

CS-403 Software Engineering

Unit -1

The Software Product and Software Process

Software Product & Process characteristics,
D Software Process Models :- Linear Sequential Model, Prototyping Model, RAD Model, Evolutionary Process Model like Incremental Model, Spiral Model, Component Assembly Model, RUP and Agile process. Software Process Customization and improvement, CMMI, Product and Process metrics.

Software :- Software is nothing but collection of Computer programs and related documents that are planned to provide desired feature, functionalities and better performance.

Software, when made for a specific purpose is called Software product.

Software is considered to be collection of executable libraries and programming code, associated documentations.

Engineering :- Application of Science, tools & Methods & to find' cost effective solution to problems.

Software Engineering :- SE is a technique through which we can developed or created software for computer system and any other electronic devices.

- * In other words, Software engineering is a process in which user needs are analyzed and software is designed based on their needs.
 - * In SE the development of software using well defined scientific principle method and procedures.
 - * Software Engineers build these software and application by using designing and programming languages.
 - * In order to create a complex SW we should use SW engineering technique as well as to reduce the complexity we should use abstraction and decomposition where, abstraction describes only important part of the SW and leave irrelevant things for later stage of development so, the req. of SW become simple.
- # Characteristic of Software :- Engineering-
- ① Functionality:- It refers to the degree of performance of the Software against its purpose.
 - * Suitability
 - * Accuracy
 - * Interoperability
 - * Security
 - ② Reliability:- A set of attributes that bears on the capability of SW to maintain its level of performance under the given condition

for a stated period of time.

Required functions are:-

- * Recoverability
- * Fault Tolerance

(3) Efficiency:- It refers to the ability of the S/W to use system resources in the most effective & efficient manner.

The S/W should make effective use of storage space and execute commands as per desired timing requirements.

Required functions are:-

- * In time
- * In resource

(4) Usability:- It refers to the extent to which the S/W can be used with ease. The amount of effort or time required to learn how to use S/W.

- * Understandability
- * Learnability
- * Operability

(5) Maintainability:- It refers to the ease with which modifications can be made in a S/W system to extend its functionality, improve its performance, correct errors, Testability, Stability, Changeability, Operability.

⑥ Portability :- A set of attributes that enables the ability of SW to be transferred from one env. to another without or minimum changes.

- Adaptability
- Installability
- Replaceability

Types of Software:-

⑦ Generic :- Create for different range of people and sold to open source, any customer who wishes to buy them.

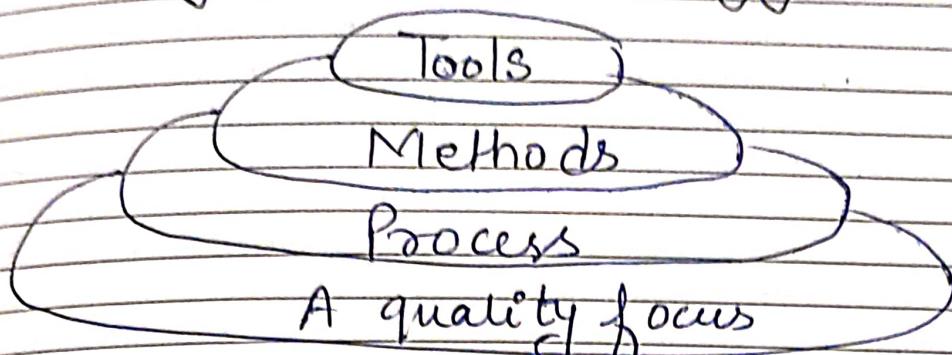
Developed to solution whose requirement are very common, fairly stable and well understood by Software Engineers.

Eg:-

⑧ Custom Software:- Developed for a single customer according to their specification.

Eg:-

A Layered Technology :-



Layered Architecture

Software engineering is totally a layered tech. That means to develop software we will have to go from one layer to another.

- ① Quality Focus :-
- Degree of goodness
 - Correctness
 - Maintainability
 - Usability
- Functional
- Non-functional (Structural)
- The degree to which correct software has been developed

- ② Process :- A framework that must be established for effective delivery of software
- "What" to do → timely development of software
 - Deals with activities, actions & tasks
 - Comes out with "How" to question.

③ Methods :-

- Deals with "How to" Implement
 - Communication
 - Requirement & design modeling Analysis.
 - Using prog. tools
 - Testing & Support
- ④ Tools :-
- Provide automated support for process & methods
 - Environment
 - Helping Hands of process
 - Automated Support
- Used for code, design, test or sell

- * Process → Management and control of SW projects. converge
- * Methods → Each method of multiple tools
eg:- requirement, Analysis, testing & support
- * Tools:- Provide automated support for process and methods

Factors in choosing a SW Process Model

- * Project Req. * Project Complexity
- * Project Size
- * Cost of delay :- Project highly time-bound with huge cost of delay, or make the timelines flexible
- * Customer involvement :- Participation of customer during the process.
- * Familiarity with technology :- developer's knowledge and experience with the project domain, SW tools, language, and methods needed for development.
- * Project Resources :- availability of funds, staff, and other resources.

Software Process Model:-

- ⇒ Software Process :- Collection of Activities, Actions, Task.
- * A framework
 - * An adaptable workflow
 - * A model
- ⇒ Activity :- Strive to achieve a broad objective

Action :- It encompasses a set of tasks that produces a major work product (e.g., an arch. design model)

Task :- It focuses on a small but well-defined objective that produces a tangible output.

⇒ S/w process can be defined as the structured set of activities that are req. to develop the S/w system.

* to solve actual problems in an industry setting, of software eng. or a team of Eng. must incorporate a development strategy that encompasses the process, method and tools layers.

This strategy is often referred to as a process model or a S/w eng. paradigm

A process model for s/w Eng. is chosen based on the nature of the project and application, the methods and tools to be used and the controls and deliverables that are required.

- * Goal of slur process Model:-
A process model defines the following
 - A set of tasks that need to be performed.
 - The I/P to and O/P for each task.
 - The pre-condition & post condition for each task.
 - The sequence & flow of these tasks.

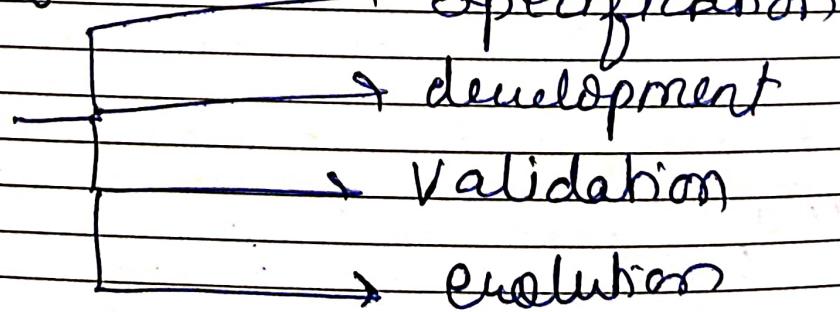
- # * Characteristics of Slur Process:-
- * Slur cost are concentrated in analysis & design and not in production.
- * Understandability :- To what extent is the process explicitly defined & how easy is it to understand the process (lef).
- * Visibility :- Do the process activities automate in clear result so that the progress of the process is externally visible.

