Page No.:
Date:

Unit-1:

Q-1) Explain changing nature of software evolution / Classify & Explain Different type of software with ex

Ans:
1. System software
2. Real time software
3. Embedded software
4. Engineering / Scientific software
5. Business software
6. Personal Computer software
7. AI software
8. web based software
9. Engineering / scientific software

1. System Software: Software comes under this category like compilers, operating systems, editors, drivers, etc. basically system software is collection of program to provide service to other programs.

It is collection of program written to service other programs. Some system software (eg. compilers, editors, and file Management.

2. Real time Software → used to monitor, control & analyze real world events as they occur.

eg: Software required for weather forecasting. Such process will gather & process the status of temperature, humidity & other environmental parameters.

Page No. :
Date :

(3) Embedded Software - This type of software is placed in "Read-only-Memory (ROM)" of the product & control the various functions of product. The product could be an aircraft, automobile, security system.

(4) Engineering / Scientific software → This software category has wide range of functions from astronomy to volcanology, from automotive stress analysis to space shuttle orbital dynamics & from molecular biology to automated manufacturing.

(5) Business Software - Largest application area. The software designed to process business application is called business software. Business software could be payroll, file monitoring system, employee management.

(6) Personal Computer Software - The software used in personal computers are covered in this category. Ex: word processors, computer graphics, multimedia, & animating tools, database management, etc.

(7) AI Software - AI makes use of non-numerical algorithm to solve complex problems that are not amenable to computation.

Q-2) Define Process framework of SE & Explain with next & clear Diagram

Ans: A process is collection of activities, actions, and tasks that are performed when some work product is to be created.

A Process framework establishes the foundation for a complete software engineering process by identifying a small number of framework activities that are applicable to all software projects, regardless of their size or complexity. In addition, the process framework encompasses a set of umbrella activities that are applicable accross the entitere software process.

A generic process framework for software engg ~~wp~~ encompasses five Activities

① Communication ② Planning ③ Modeling
④ Construction ⑤ Deployment

① Communication

Before any Technical work can commence, it is critically important to communicate with customer and other stakeholders. The intention is to understand stakeholders objective for the project and to gather requirement that helps to define software features & functions.

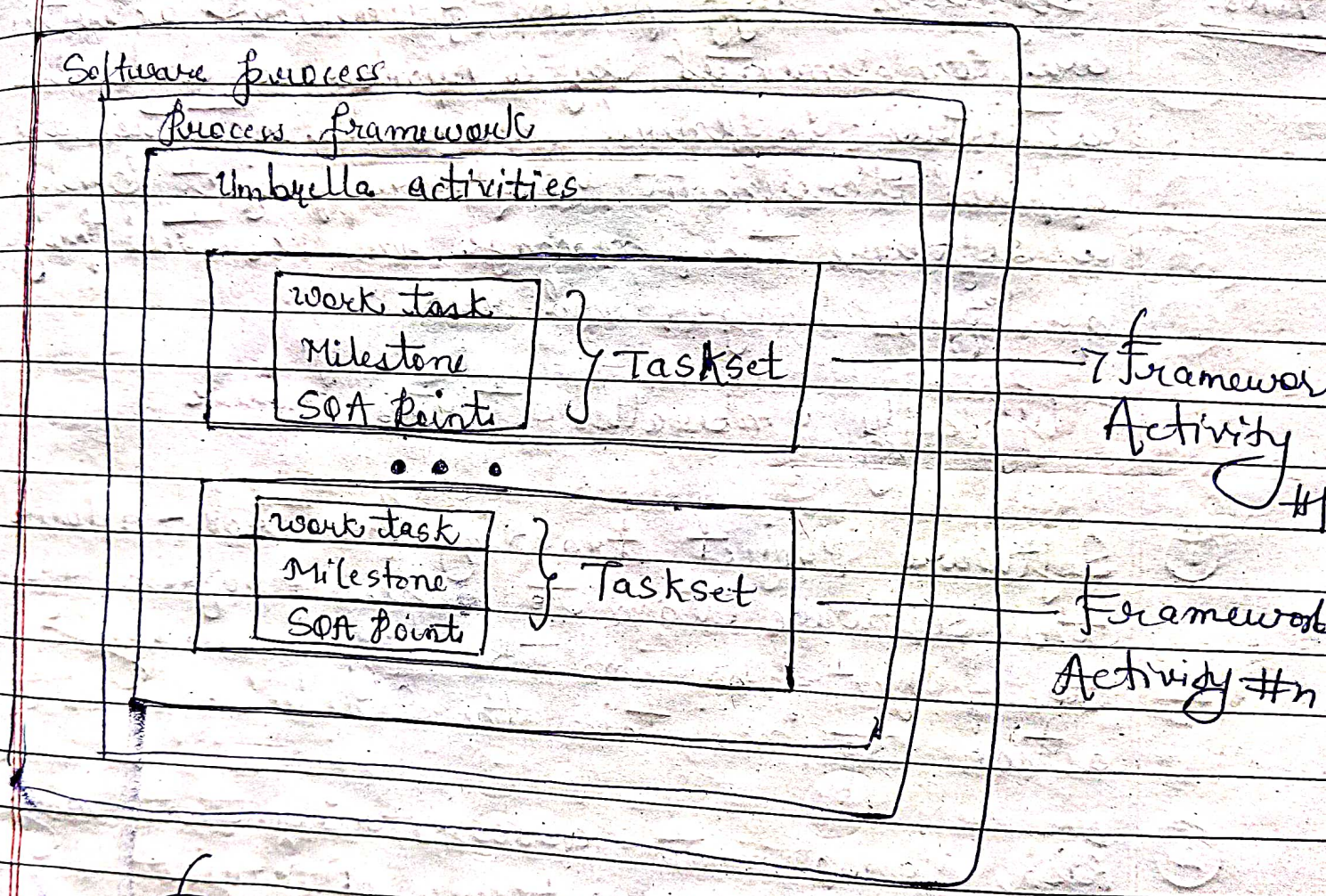
Page No. :
Date :

