

6/2/23 Software : Software is more than a program code. A program is an executable code which serves some computational purpose. Software is considered to be a collection of executable programmable code, associated libraries and documentations. Software when made for a specific requirement is called a software product.

Engineering on the other hand, is all about developing products using well defined scientific principles and methods. So we can define SE as an engg. branch associated with the development of software products using well defined scientific principles, methods and procedures. The outcomes of SE is an efficient and reliable software product.

IEEE defines SE as the application of a systematic, discipline, approach to the development, operation and maintenance of software.

Need of SE

The need of SE arises because of higher rate of change in user requirements and environment on which the software is working.

- (i) Large Software : It is easier to build wall than a house or building likewise as the size of software become large engineering has to step to give in a scientific process.
- (ii) Scalability : If the software process were not

based on scientific and engg. concept, it would be easier to recreate new software than to scale an existing one.

(3) Cost : As hardware industry has shown its skills and huge manufacturing has lowered down the price of computer software and hardware but the cost of software remains high if proper process is not adopted.

(4) Dynamic Nature : The always growing and adapting nature of software largely depends upon the environment in which the user works. If the nature of software is always changing, new enhancement need to be done in the existing one.

This is where SE plays a good role.

(5) Quality Management : Better process of SD provides better & quality Software product.

8/2/23 Software Development Life cycle

Different types of

A software lifecycle model also called a process model is a descriptive & diagrammatic representation of the software lifecycle.

- Lifecycle model represents all the activities required to make a software product transit through its lifecycle phases. It also captures the order in which these activities are to be undertaken.
- Lifecycle model maps the different activities performed on a software product from its inception to retirement. Different lifecycle models may map the basic development activities to phases in different ways.

Need for a Software Lifecycle Model

The development team must identify a suitable life-cycle model for the particular project without using of a particular lifecycle model the development of a software product would not be in a systematic and discipline manner. When a software product is being developed by a team there must be a clear understanding among team members about when and what to do.

Different Software Lifecycle Models

Many lifecycle models have been proposed so far. Each of them has some advantages as well as some disadvantages. A few important and commonly used lifecycle models are :

