

# SOFTWARE ENGINEERING

## \* Software :

It is a collection of integrated prgms.

Engineering is the application of scientific & practical knowledge to invent, design, build, maintain and improve frameworks, processes, etc.

## \* Software eng:

Engineering branch related to the evolution of software product using well-defined scientific principles, techniques and procedures.

The result of s.eng is an effective and reliable software product.

## → Importance of s.eng:

### 1) Reduced complexity:

It offers structured approaches like modular design, abstraction which help in breaking down complex systems into manageable components.

2) minimised software cost

3) reliable software

4) decreased time

## (5) -> Software.

- 5) Handling big projects
- 6) Effectiveness

### -> Best practices of Software eng:

- \* It is a broad array of concepts, principles, methods & tools that a S. eng must consider as Software I planned & developed

- \* Essential components of S. eng practice
- (1) Understand the probm:

- (2) Plan a solution:
- communication & analysis.

- (3) Carry out the plan:
- modelling & (5) design.

- code generation.

- (4) Examine the rslt for accuracy:
- testing & quality assurance

### -> (8) development myths (conclusion):

- \* They are beliefs that (5) managers, customers & developers believe basedly.
- \* myths have a no. of attributes <sup>the</sup>

make them dangerous.

- \* A greater majority of (5) related probms arises bcz of the myths that formed during the (5) development's initial stages.

- \* Some are:

### Software management myths:

- \* Most of the pressure & strain are on the managers, who own the responsibilities, including keeping the (5) costs to budget & improving quality.
- \* Adding more managers guarantees better productivity.
- \* Rushing development without adequate testing & quality assurance can lead to more bugs

### II Software customer myths:

- \* Customers may have a general idea of their needs.
- \* Customer feedback is invaluable for improving software products.
- \* Adding more features will satisfy all customers.



- \* Customer don't care about security & privacy.

### III Software developer myth:

- \* Effective communication are vital for successful collaboration on complex projects
- \* Technology is continuously evolving & developers must continue learning to ~~stay~~ improve their skills.
- \* Best developers are the fastest coders

### ⇒ Software process

- \* S. process is a set of related activities, actions & tasks that leads to the production of a (S) system.
- \* A (S) process framework is a standard conceptual generic process model of a complete S. eng process.
- \* A generic process framework for (S) process defines 5 activities which are applicable to all projects & applications:

#### (1) Communication:

It involves deep communication with the

Stakeholders → developers, designers, project managers, ~~testers~~ users, etc.

- (2) Planning:  
Describes the technical tasks to be conducted, the risks that are likely, resources that will be required & a work schedule.

#### (3) Modelling:

Involves creation of models that allow the developer & customer better understand software requirements & the design.

#### (4) Construction:

Involves the code generation & testing required uncovering errors in the code.

#### (5) Deployment:

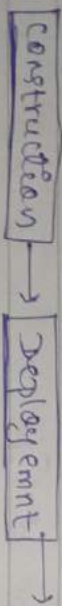
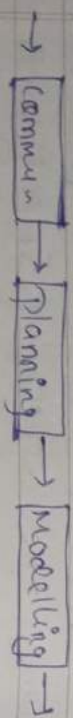
During this activity, (S) is delivered to the customer & provides feedback based on their evaluation.

- Above 5 S. eng process framework activities are usually complemented by a no. of parallel activities, 1) Risk management

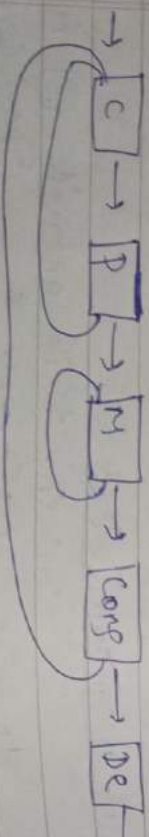


- 2) Quality assurance
- 3) Technical review.
- 4) Requirement
- 5) Software configuration management.
- 6) Reliability management.