(2) Planning : Any complicated Journey can be simplified if map exists. A software project is a complicated Journey, and the planning activities creates a "map" that helps guide the team as it makes the Journey. The map called the software project plan - defines software engg. work by describing the technical task to be conducted, the risks that are likely, the resources that will be required, the work product to be ~~product~~ produced.

(3) Modeling : Whether you're landscaper, a bridge builder, an aeronautical engineer, carpenter or an architect, you work with models every day. You create a "sketch" of thing so that you'll understand the big picture. If required, you refine the sketch into greater and greater detail in an effort to better understand the problem and how you're going to solve it. A software engg does the same thing by creating models to better understand software requirements and the design that will achieve those requirements.

(4) Construction : This activity combines code generation and the testing that is required to uncover errors in code.

(5) Deployment : The software is delivered to customer who evaluates the delivered product to provide feedback based evaluation



For many software projects, framework activities are applied iteratively, as a project progresses. That is, communication, planning, modelling, construction & deployment are applied repeatedly through a number of project iterations. Each project iteration produces a software increment that provides stakeholders with a subset of overall software features.

Page No.:
Date:

Que-3) List & explain Umbrella activities needed for Software development

Ans: Software engg process framework activities are complimented by a number of umbrella activities. In general, umbrella activities are applied throughout a software project & helps a software team manage and control progress, quality, change, and risk.

Typically Umbrella activities Include

1. Software Project tracking & Control - allows the software team to assess progress against the project plan and take any necessary action to maintain the schedule.

2. Risk Management - assesses risks that may affect the outcome of project

3. Software Quality assurance - defines & conducts the activities required to ensure software quality.

4. Technical reviews - assesses software engg work products in an effort to uncover and remove errors before they are propagated to next activity.

Page No. :
Date :

(5) Measurment - defines & collect process, project & product measures that assist the team in delivering software that meets stakeholders' needs can be used in conjunction with all other framework and umbrella activities.

(6) Software Configuration Management - Manages the effects of change throughout the software process.

(7) Reusability Management - defines criteria for work product reuse & establishes mechanisms to achieve reusable components.

(8) Work product Preparation & Production - encompasses the activities required to create work products such as models, document, logs, forms & list -

Page No. :
Date :

Ques-4) Mention characteristics of software contra-sting it with Characteristic of Hardware

Ans (1) Software does not wear out
(2) software is not manufacture
(3) Most software is custom built rather than being assembled for Components.
(4) Usability of software.
(5) Reusability of Components.
(6) Flexibility of software.
(7) Maintainability of software.
(8) Portability of software.
(9) Reliability of software.

(1) Software does not wear out

(1) In early stage of Hardware development process the failure rate is very high because of manufacturing defects. But after correcting such defects the failure rate gets reduced.
(2) The failure rate remains constant for some period of time and again it starts increasing because of environmental maladies.
(3) On the other hand software does not get affected from such environmental maladies. Hence ideally it should have an "idealized curve".
But due to some undiscovered errors the failure rate is high and drops down ASA as the errors get corrected

(4) Hence in failure rating of software the "actual curve" is shown below

(5) During the life of the software if any change is made, some defects may be introduced

(6) This causes failure rate to be high. Before the curve can return to original steady state another change is requested and again the failure rate becomes high

(7) Another issue with software is that there are no spare part for software

(8) If hardware component wears out it can be replaced by another component but it is not possible in case of software.