- 1) Classical waterfall model

2) Iterative Waterfall Model

3) Prototyping Model

4) Evolutionary Model

5) Spiral Model

1) Classical Waterfall Model

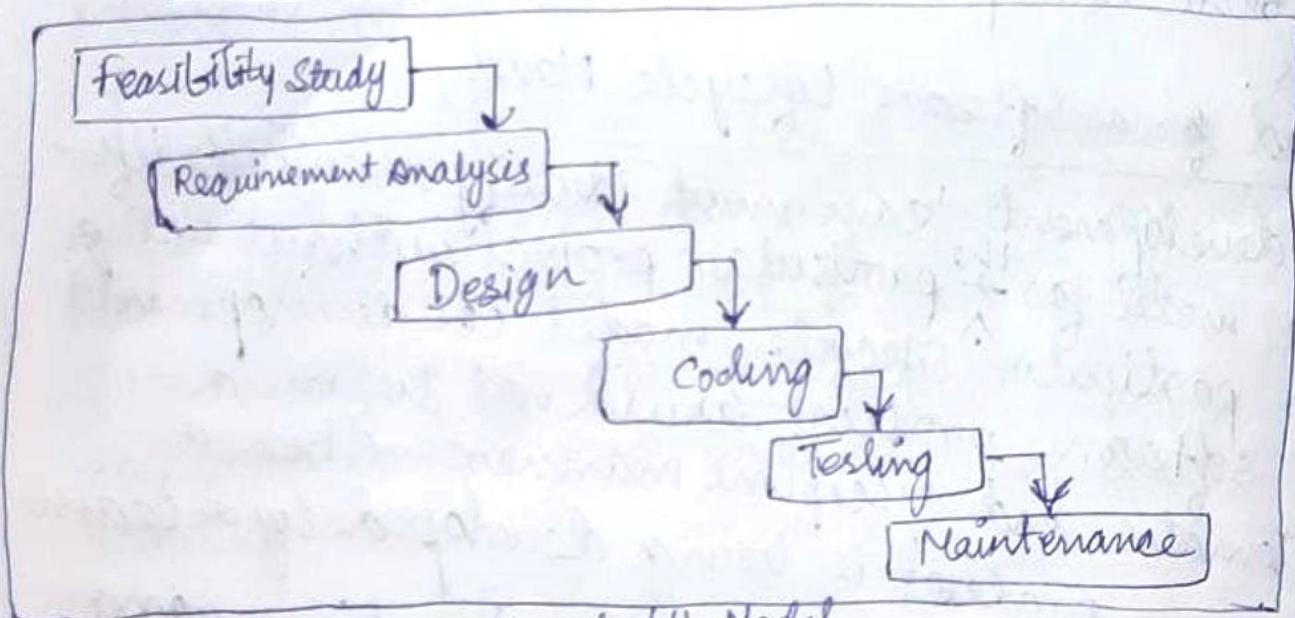


Fig: Classical Waterfall Model

It is the most obvious way to develop software. The Classical waterfall Model is elegant, it is not a practical model in the sense that it cannot be used in actual development project. Thus this model can be considered to be a theoretical way of developing software. But all other lifecycle models are essentially derived from the classical waterfall model. So in order to be able to appreciate other lifecycle models

it is necessary to learn the classical waterfall Model.
Classical Waterfall Model divides the lifecycle into
the following phases:

1) Feasibility Study: the main aim of feasibility study is to determine whether it would be financially and technically feasible to develop the product.
At first project managers or team leaders try to have a rough understanding of what is required to be done by visiting the client side. They study different input data to the system and output data to be produced by the system. They study what kind of processing is to be done to these data and what they look at the various constraints at the behaviour of the system.

After they have an overall understanding of the problem they investigate different solutions what are possible then they examine each of the ~~they examine diff~~ solution in terms of what kind of resource required, what would be the cost of development and what could be the development time of each solution.

Based on these analysis they pick the best solution and determine whether the solution is feasible financially and technically. They check whether the customer budget would meet the cost of the product and whether they have sufficient expertise in the area of the development.

11/2/23

Requirement analysis and specified specification

The aim of req. analysis and specification phase is to understand the exact requirement of the customer to document them properly. This phase consists of two distinct activities :

- i) Requirement gathering & analysis
- ii) Requirement specification

The goal of requirement gathering activity is to collect all relevant information from the customer regarding the product to be developed. This is done to clearly understand the customer requirements so that incompleteness & inconsistencies are removed.

The requirements analysis activity is begun by collecting all relevant data regarding the product to be developed from the users of the product and customer from the customer through interviews & discussions.

The data collected from such a group of users usually contain several contradictions and ambiguities since its user typically has only a partial & incomplete view of the system. ∴ it is necessary to identify all ambiguities & contradictions in the requirements and resolve them through further discussions with the customer. After all ambiguities, contradiction, inconsistencies & incompleteness have been resolved and all the requirements properly understood, the requirements specification activity can start. During this activity the user requirements are systematically organised into a software requirement specification document. The customer requirements identified during the requirement gathering & analysis activity are organized into an SRS doc & the important components of this of this document are functional requirement and non functional requirement of the goals of implementation.

(iii) Design : Traditional Design approaches
OO Design approach. The goal of the design phase
is to transform the requirements specify in the
SRS document into a structure that is suitable
for implementation in programming language.
In technical term, during the design phase the
^{architecture} software is derived from the SRS document.

To distinctly different approaches are available:

i) Traditional Design Approach: It consist
of two different activities. A structure analysis
of requirement specification is carried out
where a detail structure of the program
is analysis problem is examined. This is
followed by a structure design activity.

The during structure design the results
of st. analysis is transformed into System
Design.

ii) OO Design Approach.

In this technique various objects that occur in the problem domain or the solⁿ domain are first identified & the different relationships that exist among these objects are identified. The object structure is further refined to obtain the detailed design.