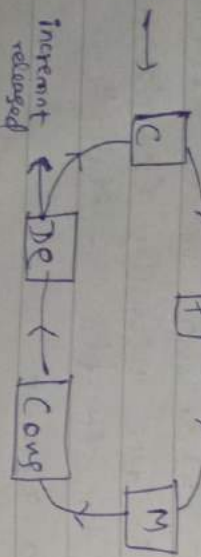
\* Linear process flow executes each of the 5 framework activities in sequence



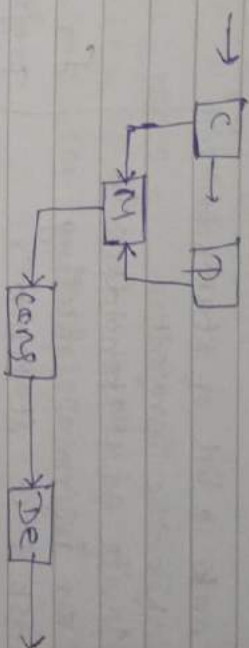
\* Iterative process flow repeats 1 or more of the activities before proceeding to next



\* Evolutionary process flow executes the activities in a circular manner



\* Parallel process flow executes 1 or more activities in parallel with other activities



→ Task sets:

\* The process framework defines a small set of activities that are applicable to all types of projects.

\* Each framework activity can be represented by a collection of task sets.

\* A task set defines the actual work to be done to accomplish the objectives of a S.Eng action.

\* eg → requirements gathering is an imp S.Eng action that occurs during the Commu-Framework activity.

\* Different task set:

- a) work tasks
- b) work products

c) quality assurance points  
d) ~~project~~ project milestones.

- \* make a list of stakeholders for the project.
  - \* Invite all stakeholders to an Informal meeting
  - \* Ask each stakeholder to make a list of needed requirements
  - \* Discuss requirement
  - \* Build a final list
  - \* Note areas of uncertainty
- for a small project, the task set for requirements gathering.

### → Software process patterns:

- \* process pattern describes a process-related problem that is encountered during software eng work.
- \* It identifies the environment in which the problem has been encountered
- \* Suggests 1 more proven soln to the problem.

\* patterns can be defined at any level of abstraction.

\* Types of P-patterns:

1) Stage patterns: defines a problem associated with a framework activity for the process. It includes multiple task pattern as well.

eg → establishing communication  
2) Task patterns: defines a problem associated with a specific action/work task.  
eg → requirement gathering.

3) Phase patterns: defines a sequence of framework activities that occur within the process, even when the overall flow of activities is iterative in nature.  
eg → prototyping.

### → (5) assessment & improvement:

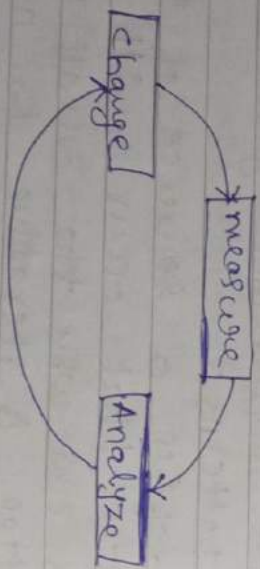
- \* Assessment attempts to understand the current state of the software process with the intent of improving it.



\* The process needs to be assessed to evaluate methods, tools & practices that are used to develop & test the software for process improvement.

\* Software process improvement (SPI) involves understanding the existing processes & changing those processes to increase product quality & reduce cost & development time.

\* General process improvement is a cyclical process -



\* Stages are:

1) Process assessment:

During this step, if more attributes of the (S) process are measured.

2) Process analyze:

Current process is assessed & process

weakness & bottlenecks are identified.

3) Process change:

process changes are proposed to address some of the identified process weaknesses.

• Approaches used for assessing (S) process →

a) CMMI: (Capability Maturity Model Integration)

\* Based on a set of system & engineering capabilities that should be present as organizations reach different levels of process capability & maturity.

