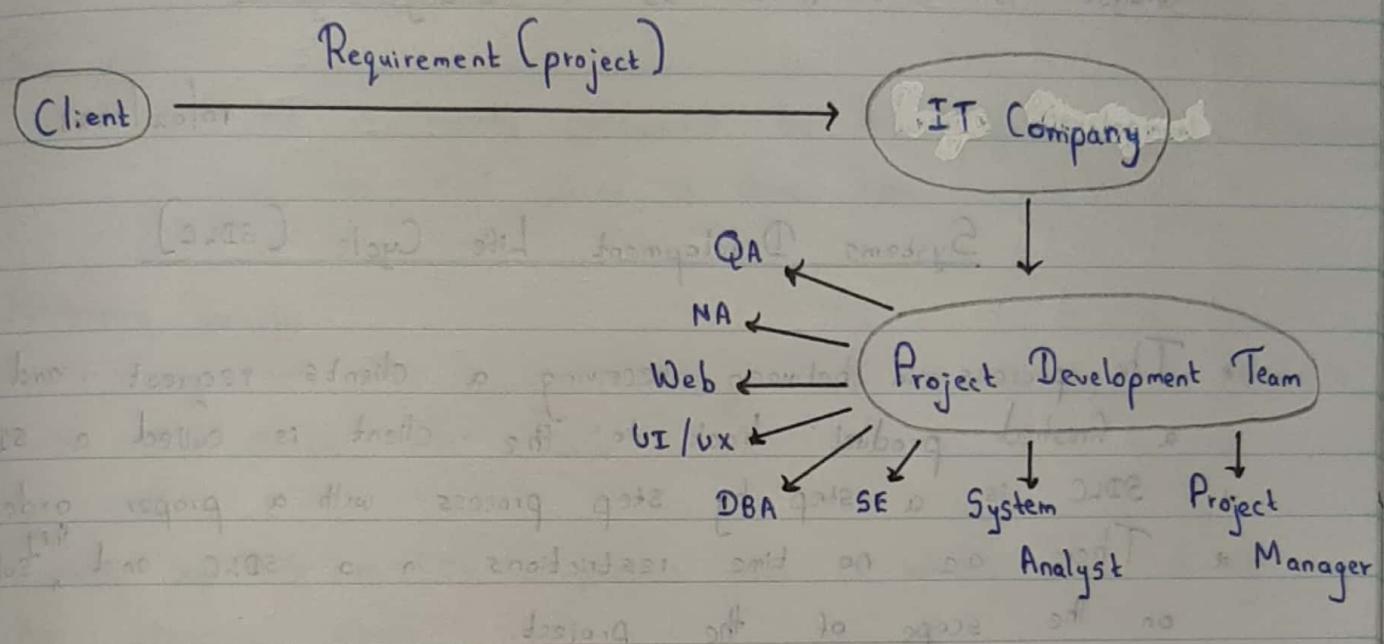


## SE 201.3 - Systems Analysis and Design

### System Analyst



- \* In a project development team except the project manager, the system analyst is the only person who is not directly involved with the front end & back end development of the software.
- \* System analyst notes the requirements and the business expectations of the client to come up with the software design package and gives software specifications that the rest of the development team can understand.
- \* System analyst develops the concept and becomes the bridge between the development team and the client. (The client does not have the ability to instruct the development team on what programming language should be used to develop the system and the size of the database)
- \* The system analyst should come to an agreement with the client on the available technologies and should properly understand the clients requirements in order to properly convey them to the development team.
- \* System analyst needs to confirm whether the final product is acceptable or not before presenting it to the client.

- \* When the system is being used there might be changes that needs to be facilitated later on due to change in mind of the client. System analyst will then have to co-ordinate with the client and make those necessary changes. Therefore, the system analyst is also known as a change agent.

10/02/2023

## Systems Development Life Cycle (SDLC)

- \* The process in between receiving a clients request and giving a finished product back to the client is called a SDLC.
- \* SDLC is a step by step process with a proper order.
- \* There are no time restrictions in a SDLC and <sup>they</sup> solely depends on the scope of the project.

Planning      Analysis      Design      Implementation

Client request      Completed product

- \* Some SDLC have more than four phases. The only difference in those are that the implementation phase is divided in to sub phases.  
Ex: testing, building, deployment, maintenance

Phase 1 - Planning

### i) Feasibility study

- \* This is the starting point of the project after receiving the clients request. The project manager and the system analyst check the feasibility of the project mainly under the categories of finance, technology, operation and schedule.

- \* If the conclusion of the study is that its not

feasible then the System analyst should come up with an alternate plan. But, if the System analyst can not come to a new agreement with the client then they will have to reject the project.

## 2) Time plan

- \* This is mainly prepared by the project manager. Allocating time for each phase is done here.

## 3) Allocating people

- \* Here the project manager will allocate the suitable people for each task.
- \* Coming up with an overall project plan is what's done during the planning phase.
- \* Main deliverable of this phase is the feasibility study report.