Coding : The purpose of the coding phase of Software Development of SD is to translate the Software Design into Source Code. Each component of the design is implemented as a program module. The end product of this phase is a set of program modules that have been individually tested. During this phase each module is

Maintenance : Maintenance of a typical software product requires much more than the effort necessary to develop the product itself. Many studies in the maintenance involves performing any one or more of the following 3 kinds of activities.

- ① Correcting Errors that were not discovered during the product development phase. This is called corrective maintenance.

Improving the implementation of an system and enhancing the functionalities of the system acc. to customers requirements. This is called perfective maintenance.

(iii) Porting the software to work in a new environment. This is called adaptive maintenance

13/2/23

Advantages

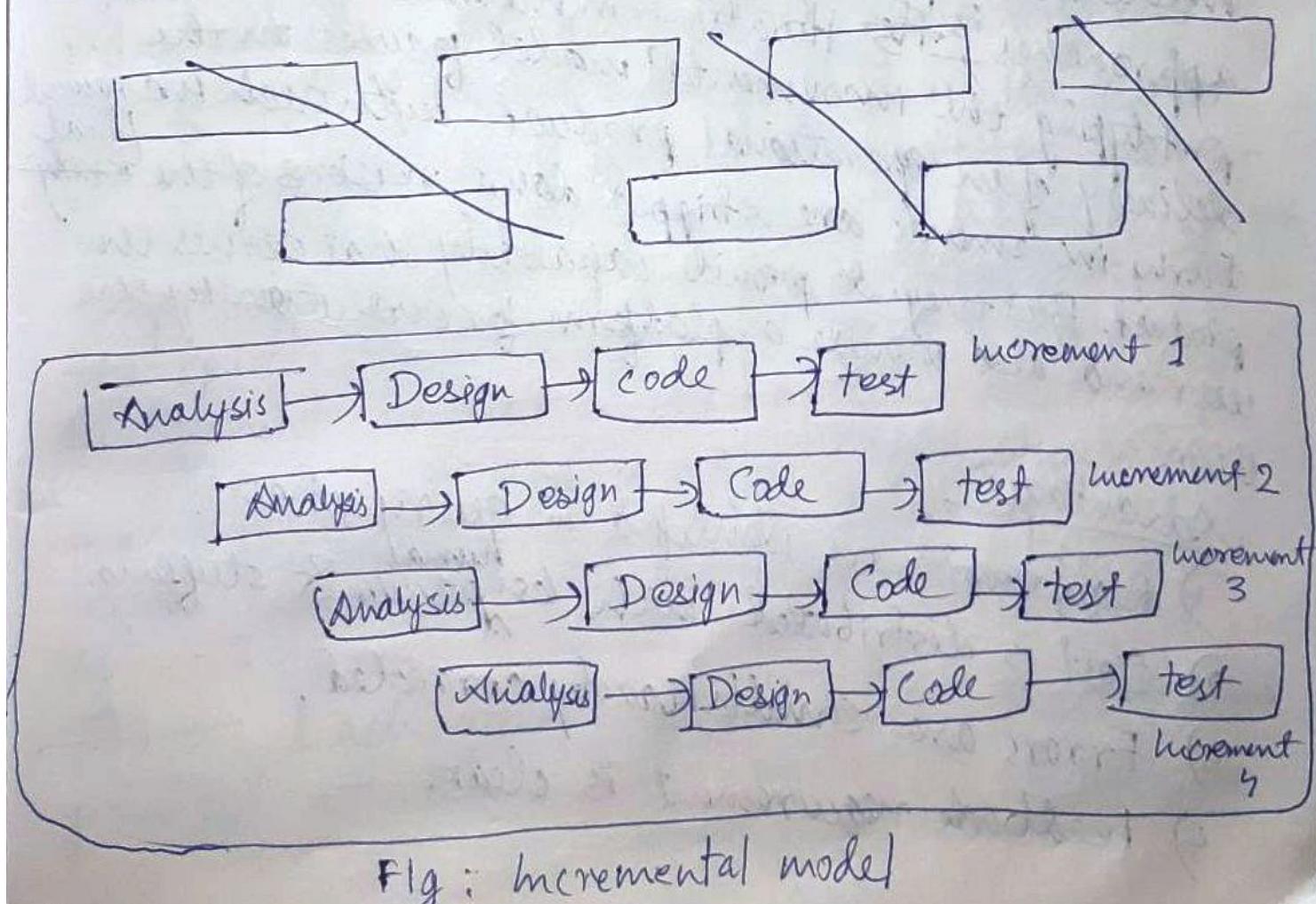
- 1) Easy to use
- 2) Easy to implement
- 3) Understood by non technical persons also.
- 4) Proper planning due to proper definition of the problem.