Supportability

- * Robustness
- * Maintainability

- * Acceptability :-
- * Reliability :-

Process Activities



- * **S/w Specification:-** In this activity Eng. defines the S/w customer & S/w of & the constraints to be produced & functions.
- * **S/w Development:-** Here the S/w Eng. design programmed the S/w as per customer specification.
- * **S/w Validation:-** In this activity S/w is checked to ensure that it is what the customer require as in the specification
- * **Software Evolution:-** In this activity S/w is modified to reflect changing customer & market promote req.

Software Product:-

A software product, user interface must be carefully designed and implemented because developers of that product and users of that product are totally different. In case of a program, very little documentation is expected, but a software product must be well documented. A program can be developed according to the programmer's individual style of development, but a S/w product must be developed using accepted S/w engineering principles.

Characteristic :-

- * Correctness :- the S/w which we are making should meet all the specifications stated by the customer.
- * Usability / Learn - ability :- The amount of efforts or time required to learn how to use the S/w should be less. This makes the S/w user-friendly even for IT-illiterate people.
- * Integrity :- Just like med. It may affect the working of another application. But good quality S/w should not have side effects.
- * Reliability :- The S/w product should not have any defects. Not only this, it shouldn't fail during execution.
- * Efficiency :-
- * Security :-
- * Safety :-

Difference b/w Software process and Software product :-

Software Process	Software Product
* process are developed by individual user and it is used for personal use.	* It is developed by multiple user & if it is used by large no. of people or customers.
* Process may be small in size & possessing limited functionality.	* It consist of multip - i.e program codes, relate documents

such as SRS,
designing documents,
user manuals, test
cases

- * Process is generally developed by process engineers.
- * Process is generally developed by process engineers. Therefore systematic approach of developing SW product must be applied.
- * Software product relies on SW process for its stability quality and control only one person uses the process, hence look of user interface.
- * It is important than SW product multiuser no lack of user interface.

Software Development Life cycle:-

- * Software Development life cycle is a process used by S/w industry to design, develop and test high quality S/w.
- * The SDLC aims to produce high-quality S/w that meets or exceeds customer expectations, reaches completion within time & cost estimates.
- * SDLC is a process framework defining tasks performed at each step on the step-by-step development process.

Gathering Requirement Analysis

↓
Defining / Requirement Analysis Review

↓
Design

↓
Building | development | Implement

↓
Testing

↓
Deployment | maintained

↓
mat

* Requirement :- Business analyst collects the requirement from the customer / client as per the clients business needs and document the requirement, and provide the same to developer in the business Reg. Specification ^{Top down} _{Zeal}.

* Analysis :- Once the reg. gathering is done the next step is to define product to be developed and document the product req. and get them apporved from the customer. This is done through S/w req. specification documents (SRS).

key people involved in this phase are Project Manager, Business Analyst and Senior members in the team.

The outcome of this phase is SRS.

* Design :- HLD :- It gives the arch. of the S/w product to be developed and is done by architects and senior developers.

LLD :- It is done by senior developer. It describes how each and every feature in the product should work.

* Development :- This is the phase where we start building the S/w and start writing the code for the product. The outcome from this phase is source code document and developed product.

* Testing :- When the S/w is ready, it is sent to the Test team. Once the testing is done, The outcome of this phase is Quality product & the testing artifacts.

* Deployment:- Deployment is done by the deployment/ implementation engg. Once when the customer start using the developed system then the actual problems will come up and need to be solved from time to time. Maintenance should be done as per service level agreement.

Linear Sequential Model :-

Waterfall Model / classic life cycle model

* earliest SDLC approach

* Sequential model

* In this model each phase must be completed before the next phase can begin

Feasibility Study

↓
Req. analysis & define

When to use :-

Design

- * Req. are clear & fixed & may not change
- * no ambiguous req.
- * when the tech. is well understood.
- * project is short & cost is low.

Coding & unit Testing

Integration & System Testing

Deployment

* Feasibility Study:- In a SW company every customer come with their problem and we analyze that we are feasible or not to solve customer problem. We estimate the rough idea of the resource req. as well as estimate time for completion, economical, legal, techniq. feasible.

* Req. analysis:- The req. gathering process is focused specially on SW. To understand the nature of program to be built, SW eng. must understand the info. domain for the SW, as well as req. fun.

Req. for both the system & the user are documented & reviewed with the customer.

* Design:- The goal of design phase is to transform the req. specified in the SRS documents into a structure that is suitable for implementation in some prog. lang.

We sketched a blueprint of the project on a paper or on a computer.

The various phases can be involved in this phase:-

→ Algo, flowchart, Pseudocode, DFD, ERD.

* Coding & Unit Testing:- Developers start build the entire system by writing code using the chosen prog lang. The system is first developed in small program called units.

testing is performed on
in this phase each unit of code is
tested

* Integration & System testing - Once
the code has been generated, program
System testing is began.

* Main Deployment :- { * Delivery
* Support
* Maintenance

→ Advantage :-

- * Simple & easy to understand
- * Easy to manage
- * works well for smaller & low budget projects where req. are very well understood
- * Easy to arrange tasks

⇒ Disadvantages :-

- * No working software is produced until late during the life cycle.
- * High amount of risk & uncertainty
- * Not a good model for complex projects
- * Poor model for long & ongoing projects
- * It is difficult to measure progress within stages.