\* CMMI's 5 maturity levels →

• Initial level: process are disorganized.

• Repeatable level: Basic project management techniques are established.

• Defined level: here, an organization has developed its own

standard (S) process.

• Managed level: here, an organization

monitors & controls its own processes through data collection.



• optimizing level: here, processes are constantly being improved through monitoring feedback from Cmt processes.

\* Capability levels →

- level 0 : incomplete
- level 1 : performed
- level 2 : managed
- level 3 : defined
- level 4 : quantitatively managed
- level 5 : optimized.

b) SPICE : [5 process improvement & capability determination].

\* It is a standard used for both process improvement & process capability determination.

\* It also develops a maturity model, known as SPICE reference model, that comprises 6 levels —

a) Not performed :

no identifiable products are created.

b) performed internally ; depends on individual knowledge

c) planned & tracked :  
Identifiable products.

The processes & products are verified according to the preceding standards & requirements.

d) well-defined

e) quantitatively controlled.

f) continuously improved.

c) ISO 9001:2000 :

[International organization for standardization established a standard → ISO 9001:2000].

\* It is to determine the requirements of quality management system.

\* main adv of this is that it achieves a better understanding & consistency of all quality practices.

⇒ SDLC : (Software development lifecycle)

\* SDLC is an approach for making (S) for the developer, user & customer.

\* SDLC focus on the internal phase to the end phase for making products (S).



\* SDLC has some specific phases →

a) Project identification:

Here, the analyst (analyst) set up some meeting with the corresponding client for making the desired (S).

b) Feasibility study:

Feasibility studies in 3 views for making particular (S) for the client.

- Technical (T)
- Financial (F)
- Social (S)

c) System analysis:

Analyst defines how & what type of desired (S) we have to make for the client.

It is a pen & paper process.

d) System design:

The analyst draw the corresponding diagrams related to the particular (S). The design include in the form of flow chart, data flow diagram.

e) System development:

Development refers in the form of coding, error checking & debugging for the particular (S).

f) System testing:

Testing refers whether analyst & developer done well it be correct & error free to the desired (S).

g) System implementation:

After completing the testing, we have to implement a particular product or system according to the customer need.

h) System maintenance:

After implementation, the users use the particular (S) to there corresponding operation to achieve their job.

Also, the related hardware, (S) & other utilities are also maintained.

i) System documentation:

Documentation refers the approach & guidelines for the user as well.



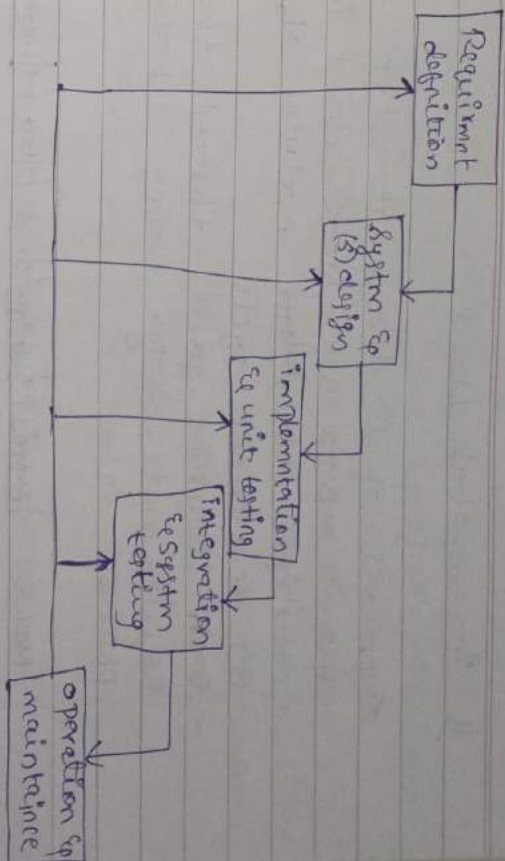
as the customer to the related (S).  
It is some writing instruction  
for how to use the related  
hardware requirement.

