

# 1) Overview of Software Engineering and software Development Process

Page No.	1
Date	

## # Definition of Software :-

- ① Set of instructions that when executed provide required function and performance
- ② data structure that enable the program to adequately manipulate information
- ③ documents that describe the operation and use of the program

## # Characteristics of Software :-

- ① Software is developed or engineered; it is not manufactured in the classical sense.
- ② Software doesn't wear out
- ③ Although the industry is moving toward component based assembly, most software continues to be custom built

## # Types / Categories of Software :-

- ① System software :- System software is a collection of programs written to service other programs. It does heavy interaction with computer hardware.  
Ex:- Compilers, Editors, OS, Drivers, etc.
- ② Real time software :- Software that analyses / controls real world events as they occurs is called real time system. It possesses very less response time ranging
- ③ Business software :- Business software process information related with business organization. It handles volume of data stored in database.  
Ex:- Payroll, Inventory Systems

④ Engineering and scientific software:- They have been characterised by "number crunching" algorithms. These are used for scientific purposes.

⑤ Embedded software:- Embedded software resides in ROM and used to control products and systems for consumer and industrial markets.

⑥ Personal Computer Software:- The personal computer's software used on PC's

⑦ Web-based Software:- They reside on web. These can be used for online trading, communication and other purposes.

⑧ AI software:- Artificial Intelligence (AI) software makes use of non-commercial algorithms to solve complex problems.



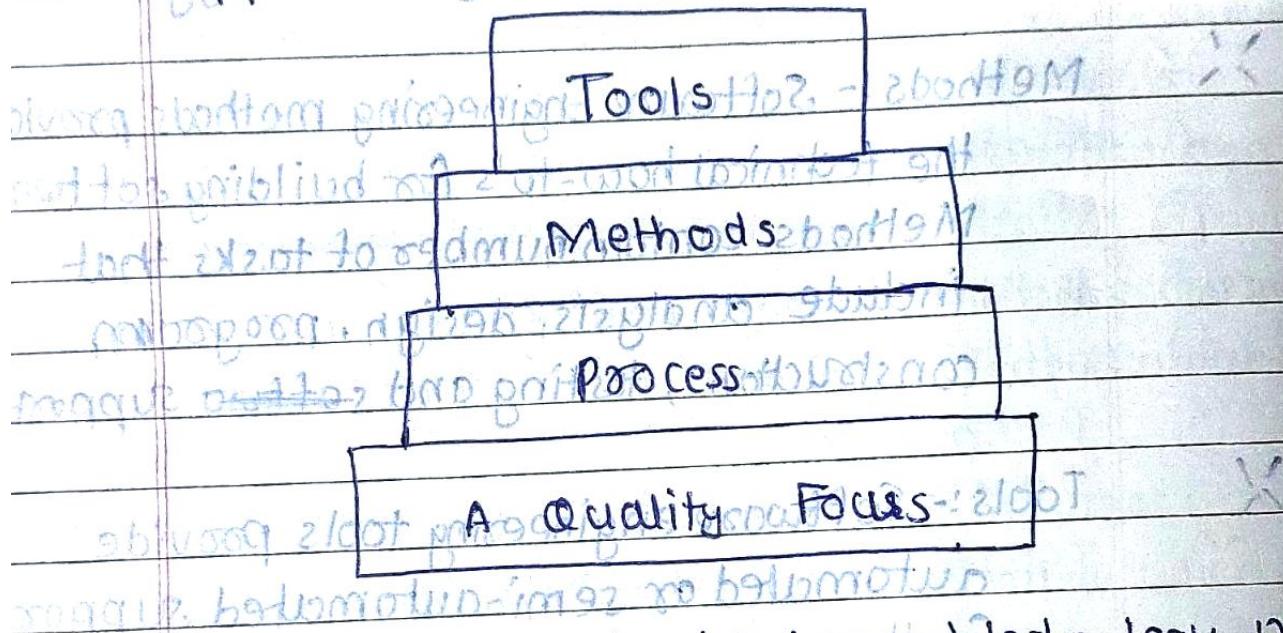
### Need of Software Engineering:

- ① Software projects are complicated. They need scientific and engineering approach to develop
- ② Software Engineering is needed for building complex and critical softwares within time and budgets.
- ③ Scheduling and controlling are the main activities of software project guided by software engineering tools.
- ④ The projects has to be divided into processes, framework activities, tasks, etc.
- ⑤ Different models are required for designing and analysis without which developing software is impossible.