## Phase 2 - Analysis

- \* An detailed plan of the project is prepared during this phase. For this a more in depth requirement gathering should be conducted.
- \* Then analyse the requirements and decide on what should be included and what should not be included. To analyse the requirements we can use user stories and use cases.
- \* After that create the initial design using use case diagrams, data flow diagrams (DFD), flow charts etc.
- \* System analyst will be responsible for all these tasks along with co-ordinating with the client and double checking everything with him.
- \* At the end of the analysis phase system analyst comes up with the project proposal which is the main deliverable of this phase.
- \* He then should show it to the client and get his approval with his signature.

## Skills needed to be a System analyst

### 1) Interpersonal skills

- └─ Communication Skills
- └─ Language Skills

### 2) Analytical Skills

### 3) Decision making

### 4) Creativity

### 5) Knowledge in both IT and business fields.

### 6) Should be a self learner.

- System analysts will have to work with clients in different fields like government, aviation, plantation etc. Therefore, he needs to have at least a basic idea about that respective field to get the correct idea about what the client requires.

## Phase 3 - Design

\* The following tasks will be needed to be completed during the design phase.

- 1) Designing the interfaces (can use wireframes)
- 2) Designing the database.
- 3) Designing the network.
- 4) Hardware specifications - Infrastructure.
- 5) Pseudocode (structure of the actual code)

\* System analyst is the one responsible for this phase as well.

When he gives the initial idea the developers in the team can work based on that.

\* Main deliverable is Software requirement specification. This includes everything from databases to interfaces to pseudocodes. This will be the input on to the next stage.

## Phase 4 - Implementation

- \* The following tasks will be completed during this phase.
  - 1) Coding
  - 2) Testing
  - 3) Debugging
  - 4) Deploying
  - 5) Providing training to users
- \* System analyst is released during this phase as he is of little use in completing the phase.
- \* Main deliverable is the working product.
- \* Can give a user manual (instructions on how to use the product) and an operational manual (which specifies who does which activity) along with the completed product.

## Feasibility Study Report

- \* It's an evaluation we perform to check whether the clients requirements are possible or not and to come to the conclusion of accepting or rejecting the project.
- \* Before starting a business, you would have to consider the feasibility of your idea in the following aspects.
  - 1) Finance
  - 2) Work load (Scope of work)
  - 3) Man power } resources
  - 4) Technology
  - 5) Time
  - 6) Legal concerns

## Exercise

① Discuss whether below projects are feasible or not:

1) Upgrading a university's building which was built in the 1970s  
- was outdated.

- Applicable - Finance, Resources, Time
- Not applicable - Legal
- Reason - No need of legal advise for upgrading a building within your own premises.

2) Technical team is working on creating a self-driving car with top speed up to 300 KMPH.

- Applicable - Legal, Finance, Resource, Time
- Not applicable - Non

3) Starting up a restaurant in your town as a family business.

- Applicable - Finance, Resource, Time, Legal
- Not applicable - Non

\* Depending on the nature of the project, the type of resources needed will change.

## Feasibility analysis of a software development project

1) Technical feasibility

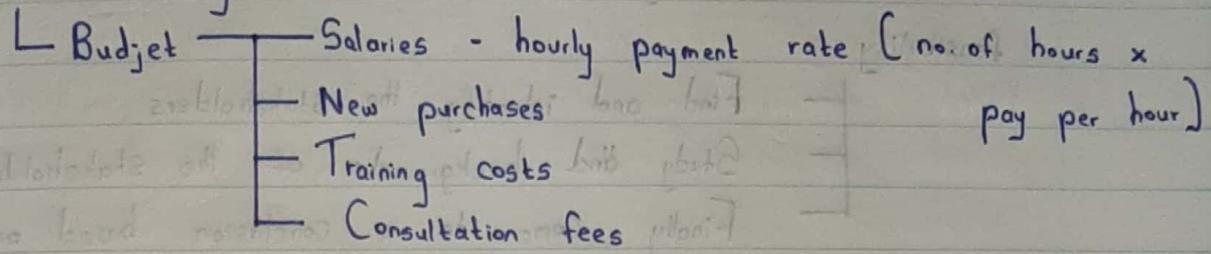
- Checks whether the employees have sufficient skills and knowledge.
- Possibility to upgrade available skills and knowledge.

    - by outsourcing

    - by workshops or training programs  
(Availability of necessary hardware and software tools)

Infrastructure

## 2) Economic feasibility



\* If the available budget is not sufficient

↳ Can expand the budget.

↳ Can reduce the scope of the project with the clients

## 3) Schedule feasibility

↳ Time (Checks whether the allocated time is sufficient to finish the project)

↳ Scope = Time. (Work load should match available time)

\* If the allocated time is not sufficient

↳ Reduce scope of the project.

↳ Request to increase the allocated time. (Budget will increase)

## 4) Operational feasibility

↳ Stakeholders (people interested in what we are doing and gains a benefit out of it.)

Technical team

Software project Stakeholders

Investors (bank,

organizers, financial

providers etc.)

System users

/ End users

System owners

(Sometimes the owner and user are different.  
ex - a cashier in a Supermarket)

\* Mainly checks how well the end user will accept our final product.