- \* Descriptive model describes the history of how a particular (S) system was developed.
- \* Prescriptive model describes how a new (S) system should be developed.

→ Prescriptive process models:

1. Waterfall model:

- \* Also → linear sequential model.
- \* Here, the whole process of (S) development is divided into separate phases.
- \* The outcome of 1 phase acts as the input for the next phase sequentially.
- \* Basic & most popular model.
- \* Execution flow is unidirectional.
- \* Waterfall development (S) model diagram.



? where we use waterfall model?

- \* project is short
- \* product definition is stable.
- \* Technology is understood.

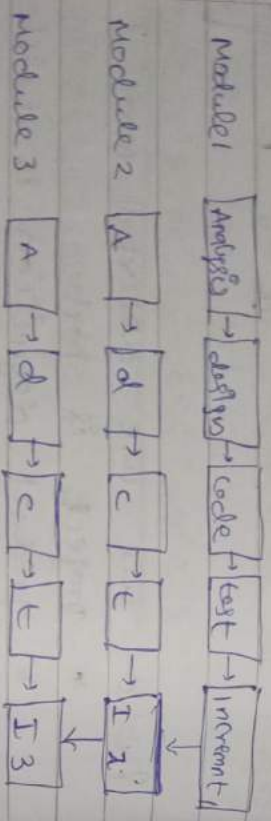
? Adv →

- \* simple & easy to understand
- \* minimize planning overhead
- \* 8th minimizes wasted effort
- \* Disadv →
- \* inflexible
- \* only final stage produce documentation
- \* Reverse is impossible.



## II Incremental Model =

- \* process of (S) development where requirements are broken down into multiple standalone modules of (S) development cycle.
- \* Each iteration passes through the requirements, design, coding & testing phases.



- \* Adv:
  - progress can be measured.
  - Easier to manage risk.
- \* Disadv:
  - Not suitable for smaller projects
  - More resources may be required.

## III Evolutionary Model =

- \* The main idea behind E-model is

SRS → (S) requirement specification.  
 collect various doc and common lang requirements.  
 continuously refine and perfect each other.

Conveyed by it's name.

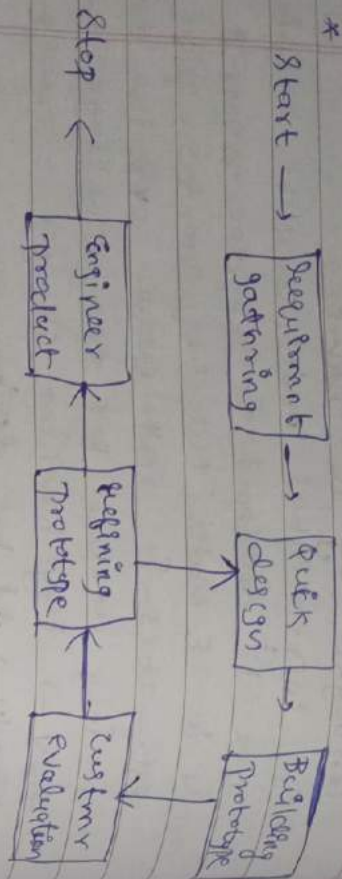
- \* In incremental model, complete requirement are 1<sup>st</sup> developed & SRS doc prepared.
- \* In E-model, the requirements, Plan, estimates & Soln evolve over the iterations.
- \* Mainly useful for ~~very~~ large products.
- \* Adv → Easy handling change requests.
- \* Disadv → Ad hoc design.
- \* 2 major E-process Model are -