- ⑥ There is huge management of resources at every stage.
- ⑦ Project teams have to continuously deal with time and new technology challenges.
- ⑧ The art of programming only is not sufficient to construct large softwares.

## # Software Engineering - A layered Technology

Approach: programming growth



Software Engineering is a layered technology. Due to this software Engineering process has become more and more matured.

\*) A Quality Focus :- Any engineering approach must rest on an organisational commitment to quality. Total quality management develops a continuous process improvement culture, and this leads to the development of more mature approach to software engineering. The bedrock that supports software engineering is a quality focus.

Process :- The foundation for software engineering is the process layer. Software Engineering process is the glue that holds the technology layers together and enables rational and timely development of computer software. Process defines a framework for a set of key process areas (KPA) that must be established for effective delivery of software engineering technology.

Methods :- Software Engineering methods provide the technical how-to's for building software. Methods contain number of tasks that include analysis, design, program construction, testing and ~~soft~~ support.

Tools :- Software engineering tools provide automated or semi-automated support for the process and methods.

# Software development and generic process framework

→ A process framework is a collection of software activities which forms a complete foundation or base for software development process or software engineering process. It is the basic structure of software development process.

## Basic framework activities:

- ① Communication :- It involves heavy communications and collaboration with the customer or stakeholder and encompasses requirement gathering and other related activities.
- ② Planning :- It establishes a plan for SE work that follows. It describes technical risks, resources, product to be produced and work schedule.
- ③ Modelling :- It encompasses the creation of models to understand requirement and the design to achieve the user requirements.
- ④ Construction :- It combines code generation and testing to discover errors.
- ⑤ Deployment :- Software is delivered to the customer who will use it to provide a feedback about it.

## Umbrella activities / Complement Activities :

- ① Software Project Tracking and Control
- ② Formal Technical Review
- ③ Software Quality Assurance
- ④ Software Configuration management
- ⑤ Document preparation and generation
- ⑥ Reusability management
- ⑦ Measurement
- ⑧ Risk Management.



PSP & TSP :- PSP and TSP are the Personal Software Process and Team Software Process models. It is important to study and understand these both models for successful development of the expected software product.

PSP focuses on the individual effort in the development of the process and TSP focuses on the efforts taken by the team to improve the software process. - gaillatowM

PSP framework activities:

- ① Planning
- ② High level design
- ③ High level design review
- ④ Development
- ⑤ Postmortem



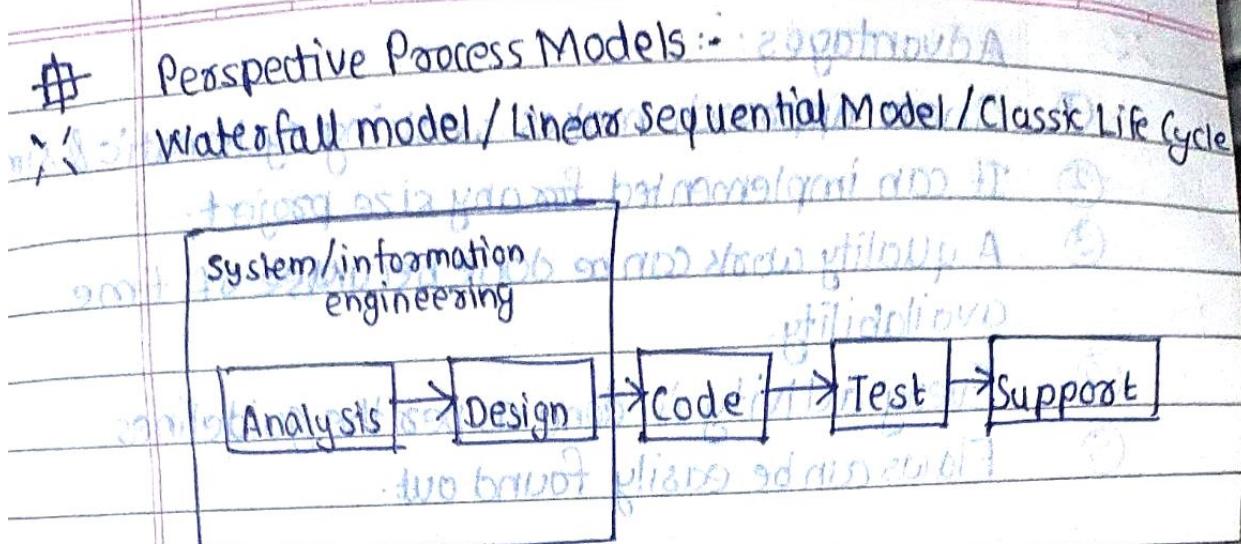
TSP framework activities:

- ① Plan
- ② Launch
- ③ High level design
- ④ Implementation
- ⑤ Integration and test
- ⑥ Postmortem

RiM

M

riM



It suggests a systematic, sequential approach to software development that begins at the system level and progresses through analysis, design, coding, testing and support.

- ① Software requirements analysis:- User requirements are gathered and finalized by doing communication with the customer.
- ② Design:- It is multi-step process that focuses on four distinct attributes of a program: data structure, software architecture, interface representations and algorithmic detail.
- ③ Code generation:- In this design is converted into programs.
- ④ Testing :- It focuses on testing the system to ensure that all errors are removed and the software works with high quality and without any problem. It is done at every stage of waterfall model.
- ⑤ Support:- software may need to modify after delivering it to customer due to errors & change in business environment. Support phase handles this issue.

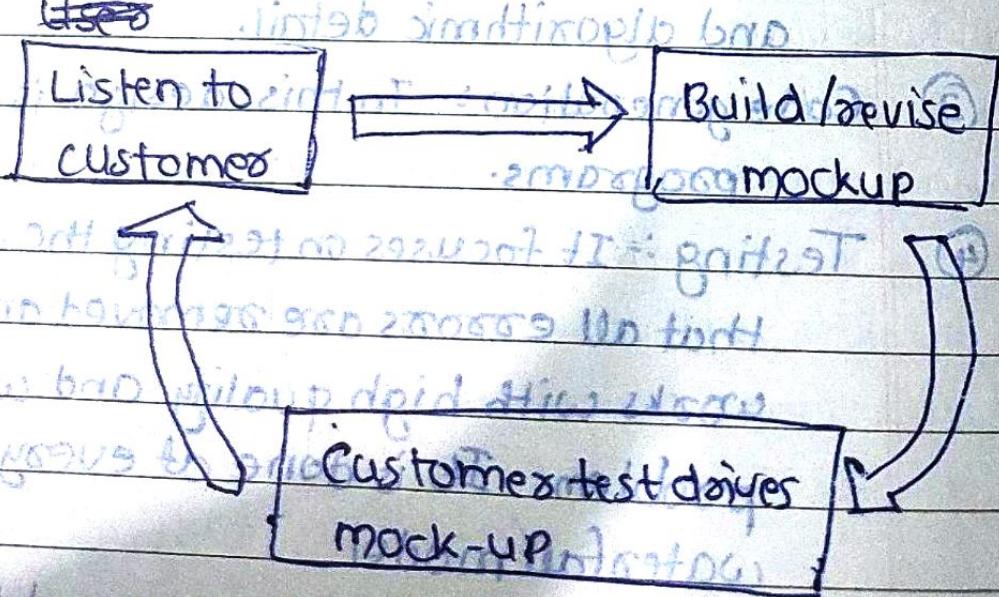
**Advantages:** - global reach with 92% of the world's population.

- ① Linear approach makes working systematic & simple.
  - ② It can be implemented for any size project.
  - ③ A quality work can be done because of time availability.
  - ④ Every activity is done as per the guidelines.
  - ⑤ Flaws can be easily found out.

**Disadvantages:** it costs a lot of money. It's not good for the environment.

- ① As real projects rarely follows the sequential flow; this model is not so useful.
  - ② Blocking states may arise due to linear nature of waterfall model.
  - ③ This model is not flexible for change.
  - ④ This model is time consuming and not cost effective.

Prototyping model > Hydistic fruitib



Prototyping is defined as the process of developing a working replication of a product or system that has to be engineered. Prototyping model is used when customers do not know the exact project requirements. A prototype of end product is developed first, tested and refined as per customer feedback repeatedly till a final acceptable prototype is achieved which forms the basis for developing the final product.