- \* A stakeholder analysis needs to be conducted in the operational feasibility.
  - Find and identify the stakeholders.
  - Study the backgrounds of the stakeholders.
  - Finally come to a conclusion based on the study.
- \* An evaluation based on the feasibility study is a must on the feasibility study report.
  - Analyse each category and give a conclusion at the end for each.
  - Finally give one overall conclusion at the end of the report.
- \* If the conclusion of the study is that the project is not feasible then need to mention the steps that needs to be taken next.

23/02/2023

### Selecting a Project Methodology

- \* An alternative process that can be used upto practise the four phases in a SDLC is called a project methodology.
- \* There are different types of project methodologies.
  - 1) Waterfall development
  - 2) Parallel development
  - 3) V-model ( variation of the waterfall development )
  - 4) Rapid Application Development (RAD)
    - Iterative development
    - Prototyping
  - 5) Agile development

## Waterfall Development

**Planning**

**Analysis**

**Design**

**Implementation**

**System**

\* Waterfall development follows a sequential approach and it's the first methodology introduced.

\* Still used in certain instances but now mostly agile development is used.

\* Need to follow each step in order like a sequence. Should start from the planning stage and complete it before going to the next stage. Can not go to the next stage without completing the current stage.

### Limitations in waterfall development

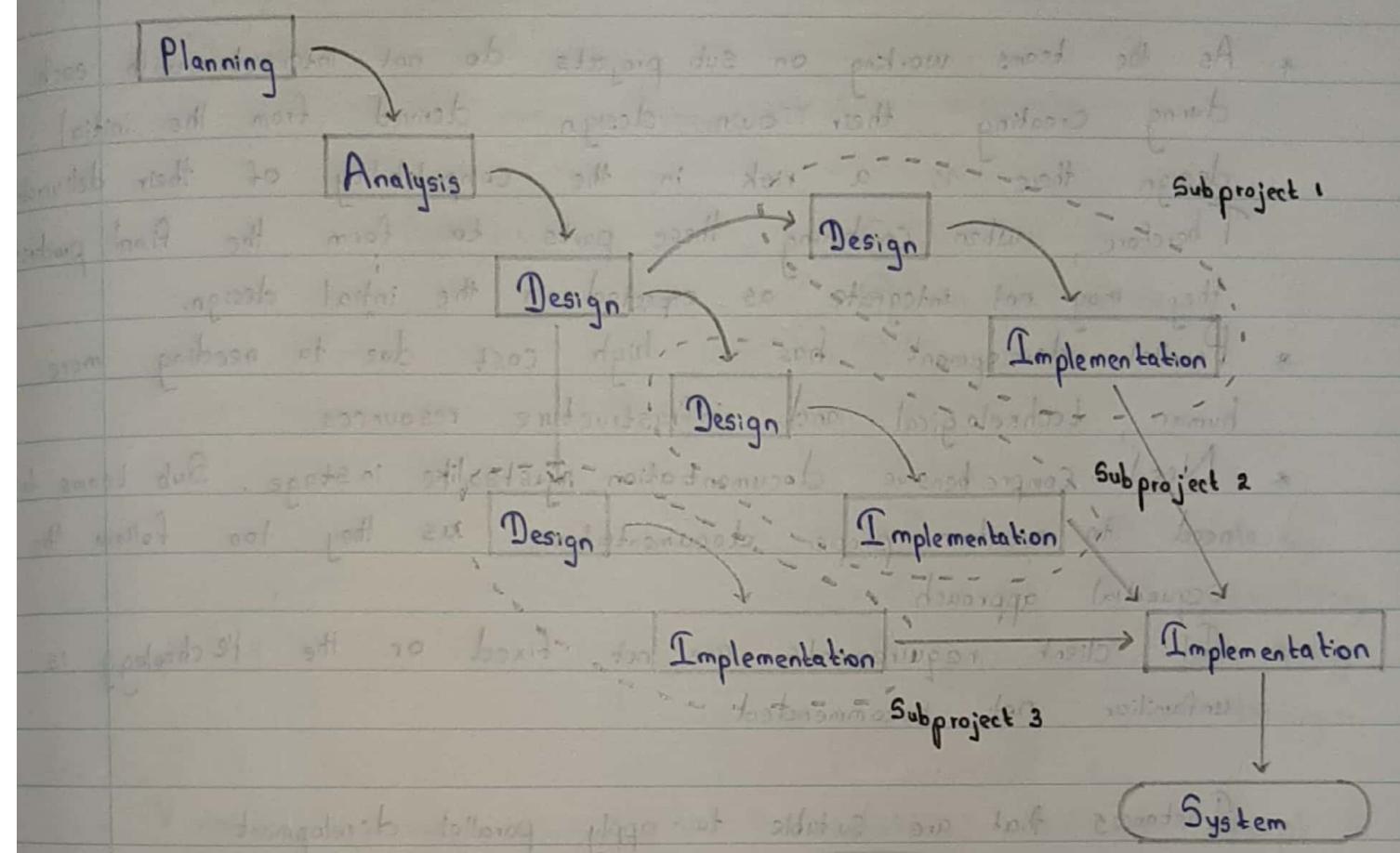
- \* The name waterfall is given because just like a waterfall it is very difficult to go back and redo things in the previous stages. Though not impossible it will consume a lot of time.
- \* At the end of each stage need to produce a comprehensive document which contains the entire workload. This will take a lot of time and if by any chance you have to change anything in a previous stage you will have to amend all the documents accordingly.
- \* Unwanted amounts of time will be wasted on documentation.