(9) Therefore, software maintainance is more difficult than Hardware Maintainance

(2) <u>Software is not Manufacture</u>

(1) Software & Hardware development are two different activities

(2) A good design is a backbone for both activity

(3) Quality problem that occur in Hardware manufacturing phase can not be removed easily. On other Hand, during software development process such problems can be rectified.

(4) In both the activities, developers are responsible for producing qualitive product.

Page No.:
Date:

(3) Most software is custom built rather than being assembled from components

(1) while developing any hardware product firstly the circuit design with desired functioning properties is created.

(2) Then required hardware components such as IC's, capacitors & registers are assembled according to the design, but this is not done while developing software product

(3) Not of the software is custom built.

(4) However now the software development approach is getting changed and we look for reusability of software components. It is practised to reuse algorithm & data structures

(5) Today software Industry is trying to make library of reusable components.

For ex.

(4) Usability of Softwares

Usability
- → Understandability
- → Learnability
- → Operability

It refers to extent which software can be used with ease. the amount of effort or time required to learn how to use the software

(5) Reusability of Components

As software never wears out, neither does it component i.e code segments. So if any particular segment of code is required in some other software, we can reuse the existing code form the software in which it is already present. This reduce work & save time & money

(6) Flexibility of Software.

A software is flexible i.e we can make any necessary changes in our software in future according to need of that time and then can use the same software then also

(7) Maintainability of Software.

Every software is maintainable i.e if there are any bugs or errors appears in software, then they can fixed.

(8) Portability of Software

we can transfer our software from one place to another platform. Due to this sharing of software among the developers and other members can be done flexibly.

Software can be relaunch in any other platform without any changes.

Software Must be made Platform Independent

Page No. :
Date :

(9) Reliability of Software

(1) Ability of software to provide the desired functionalities under every condition. This means our software must work proper in every condn

Ques-5) Write any 2 Software engg challenges?

(1) Delivery Time Challenge - There is increasing pressure for faster delivery of software. When the complexity of system that we develop increases, this challenge becomes harder.

(2) Heterogeneity Challenge - Sometime system are distributed and include a mix of Hardware & software. This implies that software system must cleanly integrate with other different software system, built by different organization & team which may be using different software & Hardware platform

Page No. :
Date :

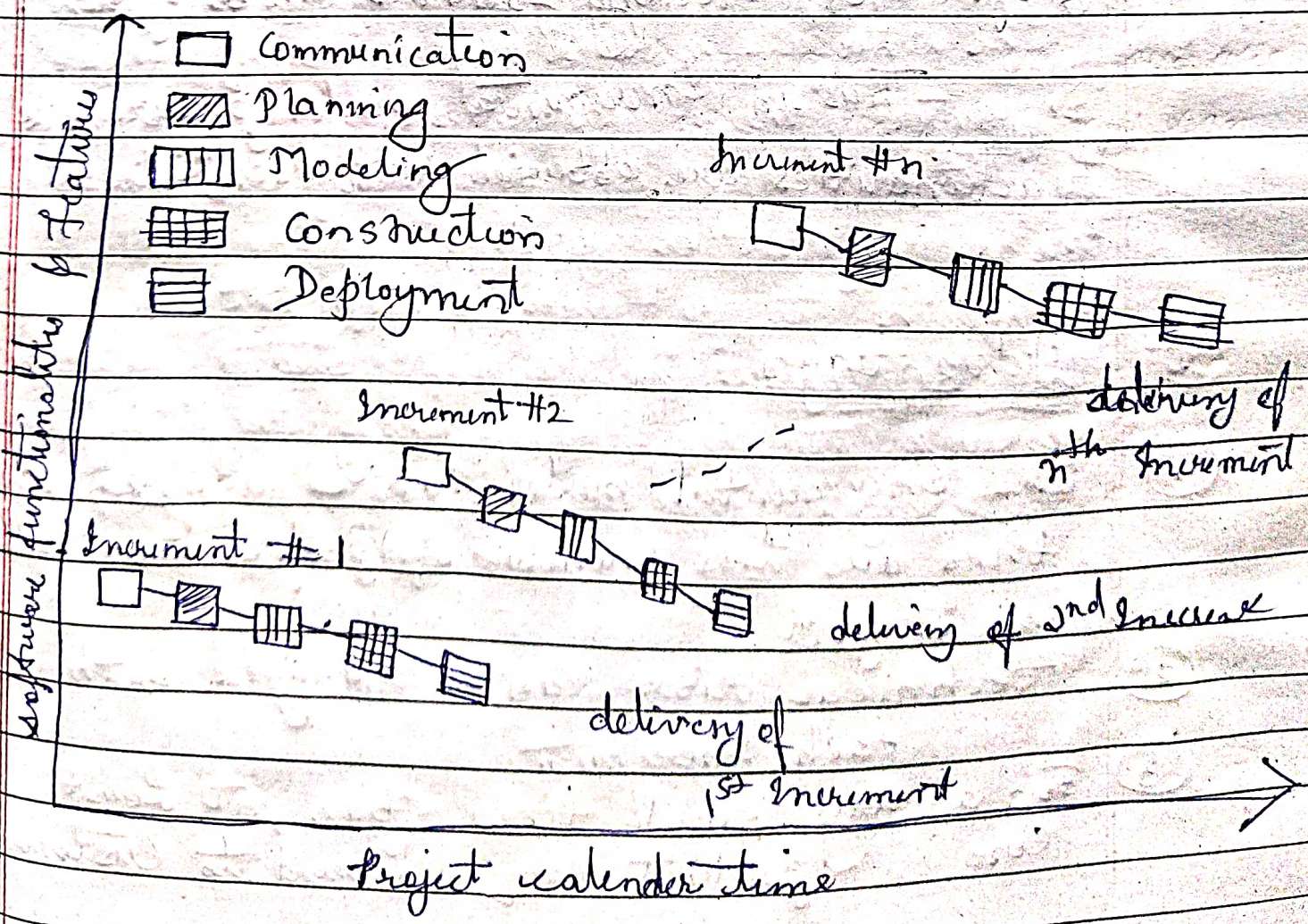
Q-7) Explain Linear Sequential model with AD & DA