#### Advantages:

- ① Customers can see the prototype earlier, so customer knows what actually he is getting.
- ② Unexpected requirements can be accommodated.
- ③ Allows for flexible design & development.
- ④ Interaction with the prototype simulates awareness of additional needed functionality.
- ⑤ Steady, visible signs of progress produced.

#### Disadvantages:

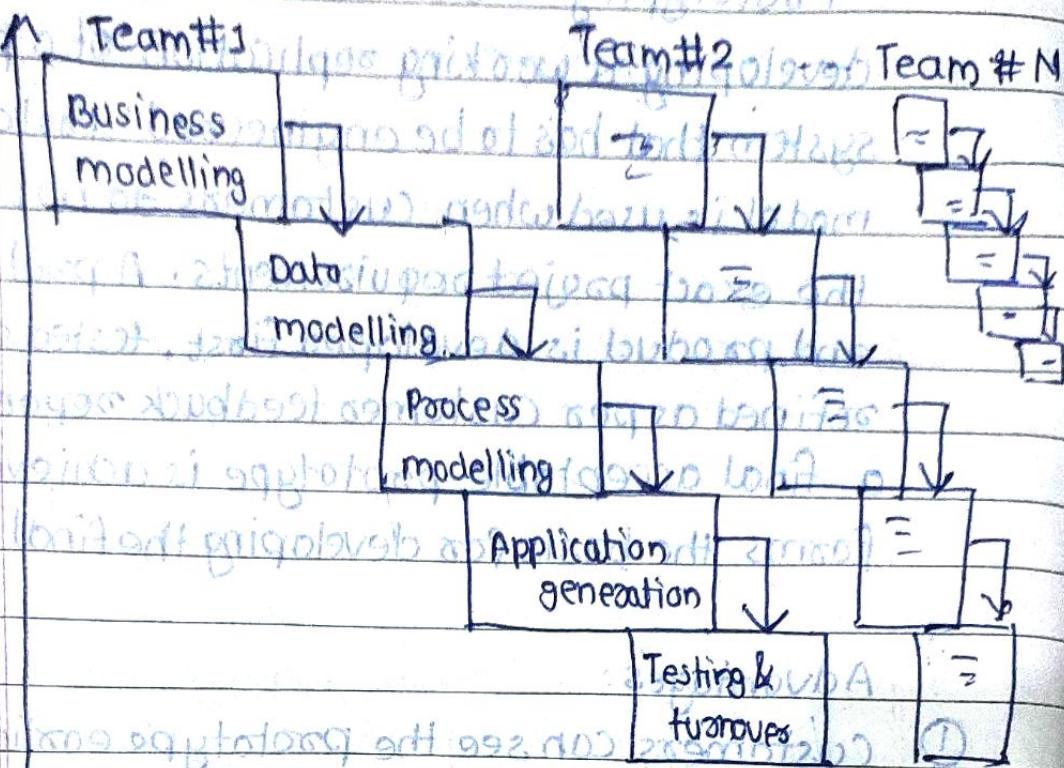
- ① Due to rush to get it product done quickly, affects overall quality.
- ② Product may need to be built to maintain high quality.
- ③ Chances of getting less-than-ideal system is too high.

grillborn 2292069

noitorgnag noitni/99A

noitorgnag noitni/99A

RAD model



Rapid Application Development is an incremental software development process model that

emphasizes an extremely short development cycle. It is a "high speed" adaptation of the linear sequential model in which rapid development is achieved by using component based construction. It consists of 5 procedures that each team follows for rapid development: Unsgva

- ① Business modelling
  - ② Data modelling
  - ③ Process modelling
  - ④ Application generation
  - ⑤ Testing and turnovers

### Advantages:

- ① It works with great speed; so productivity is increased.
- ② It includes defining of customer requirements.
- ③ Focus moves from documentation to code
- ④ uses modelling concepts to capture information about business, data and process.