### a) Prototyping Model = (no version and Demo requirements)

- \* It is also a popular life cycle model.
- \* P-model can be considered to be an extension of the waterfall model.
- \* This model suggests building a working prototype of the system, before the development of the actual (S)
- \* Prototype is developed based on the current known requirements.
- \* Adv/Strengths:
  - customer can see steady progress.
  - asked when requirements are changing rapidly.
  - know the how long it will take
  - There is no way to know the how of iterations will be required
- \* Disadv/Weakness:
  - Impossible to know how long it will take
  - There is no way to know the how of iterations will be required



Prototype :- cost - system (R) development  
 and done something, can (S) cost - no change  
 and is something and something (if any  
 change want, we change).



b) Spiral Model =

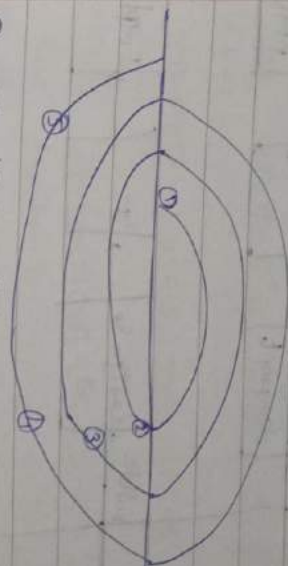
- \* Here, (R) is developed in a series of incremental releases.
- \* During early iterations, the incremental release might be a model.

\* Adv:

- Easy iterations of the project are cheap.
- Risk uses.
- All iterations meet the project needs.

Disadv:

- Complicated.
- Require knowledgeable management.
- \* This model got its name from the appearance of its diagrammatic representation that looks like a spiral pattern.



- 1 → project set
- 2 → Requirement analysis
- 3 → Design
- 4 → coding & testing
- 5 → Deliverables [doc that describes the testing process]

IV Iterative Model:

- \* Here, iterative process start with a simple implementation of a small set of (S) requirements.

\* After <sup>first</sup> version, if there is a need to change the (S), then a new version of the (S) is created with a new iteration.

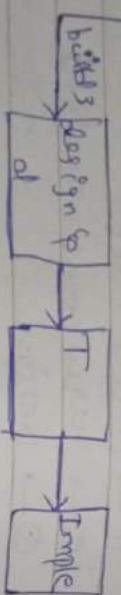
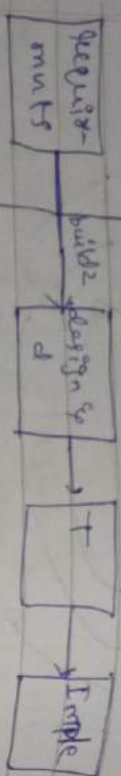
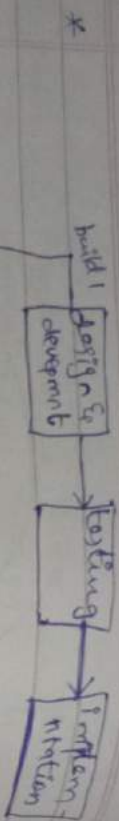
\* Adv:

- Testing & debugging is easy
- Requirements clarified & resolved

Disadv:

- not suitable for all projects.
- more resources required.





## → Specialized process Models =

### I Component Based development:

- \* provides a higher level of abstraction & divides the system into sub-problems, each associated with component partitions.
- \* The primary objective of this model is to ensure component reusability.

### II Formal Methods Model:

- \* This model is an approach to development S. eng that applies

mathematical methods to the process of developing complex systems.

\* 8 steps -

#### a) property-based specification =

- This step describes a main element in the system.
- These elements are the ops that can be performed on the system & relationships b/w the ops.
- eg:- consider a simple instant messaging app. for yr cell phn.

- same ops:
- start up
  - send msg
  - receive msg
  - display msg
  - shut down

#### b) Model-based Specification =

- This step describes the state of system can be in & how the ops can transition the system from state to state.
- eg as game as property based S.