Q-8) Explain Incremental model with its Advantages & Disadvantages

Ans: The Incremental model has some phases that are in waterfall model But it is iterative in nature. The Incremental model has following phases

(1) Analysis (2) Design (3) Code (4) Test

The Increment model delivers series of releases to the Customer. This releases are called Increment. More & More functionally is associated with each Increment



Page No. :
Date :

(3) The first Increment is called Core Product. In this release the basic requirement are implement and then in subsequent increments new requirments are added.

(4) The word Processing software package can be considered as an example of I.M. In the first Increment only the document processing facilities are available. In the second increment, more sophisticated document producing & producing facilities, file Mngt functionalities are given.

In the next Increment Spelling & grammer checking facilities can be given.

Thus in Incremental model progressive functionalities are obtained with each release.

Advantages

(1) The Incremental model can be adopted when there are less number of people involved in Project

(2) Technical Risks can be Managed with each Increment

(3) For a very small time span, at least core product can delivered to customer

Page No. :
Date :

Disadvantages

1. Complexity - Managing multiple Increments and their interdependencies can add Complexity to the Project management & Development process

2. Not Suitable for All Projects - The Incremental Model is not suitable for all types of projects, especially those that require a well defined, one-time implementation

3. Integration challenges :- Ensuring smooth Integration of different increments can be challenging.

Ques - 7) Explain Linear Sequential Model with its Advantages & Disadvantages.

Page No. :
Date :

OR

Ques - 8) A software project which is considered to be very simple and the ~~Cus~~ customer is in position of giving all the requirments at the initial stage, which process model would you prefer for developing project?

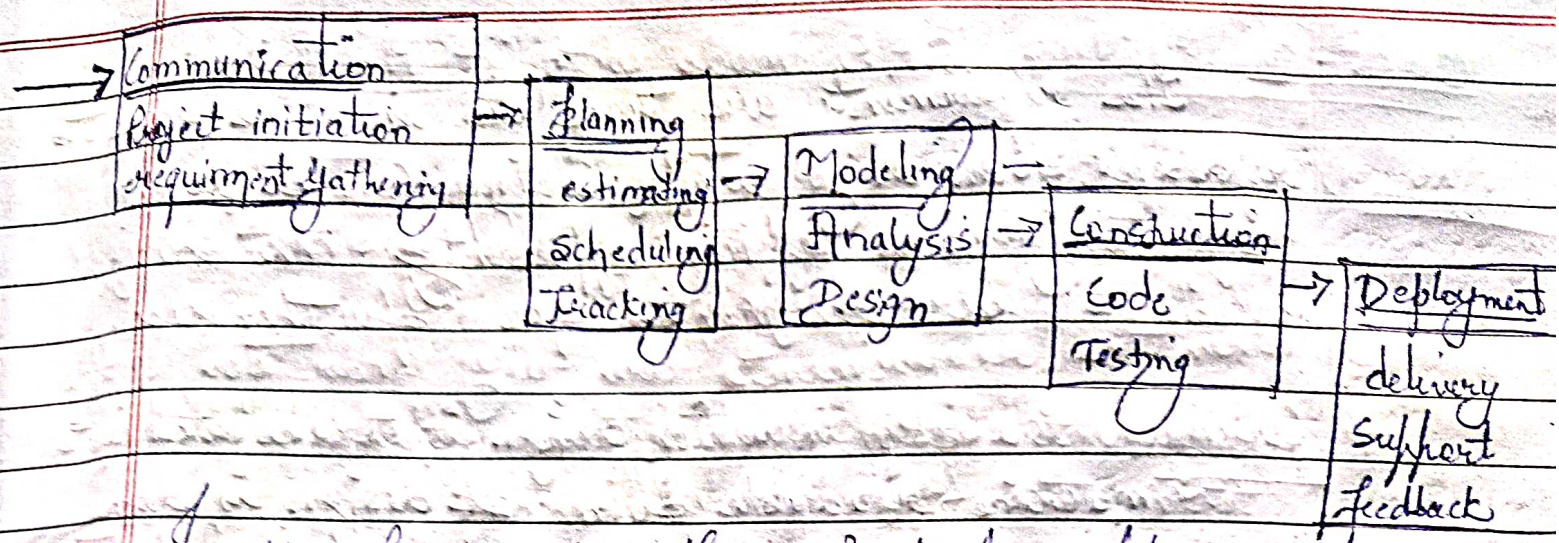
Ans:

For a software project that is considered very simple, where the customer can provide all the requirments at the initial stage a traditional and linear process model like "Waterfall" model would be a suitable choice.

* Waterfall Model / Linear Sequential Model

(1) The waterfall model is also called as "Linear - Sequential model" or "classic life cycle model" It is the oldest software paradigm. The model suggest a systematic, sequential approach to software development.

(2) The software development starts with requirments gathering phase. Then progresses through analysis, design, coding, testing and maintainance.



* In the Requirment Gathering & Analysis phase the basic requirment of system must be understood by the Software Engineer, which is also called Analyst. The information domain, function, behavioural requirments of the system are understood. All these requirments are then well documented & discussed further with the customer for reviewing.

- The Map called software project plan - defines the software engineering work by describing the technical tasks to be conducted, the risks that are likely the resources that will be required, the work products to be produced, and a work schedule.

The Design is an Intermediate step between the requirment analysis and coding. Design focuses on program attributes such as :-

1. Data Structure
2. Software architecture.
3. Interface representation.
4. Algorithmic details.

Date :

The requirments are translated in some easy to represent form using which coding can be done effectively and ~~effect~~ efficiently. The design needs to be documented for further use

* **Coding** : is a step in which design is translated into machine-readable form. If design is done in sufficient detail then coding can be done ~~efficient~~ effectively. Programs are created in this phase

* **Testing** : Testing begins when coding is done while performing testing the major focus is on logical internals of the software. The testing ensure execution of all path, functional behaviour. The purpose of testing is to uncover errors, fix the bugs and meet the customer requirments.

* **Maintenance** : Is the longest life cycle phase. When system is installed & put in practical use then error may get introduced, correcting such errors and putting it in use is the major purpose of maintenance activity.

Page No. :
Date :

* Advantages
(1) The waterfall model is simple to Implement
(2) For Implementation of small systems waterfall model is useful.

* Drawbacks / Disadvantage :

(1) It is difficult to follow the sequential flow in Software development process. If some changes are made at some phases then it may cause some confusion.
(2) The Requirment analysis is done initially and sometimes it is not possible to state all the requirments explicitly in the beginning. This causes difficulty in project.
(3) It takes large time to complete the Project.
(4) Customer can see the working model of Project only at the end.

Ques- 9) Explain RAD mode with it Adv & Dis

Ans: 1) The RAD model is a type of Incremental process model in which there is extremely short development cycle.

(2) When the requirements are fully understood & the component based construction approach is adopted then the RAD model is used.

(3) Using the RAD model the fully functional system can be developed within 60 to 90 days.

(4) Various phases in RAD are Requirements Gathering, Analysis & planning, Design, Build/construction and finally Deployment.

(5) Multiple teams work on developing the software system using RAD model parallely.

(6) In the Requirement gathering phase the Developers communicate with the user of system & understand the business process and requirement of software system.