- \* Not Suited in <sup>the</sup> case of upgrading or maintaining an existing System as they might not need to follow all steps in a SDLC.
- \* Need sufficient time to complete the project and can not perform parallel actions. Therefore, not suitable for projects of less than one month.
- \* Also not suited for <sup>very</sup> long term projects that take more than a year to complete.
- \* Due to the sequential approach all client requirements will be freezed in the analysis stage when proceeding to the next stage. But when a project is taking a lot of time or maybe due to a change in the client the system analyst will have to change the requirements. In such a scenario where the client expectations can not be kept fixed and the technology used is unfamiliar the waterfall development is not suited.
- \* Till the end we do not get a working deliverable. So, can not see a working product mid way.

### Advantages in waterfall development

- \* Suitable if starting from the initial stage.
- \* Can easily proceed on to the next stage.
- \* As comprehensive documentation is available on each stage can easily transfer knowledge.
- \* As the client requirements are freezed at the analysis stage we will have less re works and the changes in scope of the project can be minimized.

## Parallel Development



- \* Parallel development was introduced as an amendment due to the limitations in the waterfall method.
- \* Follows the same sequential approach where starting from the planning stage need to complete each stage in order.
- \* After coming up with the initial design of the system of the project is broken down into sub projects. Amount of sub projects depend on the scope of the project. Different teams will work on these sub projects at the same time parallel to each other.
- \* Each team working on a sub project is tasked with designing and developing the sub project.
- \* Teams in sub projects do not interact with each other and work independently.
- \* Finally all the deliverables of sub projects are combined and integrated together to form the complete system to get the final product.