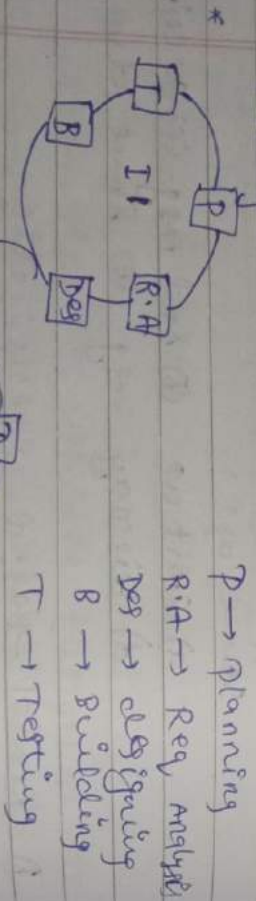
clicking the app icon to enter starting up. / press and btn to leave the ending msg.



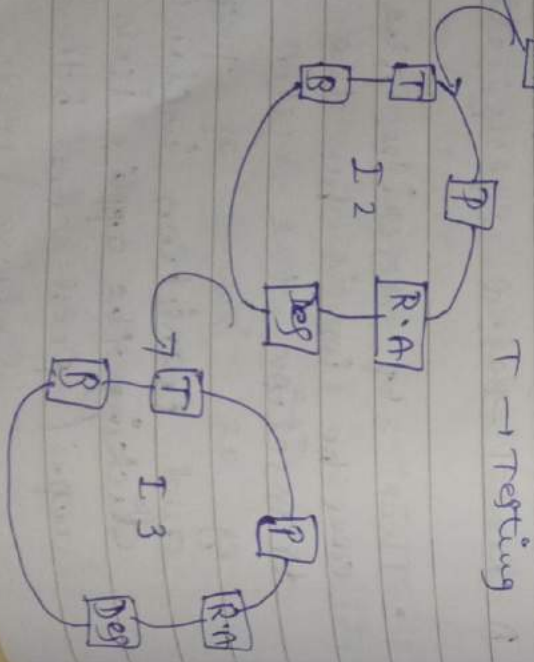
## ⇒ Agile Model :

- \* combination of iterative & incremental process models with focus on process adaptability & customer satisfaction by rapid delivery of working (S) product.
- \* It break the product into small incremental builds.
- \* Every iteration involves -

Planning, Requirement analysis & design, coding, unit testing & Acceptance testing.



Iteration  
1, 2, 3



- \* Agile process models -
- a) XP (extreme programming)
- b) Scrum
- c) Adaptive (S) development
- d) DSDM (Dynamic system development method)
- e) Crystal
- f) Feature driven development
- g) Agile modelling. (AM)

### I. Extreme programming =

↓ concept of ↓  
 \* commonly used Agile process model \* uses object oriented programming like planning, design, coding & testing

\* Follow are the values of XP :

a) Communication :

\* Building (S) development process needs communication developer & customer.  
 \* communication is imp for peer gathering & discussing the concept.

b) Simplicity :

\* Building (S) development process needs communication developer & customer.  
 \* communication is imp for peer gathering & discussing the concept.



Extreme values → 8 implications

- Communication
- Feedback
- Respect
- Courage

c) Feedback:

Feedback guides the development process in the right direction.

(d) Courage:

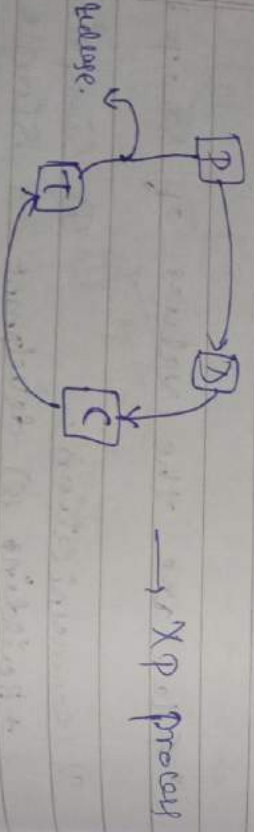
The courage (the discipline) to deal with it. Usually making the task easy.