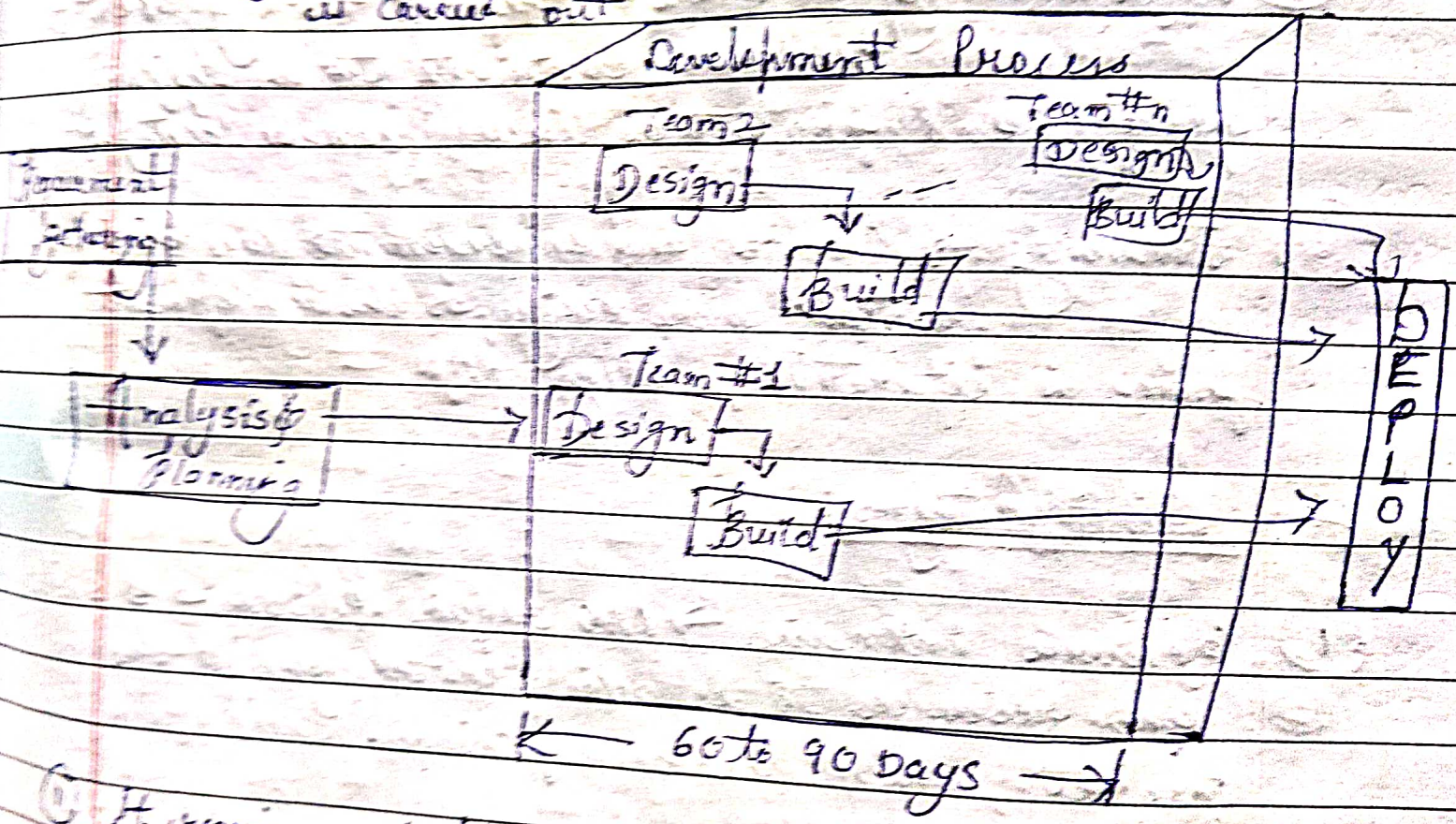
(7) During analysis and planning phase, the analysis on the gathered requirements is made and a planning for various software development activities is done.

(8) During the Design phase various models are created. These models are Business model, data model & process model.

(9) The Build is an activity in which using the existing software components and automatic code generation tool the implementation

Code is created for the software system. This code is well tested by its team. The functionalities developed by all the teams are integrated to form a whole.

(6) Finally the Deployment of all the software components is carried out



(1) It requires multiple team or large number of people
(2) It works on the scalable.
(3) This model require heavily committed developers & customer.
(4) The projects using RAD model require heavy resources

Advantages:

(1) Speed & Efficiency :- RAD focuses on rapid development and prototyping, which results in faster project delivery.

(2) High-Quality Software - Frequent testing & client feedback contribute to a higher quality end product as issues can be identified

(3) Flexibility → Changes and enhancements can be incorporated more easily as the development process is iterative, making RAD

Disadvantages:

(1) Resource Intensive → RAD require personal & skilled and resources to be available throughout the project, which can be expensive.

(2) Dependency on client Availability → Active client involvement is crucial and if the client cannot commit the necessary time & resources

(3) Limited Applicability → The RAD model is not suitable for all types of projects, especially those with well-defined requirements.

Page No. :
Date :

Ques → b) Explain Prototyping in Detail?

① In Prototyping model initially the requirment gathering is done.