Prototyping Model:-

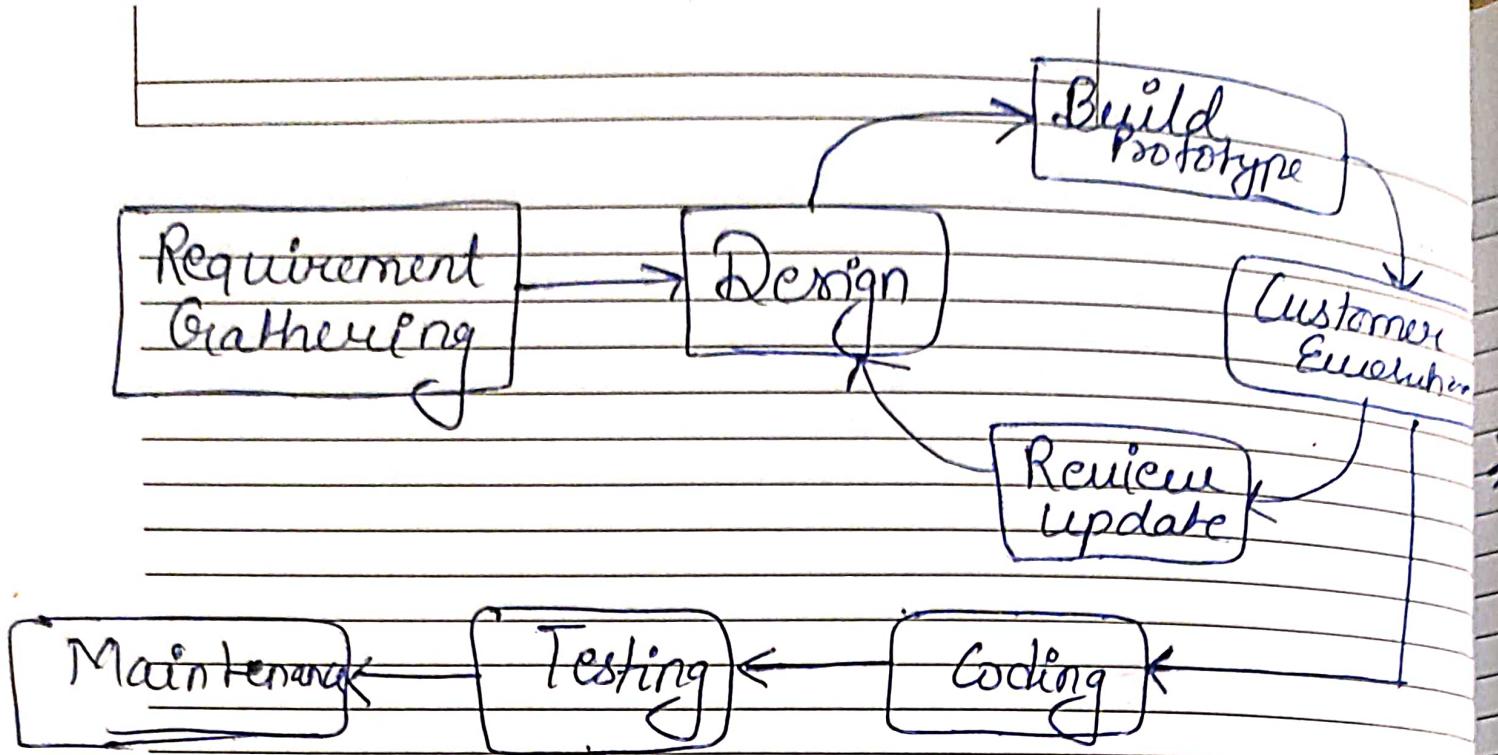
- * Prototype is a working model of S/W with some limited functionality.
- * In prototype Model, we do not develop the full S/W, firstly we develop a prototype of the S/W which means we include some most favourable functions in the product.
- * After this, the S/W is handed over to the customer and the customer will use the prototype product and check any possible shortcomings.
- * If any problem is in the product or customer is not satisfied of the product then we removed problems in the product or we create a new version of prototype of the product.

⇒ When to use Prototype model:-

- * Prototype model should be used when the desired system needs to have a lot of interaction with the end user.
- * Unclear req:- when req. are not properly understood.
- * Complicated and Large systems

⇒ Advantages:-

- * User are actively involved in the development.
- * More accurate user req. are obtained.



- * Error can be detected much earlier.
- * Quicker customer feedback provides a better idea of customer needs.
- * Missing functionality can be identified easily.
- * If customer not satisfied with prototype than we can develop a new prototype.