## Limitations in parallel development

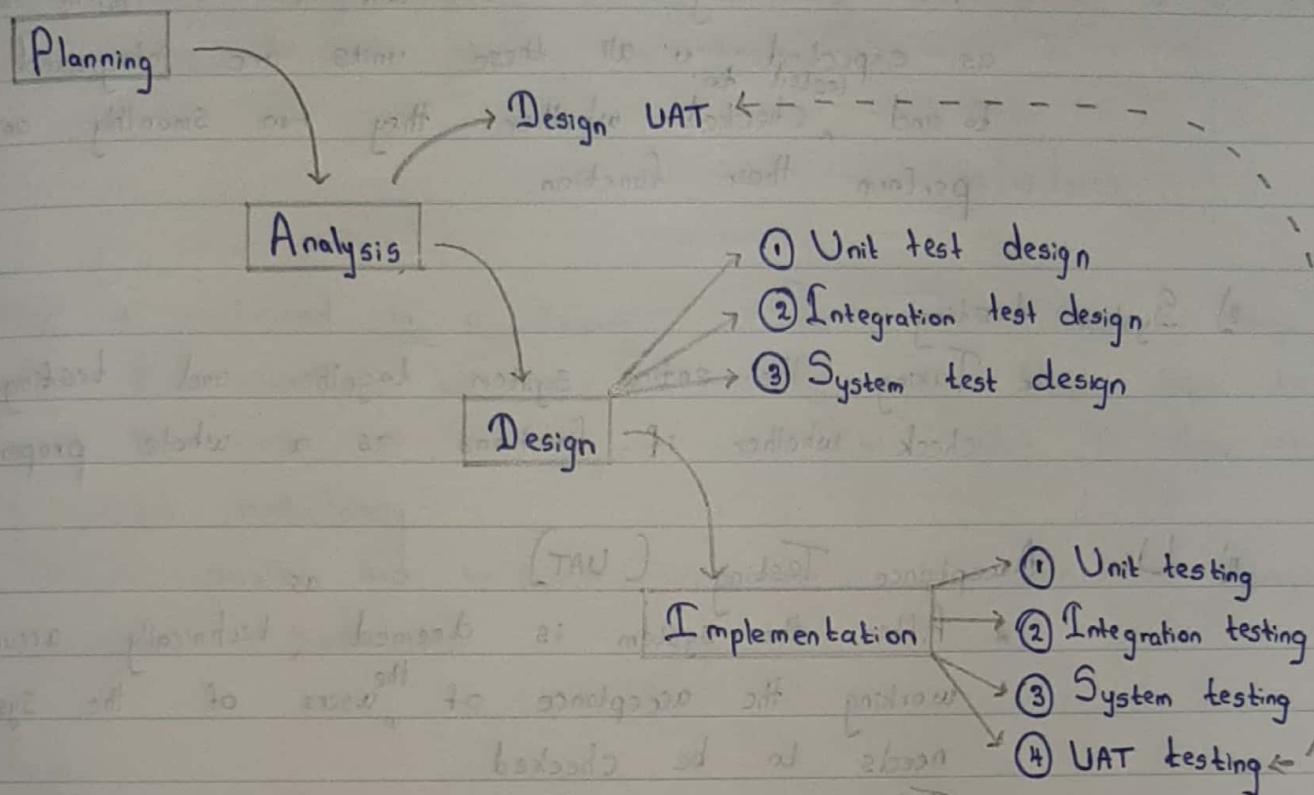
Example ① Iphone 9

- \* As the teams working on sub projects do not interact with each other during creating their own design derived from the initial design there is a risk in the compatibility of their deliverables. Therefore, when combining these parts to form the final product they may not integrate as expected in the initial design.
- \* Parallel development has a high cost due to needing more human, technological and infrastructure resources.
- \* Needs comprehensive documentation in each stage. Sub teams too need to maintain proper documentation as they too follow the sequential approach.
- \* If client requirements are not fixed or the technology is unfamiliar not recommended.

## Instances that are suitable to apply parallel development

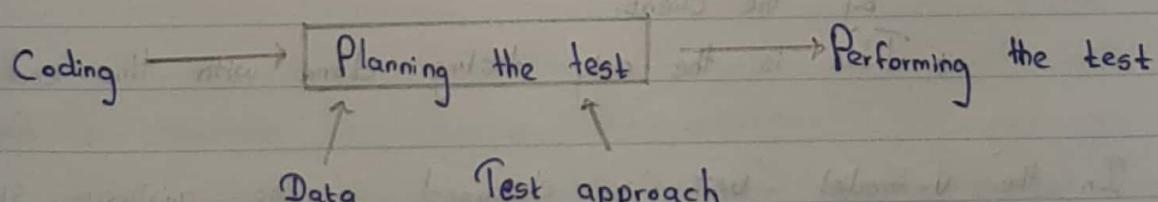
- \* When allocated time is not suited to apply waterfall method as parallel development comparatively takes less time.
- \* When you have more budget, more resources, more scope, a more complex project with fixed client requirements you can go for the parallel development.

## V-Model



\* V-model is very similar to the Waterfall method except in planning the testing.

\* Normal method:



\* But in the V-model we do test planning from the initial stages.

### Types of testing

#### i) Unit testing

\* When the smallest unit in the system is completed you test it and check whether its working.

\* Like that unit by unit you perform the testing.

Topic - V

## 2) Integration testing

- \* After confirming that each unit in the System is working as expected all these units are integrated together to and checked whether they run smoothly and correctly perform their function.

## 3) System testing

- \* Taking the entire System together and testing it to check whether it functions as a whole properly.

## 4) User Acceptance Testing (UAT)

- \* After the system is deemed technically accurate and working the acceptance of the users of the System needs to be checked.
- \* For that a sample is given to the end users and their feedback is collected to understand how well the System will be accepted by its users.
- \* The result of this test determines the success of the System as the end user acceptance is what's expected by the client.
- \* This is the only testing done with the end users.

- \* In the V-model UAT is designed in the analysis stage but the test will be performed as usual in the implementation stage.
- \* Due to this there is a hypothetical link between the analysis stage and the end of the implementation stage.
- \* The name V-model is derived from this conceptual approach of the latter stages being linked with the initial stages.
- \* As all the tests are pre planned testing can be carried out without missing any client requirements even when the system analyst is not present in the implementation stage.

- \* All the limitations of the waterfall method is present in v-model except not missing anything during testing and saving a bit of time due to pre planning all testing.

02/03/2023

- \* SDLC is performed in a Sequential manner in a step by step order. But sometimes we have to alternatively apply those steps to meet the needs of our project and that's what is done using a project methodology.
- \* Waterfall, parallel and v-model are known as Structured project development methodologies.

### Exercise

- \* Compare and contrast the features of structured project development methodologies.

### Similarities

- 1) Difficult to go back and redo.
- 2) Need to maintain comprehensive documentation.
- 3) Can not see a working model till the end of the project.
- 4) Not suitable to use if user requirements are not finalized.
- 5) Suitable when starting the project from the initial stage.

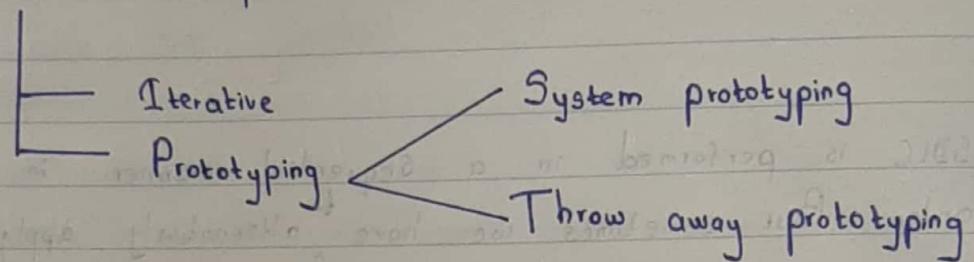
### Differences

- 1) Parallel and v-model take less time compared to the waterfall model.
- 2) Comparatively the cost is highest in the parallel model.
- 3) Using the v-model will enable testing to be done without missing any part unlike in the other two models.
- 4) Comparatively the parallel model is better for more complex projects.