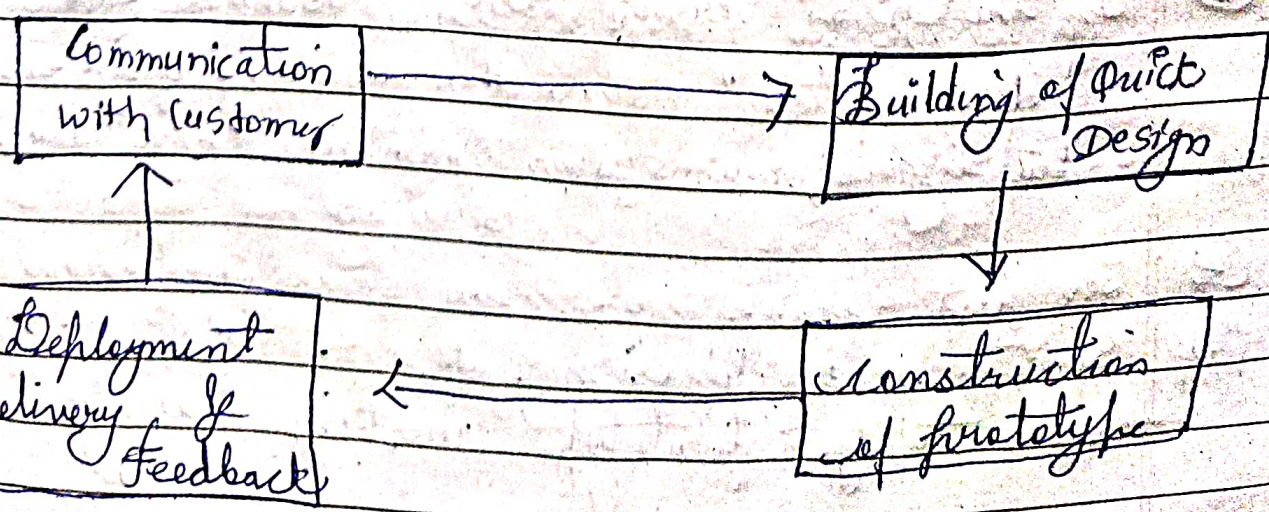
② Developer & customer defines overall objective, Identify areas needing more requirment gathering.

③ Then a Quick design is prepared. The design prefers represents what will be visible to user in input & output format.

④ From the Quick design a protype is prepared. Customer evaluate the prototype in order to refine the requirments. Iteratively prototype is tuned for satisfying customer requirments. Thus Prototype is Important to Identify the Software requirments.

⑤ When working prototype is built, developer use existing program fragments to throw away the prototype and rebuild the system to high Quality.

⑥ 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 paradigm.



Page No. :
Date :

When to Choose

(1) Software Applications that are relatively easy to prototype almost always involve Human-machine Interaction the prototyping model is suggested.

(2) General objective of Software is defined but not detailed input, processing or output requirement. Then in such a case prototyping model is useful

(3) when the developer is unsure of the efficiency of an algorithm or adaptability of OS then

* Advantages :

(1) Early problem Identification : Helps catch issues in design phase

(2) User Involvement : Allow users to provide feedback for better Product alignment with their needs

* Disadvantages :

(1) In the first version itself, customer often wants "few fixes" rather than rebuilding of system whereas rebuilding of new system maintains high level of Quantity

(2) The first version may have some compromises

(3) Sometimes developer may make Implementation compromise to get prototype working Quickly