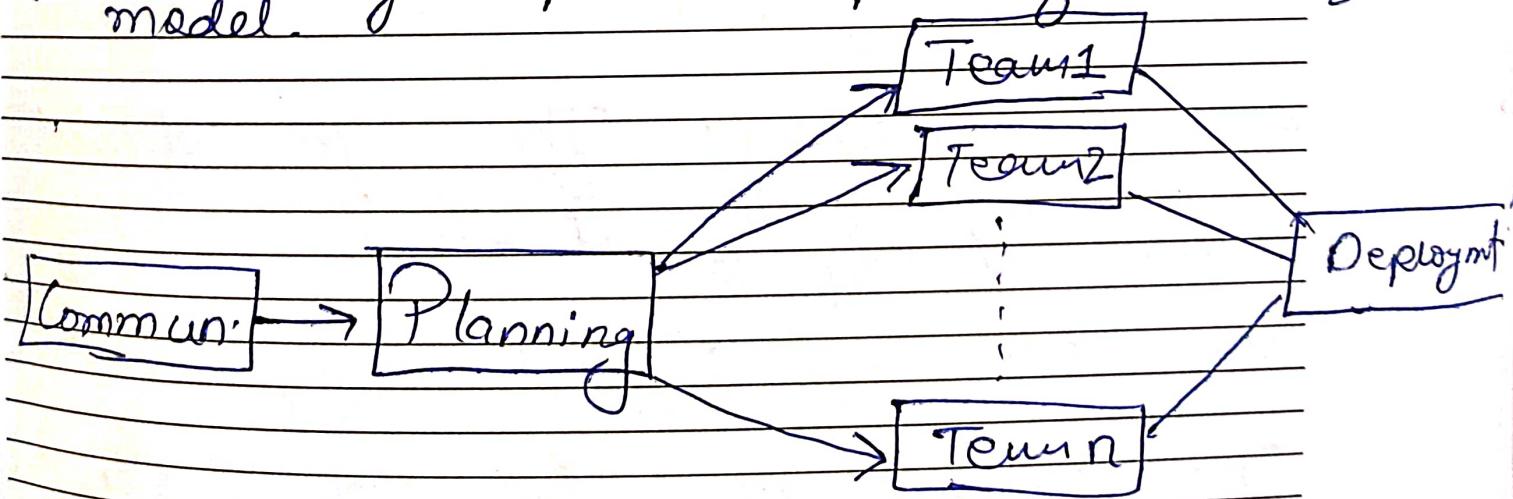
Disadvantages:-

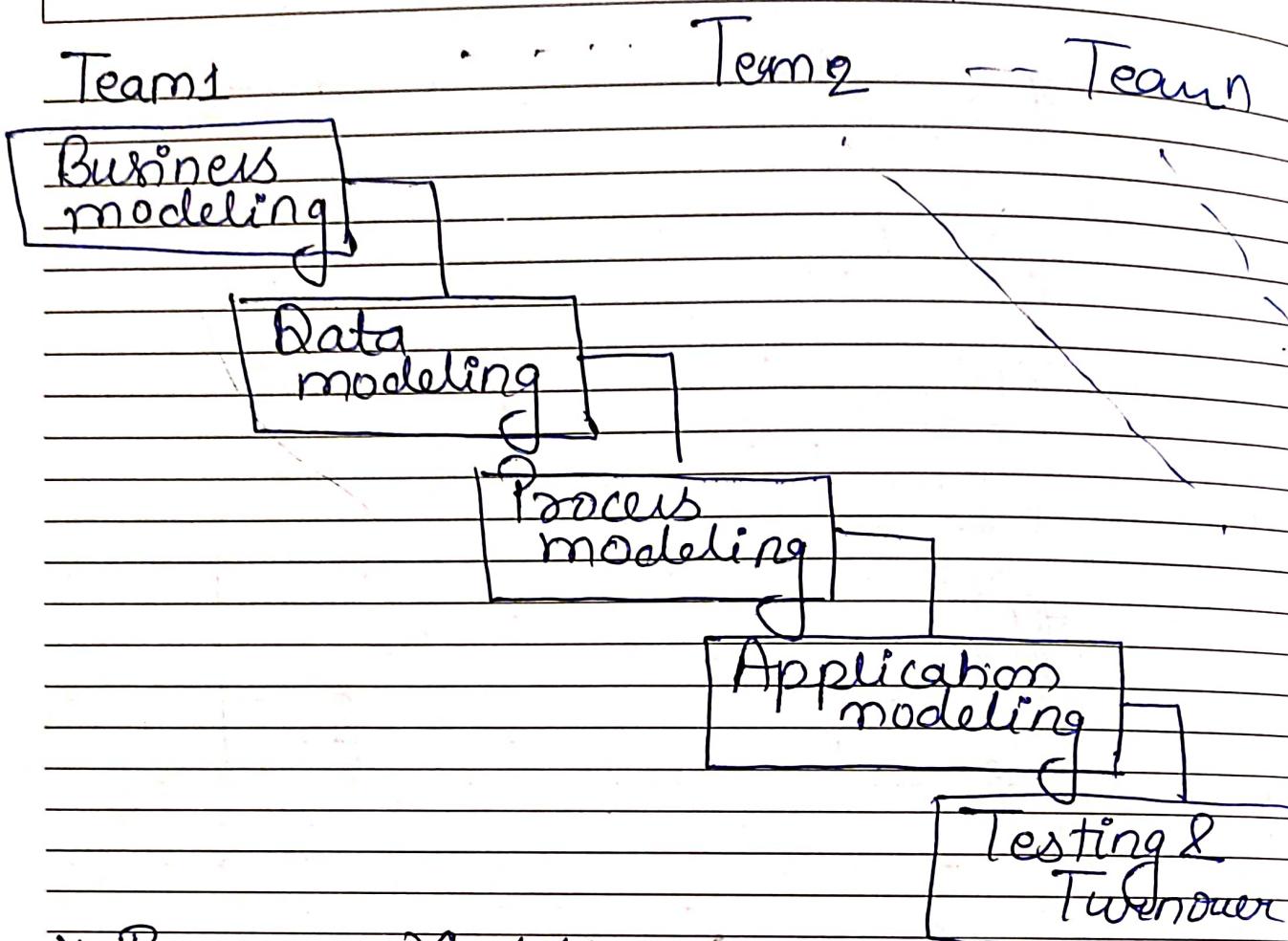
- * If the user is not satisfied by the developed prototype, then a new prototype is developed. This process goes on until a perfect prototype is developed. Thus, this model is time consuming & expensive.
- * After seeing an early prototype the user may demand some actual system to be delivered soon.
- * If end-user not satisfied with initial

prototype, he/she may lose interest in the project.

Rapid Application Development Model

- + In RAD model, the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery.
- + RAP application development is an incremental & iterative development process model that emphasizes an extremely short development cycle. The RAD model is a "high-speed" adaptation of the linear sequential model in which rapid development is achieved by using component-based construction.
- + If requirement are well understood and project scope is constrained, the RAD process enables a development team to create a fully functional system within very short time period.
- + It is high speed adaptation of waterfall model.





* Business Modeling:-

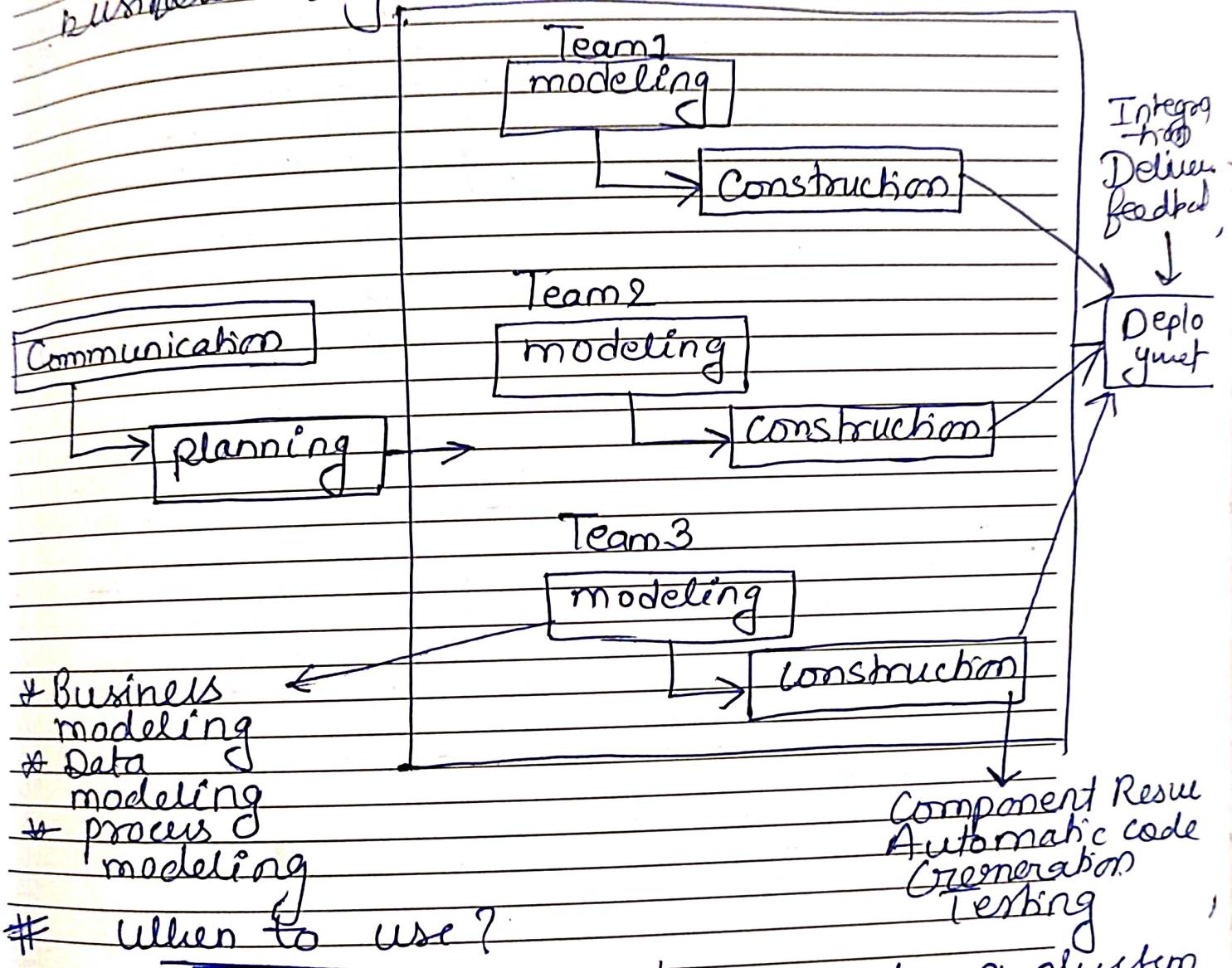
- ⇒ The information flow among business functions is defined by answering questions.
- ⇒ A complete business analysis is performed to find the info for business.