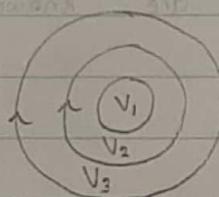
# Rapid Application Development (RAD)

• Rapid application development

Rapid application development



RAD : Iterative development



- \* In the iterative development the entire team completes the work in cycles or iterations by dividing the requirements into versions and completing all versions till all the client requirements are covered.
- \* After completing the planning & analysis stages the requirements of the client should be listed down in order according to their priority.

Example: • There are six client requirements named R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>, R<sub>4</sub>, R<sub>5</sub> and R<sub>6</sub>. Out of these six requirements R<sub>1</sub> and R<sub>3</sub> are the most important. Likewise all six requirements should be ordered according to their priority.

- R<sub>1</sub>, R<sub>3</sub>, R<sub>6</sub>, R<sub>5</sub>, R<sub>2</sub> and R<sub>4</sub> is the requirement list when ordered according to their priority.

- \* Now in the iterative development you select the requirements with the highest priority first. According to our example that is R<sub>1</sub> and R<sub>3</sub>. Next these requirements are studied in detail and a small system is developed based on these requirements.

- \* After that, version 1 is shown to the client to get his approval. With the approval of the client the first iteration will be completed and we can move on to the next iteration.
- \* Following the same procedure as in the first iteration the next set of requirements in the list is selected for development. According to our example that will be R<sub>6</sub> and R<sub>5</sub>. Version 2 is developed based on these two requirements.
- \* After completing the 2<sup>nd</sup> version you get the clients approval and proceed in to the next set of requirements R<sub>2</sub> and R<sub>4</sub>. With the completion of version 3 the project will be completed in three iterations.

### Advantages in the iterative development

- \* This method is suitable when the client requirements are unclear or not finalized as the team can first focus on the ones that are finalized and follow up with the rest.
- \* Even when the team does not have sufficient knowledge or the technology is unfamiliar this method can be used as there will be time to cover up the missing parts.
- \* Can be used on complex systems with a large scope that has a lot of requirements and functionalities as they can be broken down into small parts and enhanced through the iterations.

### RAD : Prototyping

- \* Prototyping is creating a dummy version of your system that can be shown to the client for him to get an idea on how the final product will look like.
- \* The dummy version is not a working product and its only use is receiving the approval of the client through feedback and verifying that all requirements have been met.
- \* After the dummy version is approved can directly move on

to the actual system.

- \* Generally prototyping requires more effort due to having to make a dummy version first.

### Advantages in prototyping

- \* Less reworking will be needed at the end of the system.
- \* Will have less errors at the end of the system.
- \* Suitable when having unclear requirements as there is always the chance to cross check with the client using the prototype.

### System prototyping

- \* In System prototyping the dummy itself will be converted to the final working system after making the necessary changes according to the clients feed back.
- \* The finalized dummy will be directly converted into the working product.

### Throw away prototyping

- \* In throw away prototyping the final dummy version will not be used as the final product.
- \* Based on the finalized dummy a separate system will be newly created while keeping the dummy version as it is.
- \* The dummy version will only be used for verification.
- \* The client will receive the newly created system as the final working product.

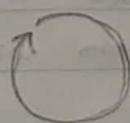
## Comparison between System prototyping & throw away prototyping

- \* In general both methods can be used for complex Software development projects.
- \* System prototyping takes less time.
- \* In throw away prototyping cost is higher due to having to make two versions. But if security is a priority this method is recommended as the final product is not the final dummy revealed to the users.

## Agile Development Methodology

- \* Agile development is the most popular methodology in the current world.
- \* It is a collaborative effort with a lot of emphasis on teamwork.
- \* Working models are frequently delivered to the client in this methodology.
- \* Suitable for short term projects and when the team is familiar with what needs to be done.
- \* Out of several frameworks Scrum is the framework most commonly used to apply agile to a project.

### Scrum Framework



Sprint

- \* Scrum is a cyclic approach where work is completed in cycles.
- \* In a cycle one or two functions of the product will be focused.
- \* A cycle lasts for a period of 2-4 weeks and never exceeds one month.
- \* At the end of a cycle the client will definitely receive a working model.
- \* One cycle is formally known as a sprint.

- \* After completing the first <sup>Sprint</sup>, the team can move on to the next one.
- \* Client will always get updates and a high level of user engagement will be maintained.
- \* The team working on a Sprint is formally known as a Scrum team.
  - Scrum master.
  - Product owner.
  - Development team.



- \* Scrum master is a person knowledgeable in the Scrum guidelines who ensures that the team follow the correct guidelines and overlooks the team just like a project manager.
- \* Product owner is the person with a clear idea about the product with a 360° view on it.
- \* Product owner can be a person from the client side or a member of the development team according to the situation.

Example: Some IT Companies like game development companies produce their own products in which case the product owner will be a member of the development team.

- \* If an external client is involved then the product owner will be a person from their side.
- \* Development team is the group of individuals that will be developing the product.
- \* Normally a scrum team will not exceed ten members.