Disadvantages

- 1) Sequential in nature which prevents enhancements
- 2) Does not have iterations
- 3) Customer review or feedback is not reviewed before developing the product.
- 4) It is not suitable for complex, large & new projects.

Incremental model or Iterative model

The incremental lifecycle model is similar to waterfall model developed by European Space Agency in 1991. The system and software concepts and requirements are first identified and then the remaining activities of Software Development are repeated each time there is a new release of the software. The incremental model makes the unrealistic assumption that the system as well as software requirements remain stable. However, requirements tend to evolve due to changes in technology and experience.



This model is a combination of linear sequential model with iterative prototyping model.

This is popular model used by industry. The new release of system enhances the functionality of previously released system.

The first increment often referred to as core product and delivered to customer use and successively a plan is made for next increment to enhance its features and functionality. The process is repeated following the delivery of ~~each~~ ^{each} increment until the complete product is produced and shown in the above figure. An incremental process model is like prototyping or other evolutionary approaches ~~is~~ ^{is} iterative in nature but unlike prototyping the incremental model focuses on the delivery of an operational product with each increment. Early increments are stripped down versions of the final product. But they provide capability that serves the user and also provide a platform for evolution by the user.

Advantages

- 1) Requirements are identified in every phase.
- 2) Cost is distributed with less ^{human} power & staffing.
- 3) Errors are simultaneously corrected.
- 4) Feedback requirement is clear.

5) Testing is easy.

Disadvantages

- 1) Requires proper planning to distribute the work.
- 2) Total cost required for the development of the product is high.
- 3) Interface model should be well defined.
- 4) Too

16/2/23 Spiral Model

This model was developed by Barry Boehm in 1988 which is particularly used for large size products in the industrial site. It is developed by combining the good features of baseline management in the waterfall model; overlapping phases found in the incremental model and early versions of a system from the prototyping model. The basic assumption in the spiral model is that the form of software development cannot be completely determined in advance. Prototyping, risk analysis, simulation, modeling, financial analysis and benchmark are needed before committing to detailed design of a software system.

~~Also~~ Each cycle of spiral model has 4 basic principal activities:

- 1) Elaborate software entity objectives, constraints and alternatives.
 - 2) Evaluate alternatives relative to objectives and constraints and identify major source of risk.
 - 3) Elaborate the definition of software entities for a project.
 - 4) Plan the next cycle.
 - 5) Terminate a project if it is too risky.
 - 6) Secure management commitment.
- 1) Objectives to determination
and identify alternative solution
- 2) Identify and resolve risk

- 7) Review and plan for next phase

1) Objectives

- 3) Develop next version of the product

This model first identifies the objectives of the phase and the alternative solutions possible for the phase under consideration. Secondly, the alternative solutions are evaluated to select the best possible solution. For the selected solution, the risks are identified. A risk is any adverse circumstances that might affect a successful completion of the software product. Thirdly the developing and verifying the next version level of the product is carried out. Finally the results of the stages traversed so far with the customer & planning for the next iteration around the spiral.