* Data modeling:- The data collected from business modeling is refined into a set of data objects:

The attributes of all data sets is identified and defined

The relation b/w these data objects are established and defined in detail in relevance to the business model.

* **Process Modeling** :- The data objects defined in the data modeling phase are converted to establish flow of the business information needed to achieve specific business objectives.



* When to use?

- * There is a need to create a system that can be modularized in 2-3 months of time.
- * High availability of designers & budget cost for modeling along with the cost of automated code generation tools.
- * Resource with high business needs are available.

Risk

~~# Spiral Model :-~~

→ Advantages :-

- * Reduced development time
- * Increases reusability of components.
- * Quick initial reviews to occur.
- * Encourages customer feedback.
- * Integration from very beginning solves a lot of integration issues.

→ Disadvantages :-

- * For large but scalable projects, RAD req. sufficient human resource.
- * Project fail if developers and customers are not committed in a much shortened time-frame.
- * Problematic if system can not be modularized.
- * Not appropriate when tech. risks are high.

Evolutionary Model :-

- * Evolutionary Software Process model are iterative.
- * In programming "Iteration" means sequential access to object S/W engineers can follow this process model that has been clearly designed to put up a product that's regularly complete over time.

Iterative model :-

Iteration-1

Requirement



Analysis



Design



Testing



Imp. Review



Review

Iteration-2

Design



Testing



Implementation



Review

Iteration-3

Design



Testing



Imp.



Review



Development



maintenance

* When to use :-

* Req. of the customer are clearly defined & understood.

* When the project is big.

30
→ Major req. must be defined, however some details can evolve with time

→ Advantages:

- Generate working SW quickly & easily during SW life cycle.
- Easier to test and debug during a smaller iteration.
- Easier to manage risk because risky pieces are identified and handled during its iteration.

→ Disadvantages:-

- It is not suitable for smaller projects
- It is not suitable for changing requirements

In iterative model the organization starts with some of the SW specification and develops the first version of the SW. After the first version if there is a need to change the SW then a new version of the SW is created with a new iteration.

It will repeat until deployment of the SW.

* Iterative process starts with a simple implementation of a subset enhanced SW req. and iteratively enhances the evolving version until the full system is implemented. At each iteration design modification are made & new fun. capabilities are added.

→ Evolutionary Process Model is of 2 types:-

Incremental Model
Spiral Model

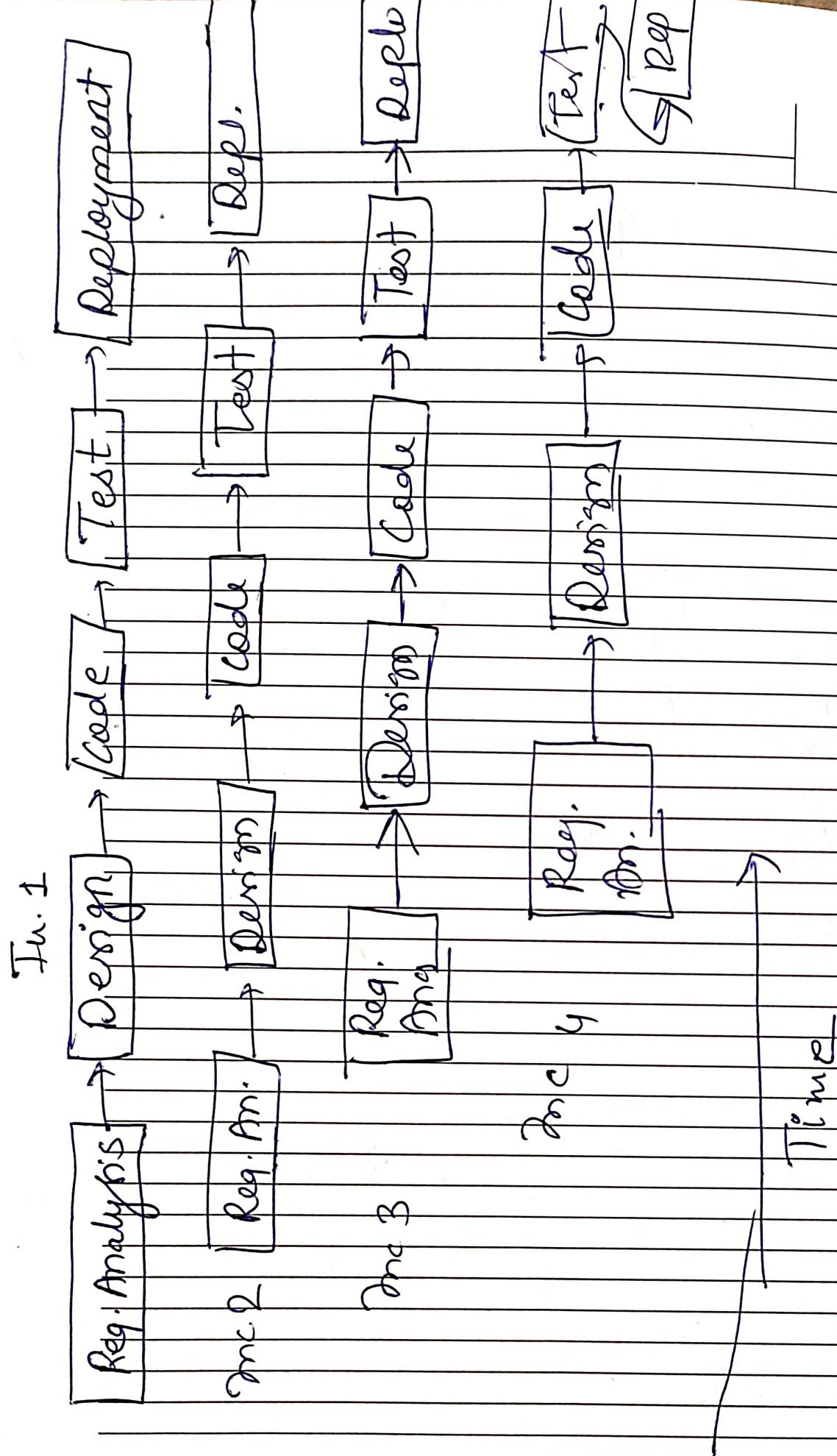
Incremental Model :- In the incremental model, we do not develop the full SW in a single go instead first, we break down the customer's requirements into multiple standalone modules and then develop each module one-by-one using the waterfall approach and increment each module with the previous modules.

So in this model, the whole SW project developed module-by-module and incremented to the previous module.

First, the development team develops the core module of the SW project and after that other modules are developed one-by-one, and each module is increment with the core module and so on.

Core module means first the developers develop the highest priority requirements to when first the core module is developed, it delivered to the customer, so the customer can use it and give us the feedback of the developed SW.

After that other modules are developed one-by-one and incremented to the core module, and delivered to the customer after each increment.



my. 01/8

Advantages :-

* Customer Interaction Maximum :-

Customer interaction maximum because in every increment, the product delivered to the customer.