### Disadvantages:

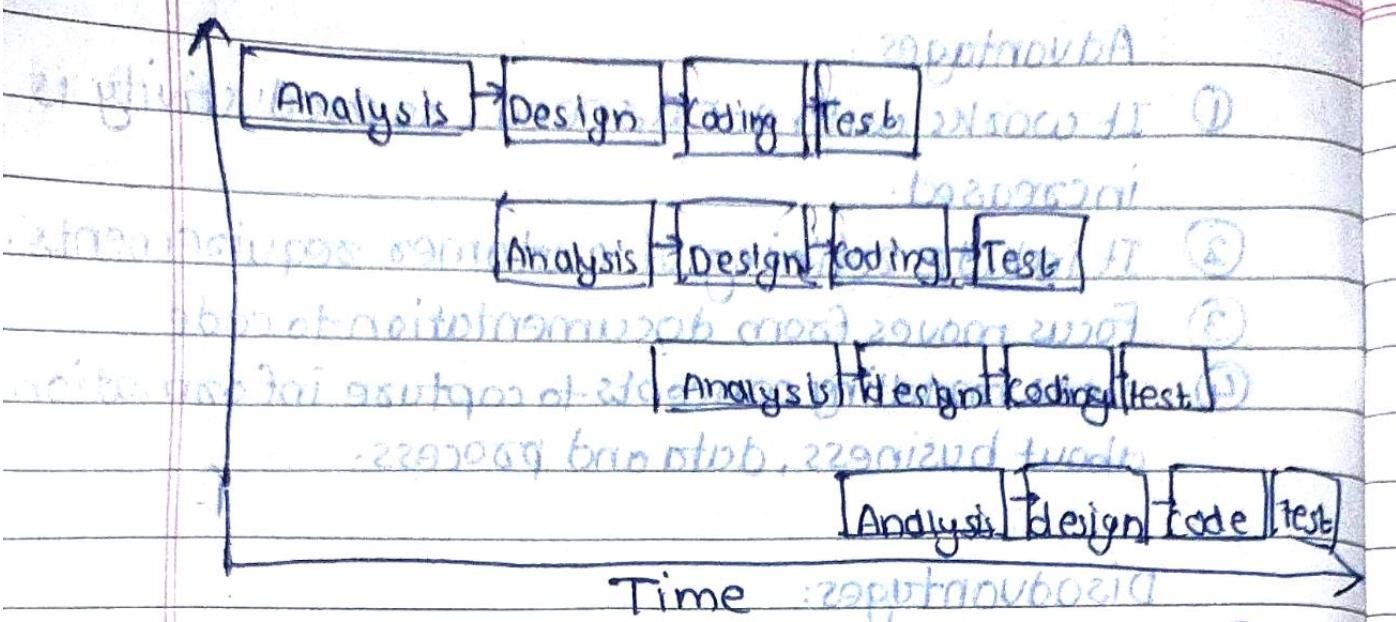
- ① RAD requires sufficient human resources to create the right number of RAD teams.
- ② Not appropriate for all types of applications.
- ③ RAD approach does not work properly if high performance is a major issue.
- ④ RAD may not be suitable when technical risk is high.

### # Evolutionary Software Process models:

↳ Incremental model: - The incremental build model is a method of software development where the product is designed, implemented and tested incrementally until the project is finished. The product is defined as finished when it satisfies all of its requirements.