In the spiral model of development, the project team decides how exactly to structure the project ~~into~~ ^{into} phases. Normally projects start by using some generic model and then adding extra phases. Whenever some spiral risks are identified later during the project. The distinct feature of this model is its ability to handle risks. Thus it is realistic approach to the development of large scale systems and softwares.

Advantages of Spiral Model

- 1) User is able to see & perceive ~~pro~~ the project.
- 2) Risk analyzer which resolves higher priority errors

3) Project is very much refined.

4) Reusability of the software.

Disadvantages

1) It is only suitable for large size projects.

2) Model is complex to use.

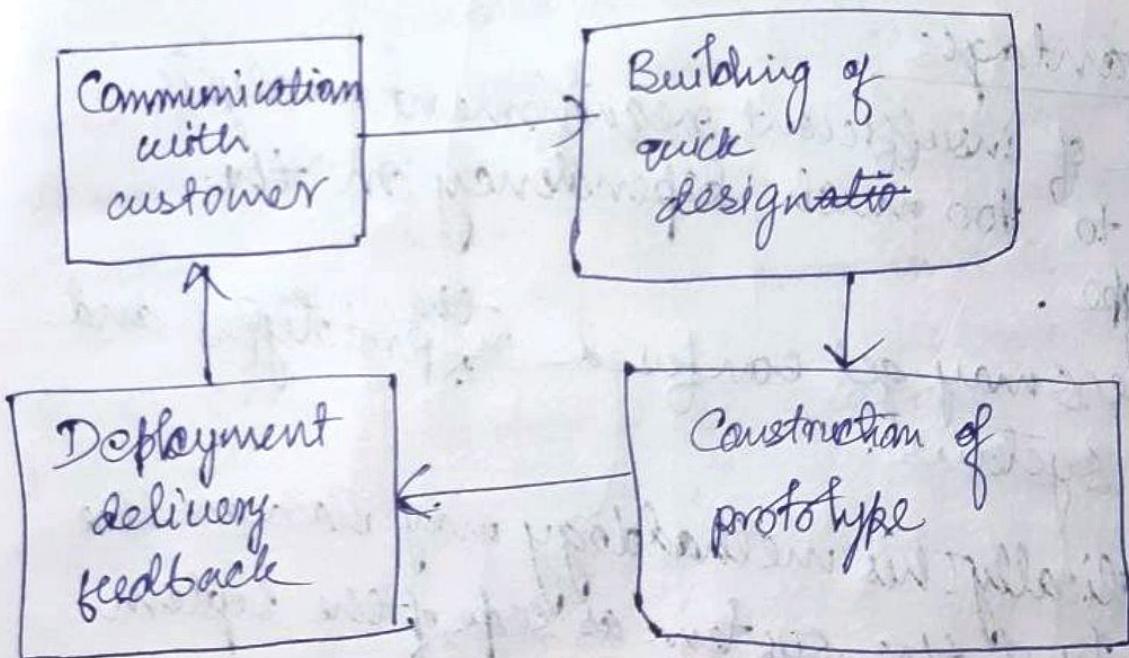
3) Management skill is necessary so as to analyze the risk factors.

Q Difference between Waterfall Model & Iterative Model

Prototyping Model

In prototyping model initially the requirement gathering is done. Developer and customer define overall objectives, identify areas needing more requirement gathering. Then a quick design is prepared. This design represents what will be visible to user in input and output format. From the quick design a prototype is prepared. Customer or user evaluates the prototype in order to refine the requirements. Iteratively prototype is tuned for satisfying.

customer requirements. Thus prototype is important to identify the software requirements. Thus when working prototype is build developers use existing programs fragments or program generator to throw away the prototype and rebuild the system to high quality. Certain classes of mathematical algorithms, subset of command driven systems and other applications where results can be easily examined without real time interaction can be developed using prototyping model.



Advantage

- 1) All type of Software Development can be done using this model.
- 2) This model provides accurate picture of the current state of the project.

- 3) Increased user involvement in the product even before its implementation.
- 4) Since a working model of the system is displayed, the user get a better understanding of the system being developed.
- 5) Reduces time and cost as the defects can be detected much earlier.
- 6) Quicker user feedback is available to better solutions.