* Early Customer feedback :- With every release of the product, the customer can give us feedback, which helps the developers to the next module.

* Initial Product delivery is faster :- The product can be delivered at the first initial stage so the customer can use the product early and gives us the proper feedback.

* Lowers initial delivery cost :- Initially, product cost is low, because the product is delivered increment-by-increment, so the customer does not have to pay all costs initially.

* More Flexible :- This model is more flexible because the product developed in stages, so the company does not have to use all of its cost & manpower and also because of customer's feedback the company knows, what the actual customer expectation from the product, so they can full the customer's req. in the next product release.

Disadvantage :-

* This model is used when the requirement are well understood, so it needs good & effective planning.

* It needs more cost & also needs more time.

* Interactive Model:-

- * Interactive model is based on a cyclic process of prototyping, testing, analyzing, and refining a product or process.
- * Based on the results of testing the most recent iteration of a design, changes and refinements are made.
- * This process is intended to ultimately improve the quality & functionality of a design.
- * In iterative design, interaction with the designed system is used as a forum of research for refining and evaluating a project.
- * Thus, here it focuses on initial and simple implementation first, which then progressively adding more complexity to it until the final system is complete.

* When to use:-

- * Final S/w req. are well-defined.
- * Large-sized projects
- * Market constraints are well-known and by the time the S/w gets released, it still be applicable.
- * Scope for new Tech. to incorporate.

* Advantage:-

- * Every iteration results in partial operational S/w.
- * Measurable progress.
- * Risks are easy to control and mitigate within the iteration.

- * Each iteration is a lesson to improve the process and eliminate the repetition of mistakes.
- * Incorporate the change requests, customer evaluation and feedback is facilitated which enhances the quality.

Disadvantages:-

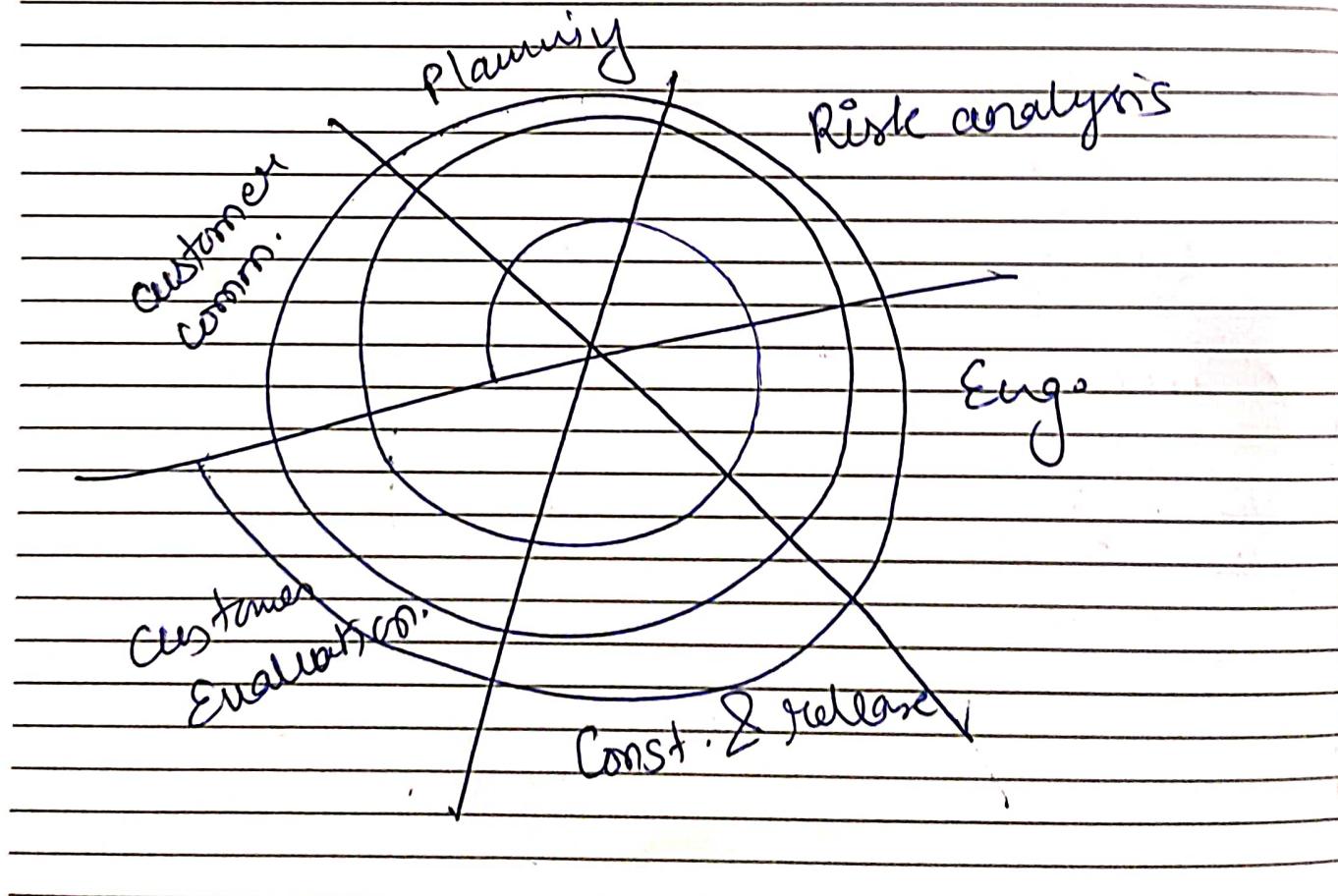
- * Management is the continuous activity to be handled.
- * Need of more resources and planning.
- * Risks cannot be determined for the later stages.
- * Risk Management requires highly skilled resources.

When to use Iterative :-

- * Req. of the system are clearly understood.
- * When demand for an early release of a product arises.
- * When SW engg. team are not very well skilled.
- * Large size project.
- * Project with new technology.

Spiral Model :-

- * Risk analysis of s/w.
- * Spiral model follows a risk-driven approach for s/w development.
- * It is a combination of waterfall and iterative models.
- * In this, the s/w undergoes through four phases iteratively in a "spiral" until it is completed fully allowing for multiple rounds of refinement.
- * This model allows in building of a highly customized product.
- * User feedback can be incorporated from early on in the project.



- * Customer Comm:- task required to establish effective comm. b/w developer & customer.
- * Planning:- task req. to define resources, timelines and other project related information.
- * Risk analysis:- task required to assess both technical & management risks.
- * Engineering:- task req. to build one or more rep. of the application.
- * Construction and release:- task req. to construct, test, install and provide user support (eg. documentation & training).
- * Customer evaluation:- task required to obtain customer feedback based on evaluation of the SW rep. created during the Engg. Stage and implemented during the installation Stage.

When to Use:-

- * When the project is large.
- * When req. are unclear & complex.
- * When changes may require at any time.
- * Large and high budget projects.
- * When risk and costs estimation is imp.