Ques-11) Explain the Concept of Spiral Model with neat & clean Diagram

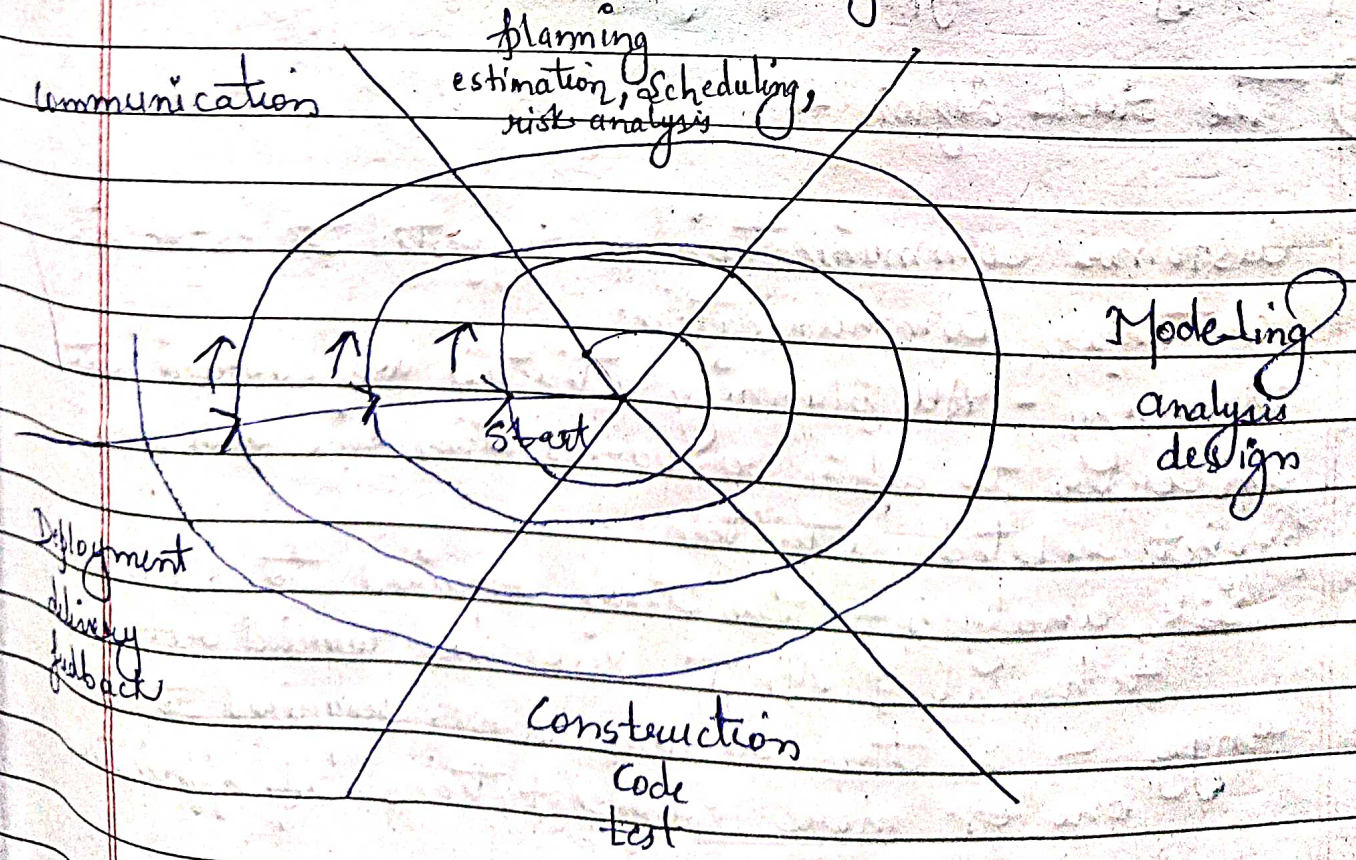
Ans: ① The Spiral model is an evolutionary software process model that couples the iterative nature of prototyping with the controlled & systematic aspects of waterfall model

② It provides the potential for rapid development of increasingly more complete version of software

③ This model posses the iterative nature of prototyping model & controlled and systematic approaches of linear sequential model

④ This model gives efficient development of incremental version of software. In this model, the software is developed in series of increments

⑤ The Spiral model is divided into a number of framework activities. These framework activities are denoted by task regions



(6) Spiral model is realistic approach to development of large-scale systems & software because software & developer better understand the problem statement at each evolutionary level. Also risks can be identified at each such level.

(7) In the initial pass, product specification is built and in subsequent passes around the spiral the prototype gets developed and then more improved versions of software gets developed.

(8) During planning phase, the cost & schedule of software can be planned and adjusted based on feedback obtained from customer evaluation

(9) In spiral model, project entry point axis is defined. This axis represents starting point for different types of projects

(10) The task Region –

(1) Customer Communication → Suggested to establish customer communication

(2) Planning – All planning activities are carried out in order to define resources time line and other project related activities

(3) Risk analysis – The tasks required to calculate technical & Mngt risks are carried out

(4) engg – In this region task required to build one or more representations of applications are carried out

(5) Construction & Release - All the necessary tasks required to construct, test, install the application are conducted. Some tasks that are required to provide user support are also carried out in the task region

(6) Customer Evaluation - Customer's feedback is obtained and based on customer evaluation required tasks are performed & Implement at Installation stage.

In each region, number of work tasks are carried out depending upon the characteristics of project. For a small project relatively small number of work tasks are adopted but for a complex project large number of work tasks can be carried out

In the spiral model, the software engg team moves around the spiral in clockwise direction beginning at the core.

Advantages of Spiral Model:

(1) Requirment changes can be made at every stage

(2) Risks can be identified and rectified before they get problematic

Page No. :
Date :

* Disadvantages of Spiral Model

(1) It is based on customer communication. If the communication is not proper then the software product that gets developed will not be up to mark.

(2) It demands considerable risk assessment. If the risk assesment is done properly then only the successful product can be obtained.