(e) Respect:

Respect all team members, other stakeholders & cost.

\* Comprises 4 phases work activities:

Planning → Design → Coding → Testing



\* Adv

Have creates working cost (5) testers.

Disadv

Lack of decision makers

## II Scrum =

\* It is an agile (5) development method.

\* Includes the framework activities like Requirement, analysis, design, evolution & delivery.

\* work tasks occur within a process pattern in each framework activity → sprint.

\* Scrum consists of the use of a set of (5) process patterns —

(1) Backlog:

\* A prioritized list of project requirements/features that provide business value for the customer.

\* Items can be added to the backlog at anytime.

(2) Sprints:

\* consist of work units that are required to achieve a requirement defined in the backlog.

\* changes are not introduced during the sprints.

(3) Scrum meeting:

\* Short meetings are held daily by the scrum team.

\* Key questions are asked & answered by all team members.

(4) Demos:

\* deliver the (5) increment to the customer.



### III DSDM [Dynamic System Development Method]:

- \* It is an agile (S) development methodology.
- \* It is an iterative, incremental approach that is largely based on the rapid application development (RAD) methodology.
- \* Provides a 4-phase framework —
  - (1) Feasibility & business study
  - (2) Functional model
  - (3) Design & build iteration
  - (4) Implementation.

### IV FDD [Feature Driven Development]:

- \* It is an agile framework as it is organized around making progress on features.
  - \* It is designed to follow a 5-step development process, built largely around discrete feature projects.
- The project lifecycle looks like —
- 1) Develop an overall model
  - 2) Build a features list
  - 3) Plan by feature.

- 4) Design by feature
- 5) Build by feature.

### V LSD [Lean (S) Development]:

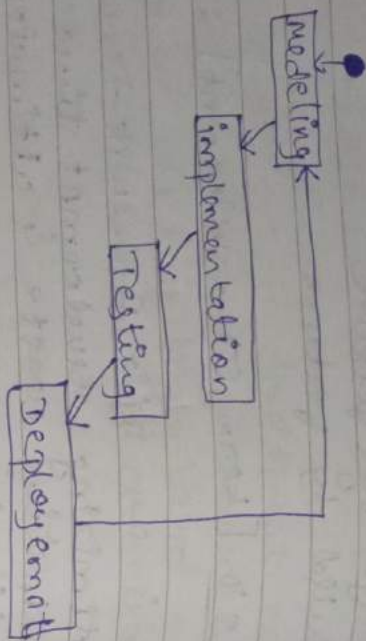
- \* It is an agile framework based on optimizing development time & resources, eliminating waste & ultimately delivering only what the product needs.
- \* LSD strengths:
 

Eliminates unnecessary activity & as a result can reduce costs.	LSD weaknesses: depends on strong documentation, & failure to do so can result in development mistakes.
---	---

### VI Agile unified process (AUP):

- \* AUP is the agile version of Rational unified process (RUP).
- \* It is an iterative — incremental process consisting of 4 sub-processes — Based on simplicity, agility, total dependence & it is known what they are doing.





- \* Modeling → to understand the business of the organization & to identify a viable (working) solution to address the probm domain.
- \* Implementation → to transform your models into executable code
- \* Testing → to perform an objective evaluation to ensure quality
- \* Deployment → to plan for the delivery of the system & execute the plan to end users.

## VII. Agile modelling (AM) :

- \* It is a practice-based methodology for effective modelling & documentation

of (S) based system.  
 \* It is a supplement to other agile methodologies such as -

- Extreme programming
- Select perspective
- Scrum

\* Core values of AM -

- courage - choosing AM requires courage.
- common - models exists for the purpose of communicating the people.
- feedback - just feedback on yr models
- Humility - Agile modellers hold their models loosely & will lay them down if required.

\* Adv :

Highly adaptive,  
 Constant customer  
 feedback

Disadv :

Difficult to make  
 additions within an  
 iterations.