Advantages:-

- * Risk analysis at early stages
- * Suitable for large projects
- * Flexible
- * customer satisfaction.

~~#~~ Disadvantages :-

- * Complex
- * Expensive
- * Too much analysis of Risk Analysis
- * Time consuming

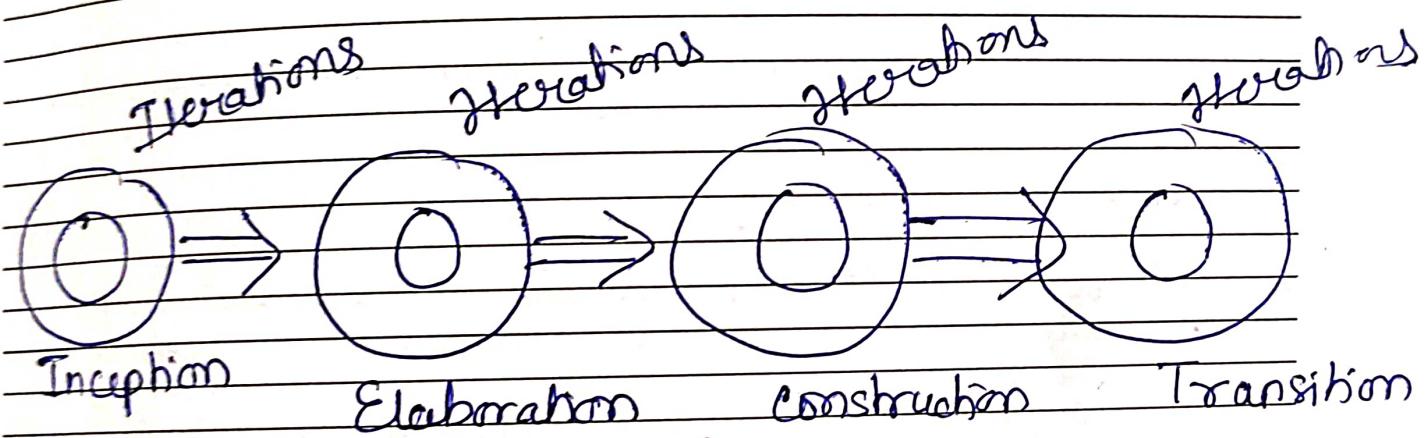
Req. highly specific expertise

~~#~~ Rational Unified Process :-

- * It is also a slow development process for object oriented Model
- * RUP methodology uses the object oriented approach in its design & the use of UML.
- * It's an iterative development methodology based upon an industry-profiled best practices.
- * Features of RUP:-
 - Iterative Development
 - Requirement management
 - Visual modeling of system
 - Quality management
 - Change control management
- * Iterative development help in continuously iterations with feedback & adaptation.
- * Iteration are time boxed.
- * RUP focuses on addressing the needs of both lightweight small projects & large projects.

Phases in RUP:-

- * RUP is divided into four phases, named:
 - * Inception
 - Elaboration
 - Construction
 - Transition
 - * Each phase has iterations, each having the purpose of producing a demonstrable piece of software. The duration of iteration may vary from two weeks to six up to nine months.



Inception :-

- * Objective of the projects
 - * Acceptance criteria
 - * Some req. are established
 - * An estimation of resources
 - * An initial risk assessment

Elaboration:-

- Elaboration:-

* Elaboration means describing something in more details.

→ mention a management plan, a phase plan, defining the no. & contents of the iteration (and a test plan prepared).

* Here we go into more detail of 1st phase.

* Redefine if we feel need (cancel project as well) if needed.

* At the end a detailed S/w development plan with an updated risk analysis

Construction:-

* Here we develop and complete the project based on the data we get from previous stages.

* Coding is done here.

* All kind of testing are also done here.

Transition:-

* Here finally project is transit from development environment to production.

* Here we also set the project on beta testing mode.

* Remove the bugs from project based on customer feedback.

Production:-

* Project is maintained here.

* Project is updated here accordingly.

* Analysis of the performance of the project relative to its original & revised success criteria is taken care.

* It involves marketing, packing, installing, configuring, etc.



Business training
Project management
Implementation
Test
Reporting

The Agile Model in SE :-

- * The term Agile means "The ability to respond to changes" and here changes refer to changes in terms of Req., Tech., and people.
- * Agile model is combination of iterative & Incremental process of model.
- * It is basically a methodology that promotes continuous development and testing of the software throughout the SDLC of the project.
- * In this the product is broken into small build and these build are provided in iterations.
- * Iteration typically lasts from about two to four weeks.
- * Each build is incremental in terms of features and the final build holds all the features req. by the customer.

When to use:-

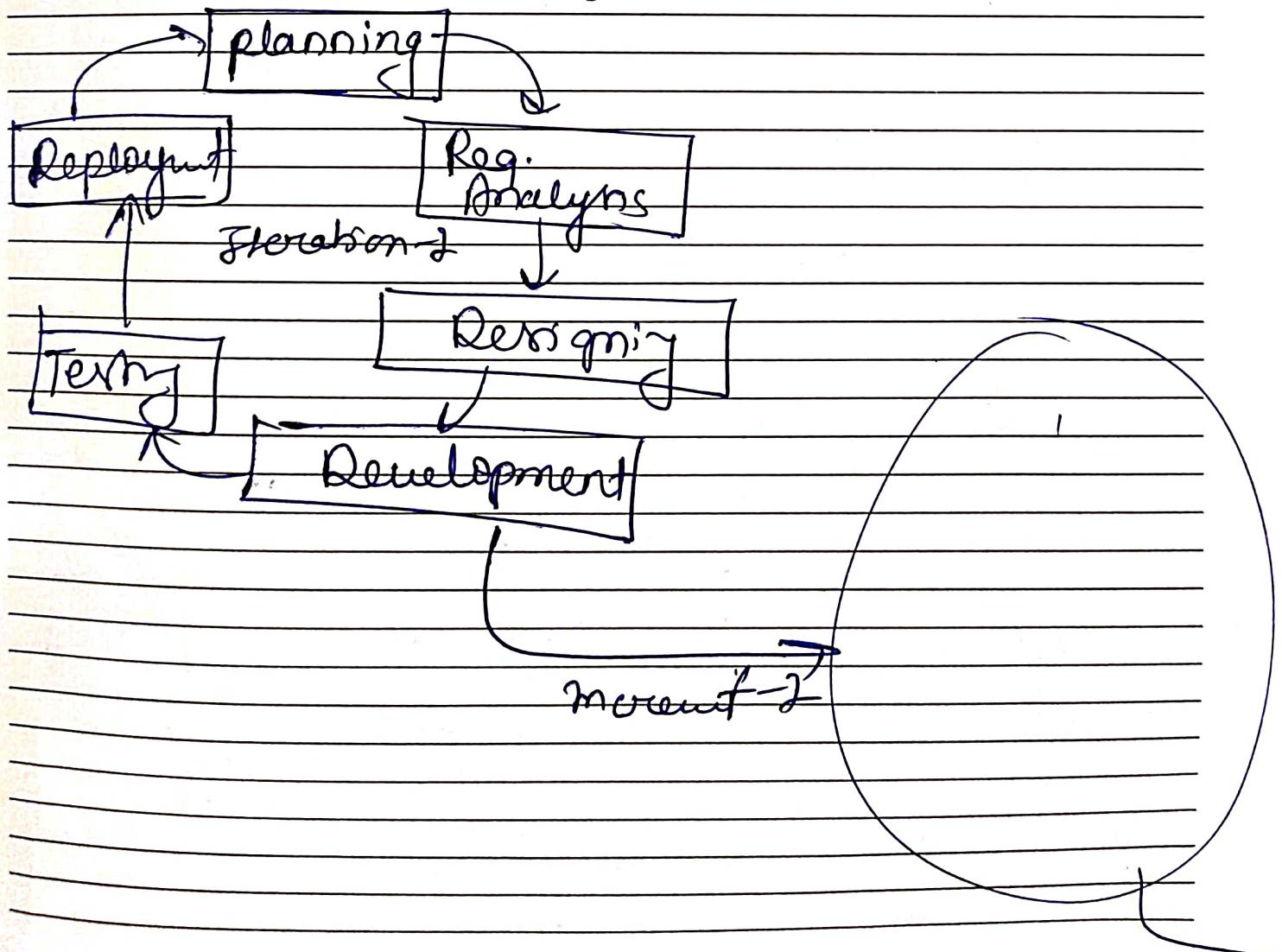
- * frequent changes are req.
- * highly qualified & experienced team is available
- * Customer is ready to have a meeting with a SW team all the time