- Progressed with the help of feedback

beginning with a small subset of requirements

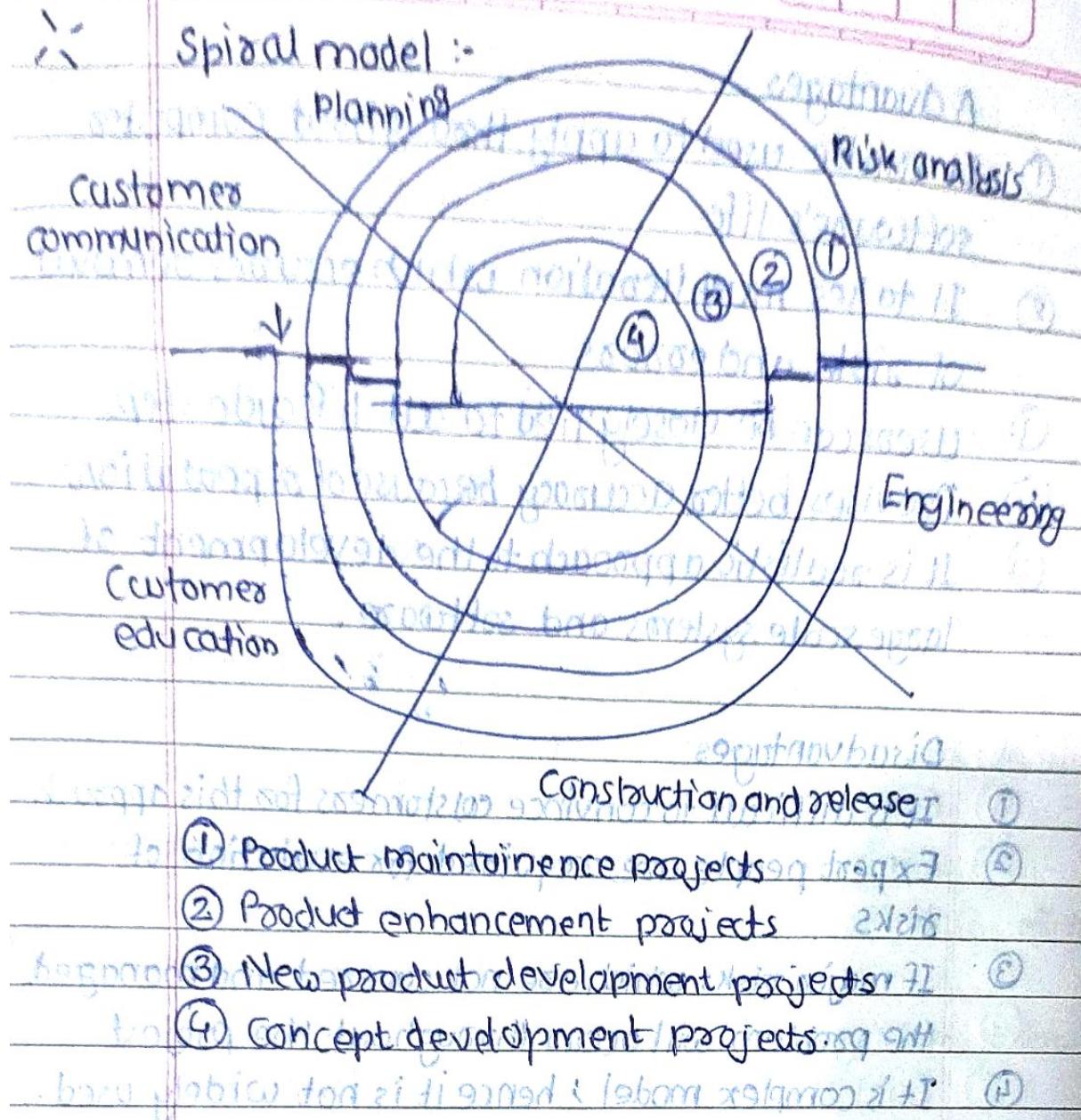


Advantages:

- ① Initial product delivery is faster.
- ② Each release delivers an operational product.
- ③ Uses "divide and conquer" breakdown of tasks.
- ④ No problem of resource requirement or staffing upfront.
- ⑤ Increments can be planned to manage technical risks.

Disadvantages:

- ① Each phase of iteration can be rigid and may not overlap each other.
- ② Iterations may create confusions.
- ③ Problems may occur as all requirements are not gathered at the beginning.
- ④ Well defined module interfaces are required.



Spiral model is one of the most important Software Development Life Cycles (SDLC) model, which provides support for Risk Handling. Each loop of the spiral is called as phase of the software development process. It may contain 3 to 6 regions:

- ① Customer communication      ② Planning
- ③ Risk analysis      ④ Engineering
- ⑤ Construction and release
- ⑥ Customer evaluation

### A dvantages:

- ① It can be used throughout computer software's life.
- ② It takes much iteration which ensures removal of risks and errors.
- ③ Users can be closely tied to all lifecycle steps.
- ④ Provides better accuracy because of repetitions.
- ⑤ It is realistic approach to the development of large scale systems and software.

### Disadvantages:

- ① It is difficult to convince customers for this approach.
- ② Expert people are required for evaluation of risks.
- ③ If major risks which are not covered and managed, the problems will occur throughout the project.
- ④ It is complex model; hence it is not widely used.

# (Agile software development) (Q. generator)

Agile software development (Read it from notes) —

— H-30 92nd q-20 bally 21. 10/12. 9/11 70 901

introduction II - 22970159 dr. nqolavab 960wt702,