- \* Though agile was invented in the software development field it is now used in a lot of other fields as well.
- \* In agile the process is broken down into small parts and the team works to achieve those small milestones and complete the project.

### Core values in agile

- \* The four main values of agile is discussed in the agile manifesto.

- 1) Teamwork and communication is the core in agile. The development of the software is done as a collaborative effort with good communication among team members. The entire work will be visible to all team members maintaining a lot of transparency among team mates.
- 2) More focus will be on developing the software and only the most important points will be documented.
- 3) Client feedback is highly valued and the team will communicate with the client and end users frequently.
- 4) Agile development is always open for changes and will welcome any changes required by the client to the initial plan.

- \* There are several frameworks used to apply agile in to a project.

Example: Scrum, Kanban, Extreme programming (XP), Lean

## Continuation of scrum framework

- \* In a sprint the task which the development team focuses on completing is called a user story which can be one functional unit.
- \* In some situations more than one user story is focused in a sprint. But normally only one user story will be in a sprint.
- \* At the end of each sprint a working functional unit is produced and presented to the client. Like that the entire product will be completed part by part.
- \* In a sprint all the user stories are listed according to their priority and ones with the highest priority will be selected first for the sprint.
- \* A sprint consists of four main components.

- 1) Sprint planning
- 2) Daily scrum
- 3) Sprint review
- 4) Sprint retrospective

- \* Sprint planning
  - \* At the beginning of the sprint the entire team will get together to plan the sprint and its scope.
  - \* To delegate tasks for the sprint a product backlog will be used.
  - \* Product backlog contains all the functions of the entire product and it can be used to identify user stories.
  - \* A portion of tasks will be selected according to their priority from the product backlog to be completed within the sprint.
  - \* That selected portion will be called a sprint backlog and will be the scope for that particular sprint.

## 2) Daily Scrum

- \* During the entire course of the sprint the team members will meet every day in the morning for a quick meeting of 15 minutes which is called a daily scrum or a daily stand up.
- \* This meeting is conducted before starting the work for the day and each member should give an update from their side about their progress, limitations or issues they faced and what they are going to do on that day.
- \* These meetings help on keeping all team members up to date on the progress of the project.

## 3) Sprint review

- \* A sprint review is a meeting conducted at the end of the sprint with the participation of the product owner, scrum master and team leads.
- \* A discussion will take place to identify what was covered during the sprint and what's left from the sprint.
- \* This will be a review at the end of the sprint to discuss what actually happened during the sprint.

## 4) Sprint retrospective

- \* The entire development team will take a meeting to discuss what went right, what went wrong and what can be improved from the sprint.
- \* Insights about the next sprint and what should be included in it from the previous sprint will be discussed during this meeting.
- \* Scrum is only a framework used to apply agile in to a project and therefore, there are differences between Scrum and agile. They are not the same model.

## Kanban Framework

- \* Kanban is a popular framework used to implement agile that follows the main principles of agile.
- \* In kanban a large user story that can be divided into several smaller stories called an epic is used.  
Example: The user management component (epic): payment gateway can be further divided into stories of payment details, payment authentication, payment options etc.
- \* In kanban workers are not restricted to time periods but given milestones to be completed before a deadline.
- \* The work that is yet to be finished is called work in progress (WIP). Normally there is a limit for daily WIP.  
If due to any reason it exceeds the limit the team manager will have to take action in order to overcome those bottlenecks.
- \* The target is to reduce the WIP and complete the project component by component.
- \* The kanban board is used in kanban using software like Trello where you breakdown your work into the to do column.

To do	WIP (Dev)	WIP (UI/UX)	WIP (Testing)	Completed
1	1		1	
2				
3				
4				
5				

- \* These kanban boards are mainly used to track the progress of the project.
- \* The columns in the board will change depending on the nature of

the project.

- \* In the above board task 1 had first appeared in the to do list then moved on to the other columns according to its current progress. And finally it will appear in the completed column if it successfully passes the testing or will be moved to the development column if failed.
- \* A task can only appear in one column at a time unless multiple people are working on the same task and each of their progress needs to be shown.
- \* Each task in the board is called a kanban card.
- \* When a developer is assigned a task they will make a kanban card in the board and move it to the relevant column.

#### Differences between kanban & scrum

- 1) The time limit is fixed to 30 days in scrum whereas there is no fixed time limit in kanban.
- 2) In scrum there is a pre defined time box called a sprint. But, even though kanban follows a cyclic approach there is no sprint.
- 3) In kanban a full approach is taken in comparison to scrum where each task has a fixed time period. Kanban does not control people and when a task is assigned they just have to finish it before the deadline.
- 4) Kanban uses a kanban board to track the progress of the entire project.
- 5) There is a scrum master in scrum and there is no such person in kanban. But both have the same development team.
- \* In agile you can see a mix of kanban & scrum with the procedure of scrum and the use of kanban board applied together.