Advantage

- * Fast Delivery
- * Efficient design & fulfills the business req.
- * Anytime changes are acceptable.
- * fast feedback
- * Improve quality

Disadvantage :-

- * Less documentation
- * maintenance problem
- * Req. Highly qualify team



Capability Maturity Model:-

- * CMM was developed and is promoted by the Software Engineering Institute (SEI) a research and development center.
- * It is not a S/w process model. Capability Maturity Model is used as a benchmark to measure the maturity of an organization's S/w process.
- * CMM is basically a framework which is used to analyzes the capability and path followed by any org to develop a S/w product.
- * It provides the guidelines to further enhance the maturity.
- * There are 5-different level of CMM
- * Each level shows a process capability level.

Level One - Initial :-

work is performed Informally:-

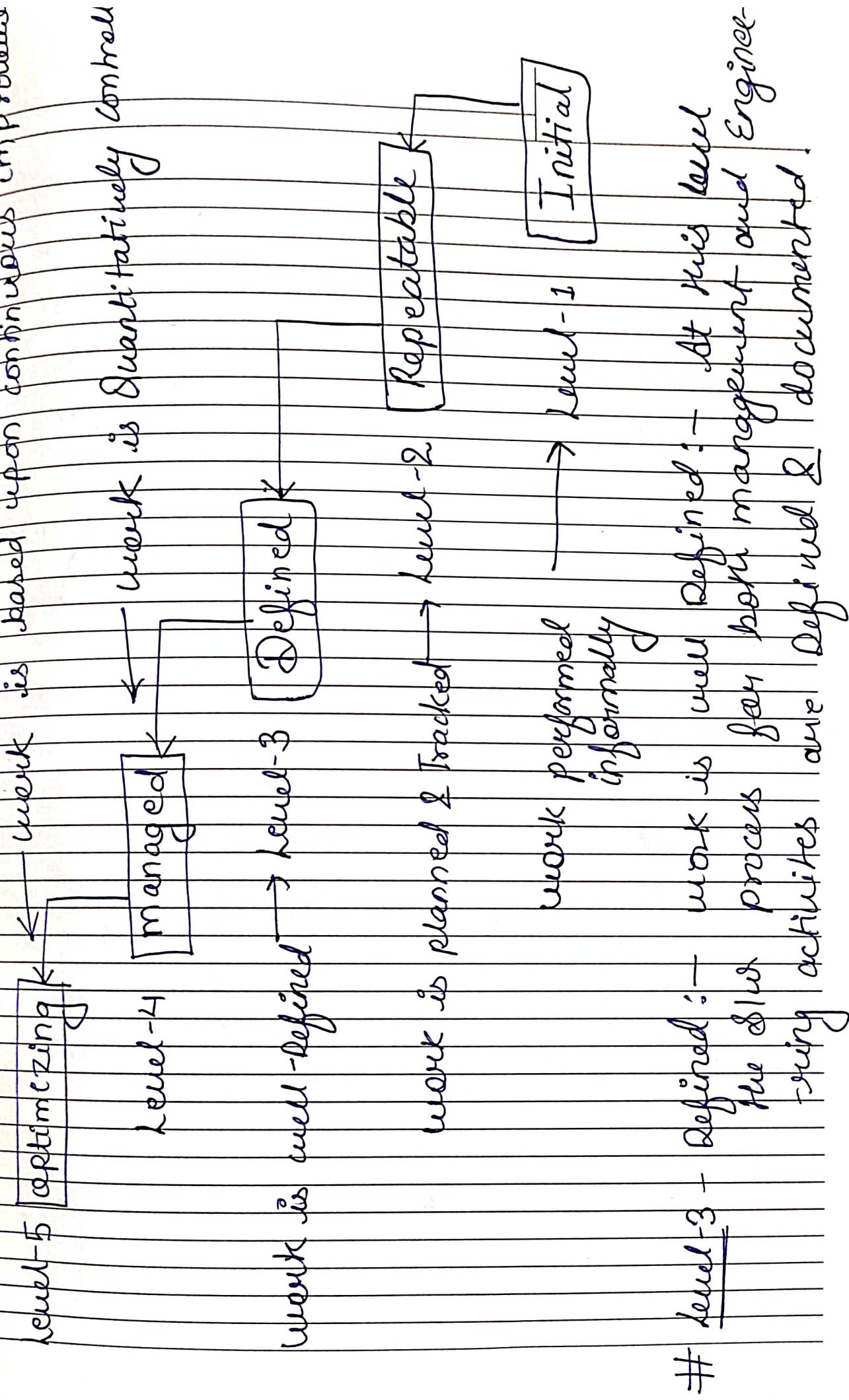
A S/w development organization at this level is characterized by ad hoc activities (organization is not planned in advance).

Level Two - Repeatable:-

work is Planned & Tracked:- This level

development organization has a basic and consistent project management processes to track cost, schedule and functionality. This process is

in place to repeat the earlier successes on projects with similar applications.



Level-4 : Managed :-

work is Quantitatively Controlled :-

SIL Quality Management :- Management can effectively control the SIL development effort using precise measurement. At this level, organization set a quantitative quality goal for both SIL process & SIL maintenance.

Quantitative Process Management :- At this maturity level the performance of processes is controlled using statistical and other quantitative techniques, and is quantitatively predictable.

Level Five :- Optimizing :-

work is based upon continuous improvement :-

The key characteristic of this level is focusing on continuously improving process performance key features are :-

- * Process Change management
- * Technology Change management
- * Defect prevention.