Disadvantages

- 1) Risk of insufficient requirement analysis owing to too much dependency on the prototype.
- 2) Users may get confused in the prototypes and actual systems.
- 3) Practically this methodology may increase the complexity of the system as scope of the system may expand beyond original plans.
- 4) The effort invested in building prototypes maybe too much if it is not monitored properly.

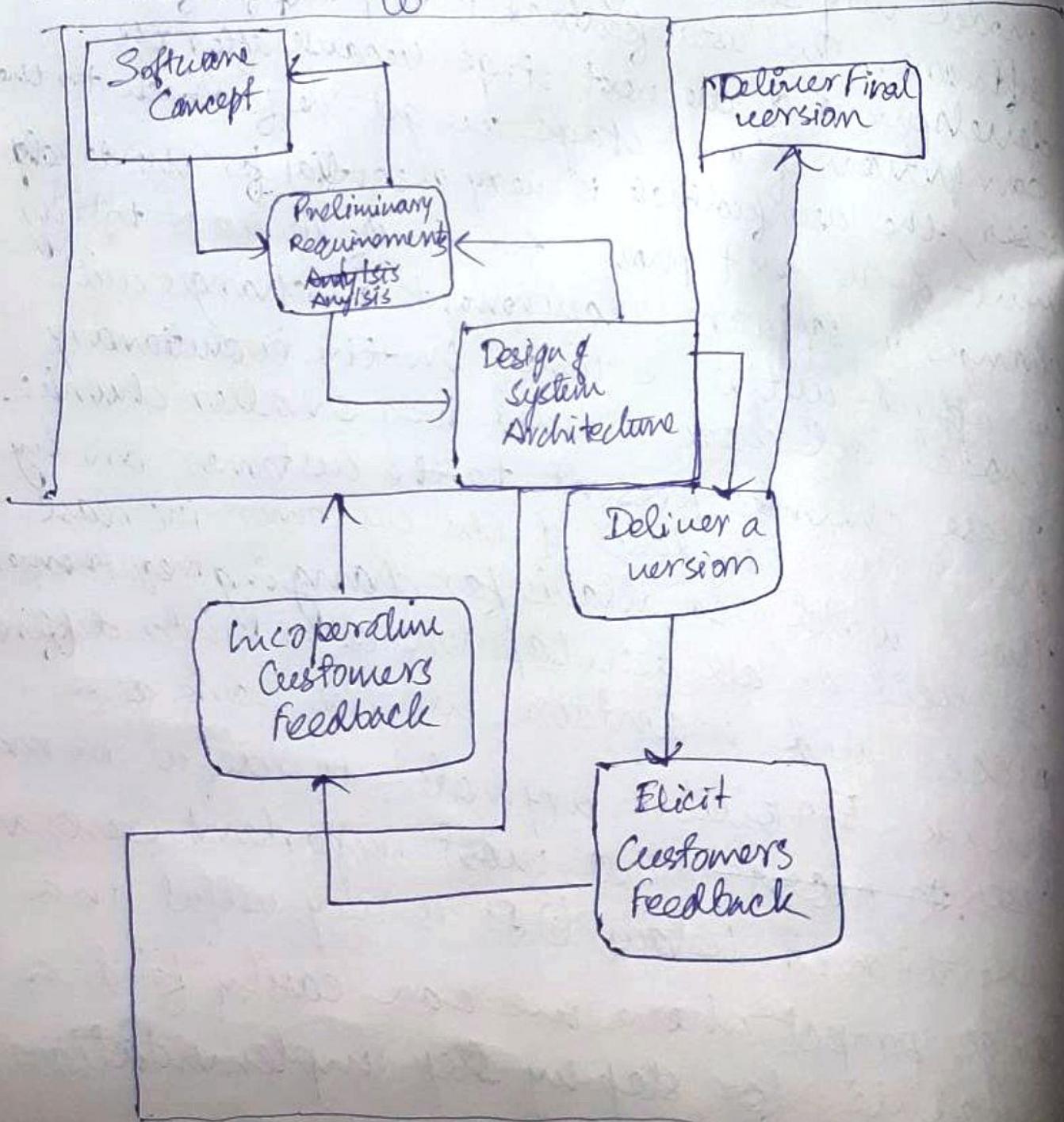
22/2/23

Evolutionary Model

The evolutionary model is the model of Software Engineering where all the work is done during the development phase. In this model all work divided into some small chunk or module. It is very suitable model because of user feedback and other factors that make the model very suitable for the development of complete software. The user feedback is very helpful for the development of the next stage because after the completion of one stage we get the feedback to the user, the user feedback is very essential for the development of the next phase. ~~We i~~ If we want to apply changes in software requirements all changes will be applied within a time. In the evolutionary model all work divided into smaller chunks. These chunks present to the customer one by one. The confidence of the customer increase. This model also allows for changing requirements as well as all development done into different pieces and maintains all the work as a chunk. Evolutionary model is useful in many cases. ~~To most~~ Two most important cases are mentioned below: (1) It is very useful in a large project where we can easily find a module for step by step implementation.

This model is used when the user need
→ to start using the many features instead of
waiting for the complete software.

(ii) This model is also very useful in object oriented
software development because all the development
is divided into different units



Advantages :

- 1) The big advantage of this model is that the user has to check every stage during the development and it is helpful in achieving customer confidence.
- 2) There are fewer chances of efforts errors because all the modules are well synchronized.
- 3) It helps to reduce the risk of software projects.
- 4) It also reduces the cost of development.
- 5) Minimizes serious problems during testing.

Disadvantages :

- 1) The delivery of full software can be late due to different changes by customers during development.
- 2) It is difficult to divide the problem into several parts that would be acceptable to the customer which can be incrementally implemented and delivered.