## Exercise

- \* Note down the key factors to be considered when selecting a methodology for a project.
  - 1) Available time and resources. (human, financial, technological etc.)
  - 2) Scope of the project.
  - 3) Documentation requirements.
  - 4) How frequent the client interactions will be with the team.
  - 5) Clarity of user requirements.
- \* If the project is short term better <sup>to</sup> use Agile rather than using a structured project development methodology.
- \* If the user requirements are not clear or fixed go for agile or prototyping.
- \* If the technology being used is familiar can use a structured project development methodology or else if unfamiliar go for agile.
- \* If the work is too complex waterfall is recommended. (depends on the user requirements)
- \* If the product should maintain its privacy till release then System prototyping is not recommended. Can use any other method.

## Planning an information system project

### Time and Scope management

- \* Depending on the project development methodology used the way the project is going to be performed will differ.
- \* Regardless of the methodology used a time plan will be required.
- \* An idea on the most suitable duration for each phase of the SDLC will be required for this.
- \* Normally the project manager will come up with the time plan relying on his experience.
- \* But if the person lacks experience there is a method to make a time plan using industry standards.
- \* This method is a guideline and does not require strict following.

Example : There are 100 weeks to complete the project. It can be divided as follows,

Planning	Analysis	Design	Implementation
15%	20%	35%	30%

15 weeks 20 weeks 35 weeks 30 weeks

- \* These guidelines are followed by people to the industry.
- \* If a product is directly developed after receiving the initial design without consulting with the client although we get a working product there is a high chance for the user to reject it.
- \* That is the reason for 70% of the time being allocated for the first three phases before starting the actual coding part.

## Identifying the scope of a project

- \* Scope is a boundary line that is used to determine what needs to be done during the project and what can be eliminated from the project.
- \* There are no industry guidelines to determine the scope of a project and therefore, it depends fully on the person.
- \* The tasks within the scope (inside scope) and the tasks outside of our work (outside scope) should be clearly identified.  
 Example: If the given task is registering students at FOC the other faculties FOB & FOE can be directly ignored.
- \* There should be an objective to the project and after identifying the objective it needs to be checked whether all the necessary tasks are covered within the scope of the project.
- \* If not they can be clarified using a work breakdown structure (WBS)

### Work Breakdown Structure (WBS)

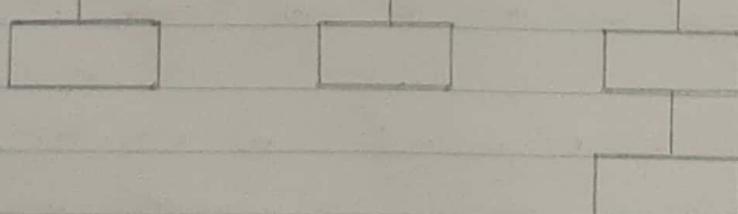
- \* WBS is a tree-like structure.
- \* After identifying the project objective focus should be shifted on to the deliverable (outcomes) of the project.
- \* Breakdown the outcomes in to smaller parts and cover the entire project in that way.

Final outcome

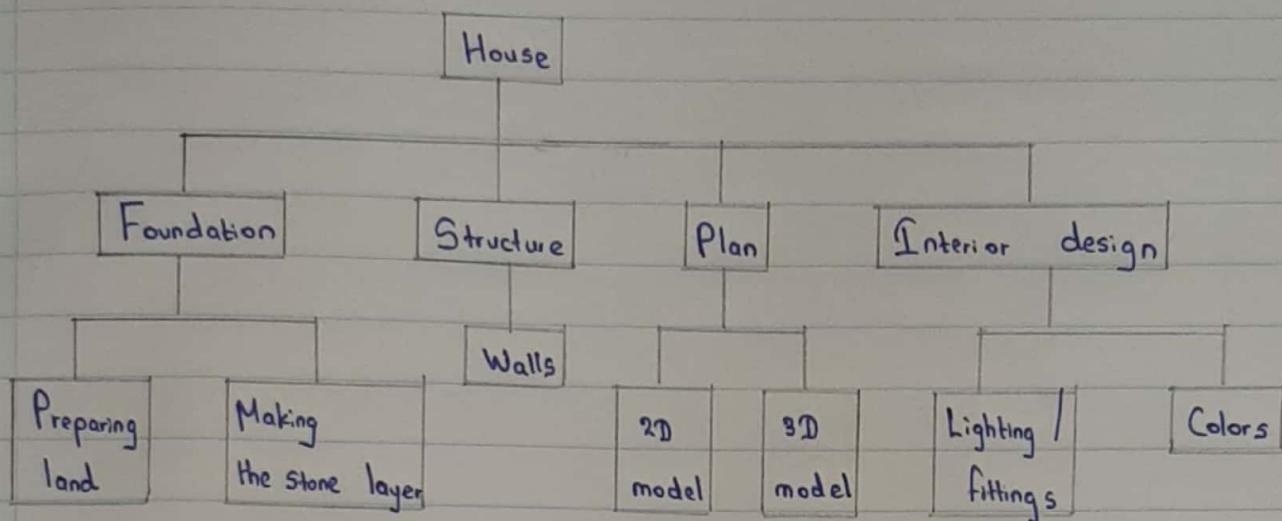
deliverable

deliverable

deliverable



Example : Constructing a new house.



- Like the above example Software development too can be divided into deliverables. The main System can be divided as front-end and back-end.