order

3/3/23

Software Project

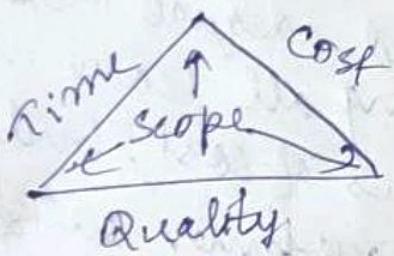
The job pattern of an IT company engaged in Software Development can be seen split in two parts:

- (i) Software creation
- (ii) Software project management

A project is a well defined task which is a collection of several operations done in order to achieve a goal. A project can be characterized as - Every project may have a unique & distinct goal. Project is not routine activity or day today operations. Project starts with a start time & end time. Project ends when its goal is achieved since it is a temporary phase in the lifetime of an organization. Project needs adequate resources in terms of time, manpower, finance, material, knowledge bank. A software project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to execution methodology in a specified period of time to achieve intended software product.

Need of Software Project Management

Software is said to be an intangible product. Software development is a kind of all new world business and there is very little experience in building soft software products. Most software products are ~~taylor~~ and tailor made and fit client's requirement. The most important is that the underline technology changes and advances show frequently and rapidly, that experience of one product may not be applied to the other one. All such business and environmental constraints bring the risks in software development hence it is essential to manage software projects efficiently.



The image above shows that the triple constraints for software projects. It is an essential part of software organisation to deliver quality product keeping the cost within client's budget constraint and deliver the project as per schedule. There are several factors both internal and external which may impact these triple constraint triangles. Any of three factors can severely impact the other two.

Software Project Manager

- ★ Software project manager is a person who undertakes the responsibility of executing the software projects. Software project manager is thoroughly aware of all the phases of SDLC that the software would go through.
- Project manager may never directly involve in producing the end product but he controls and manages the activity involved in production.
- ★ A project manager closely monitors and coordinates process, prepares and executes various plans, arranges necessary and adequate resources, maintains communication among all team members in order to address issues of cost, budget, resources, time, quality and customer satisfaction.