

Course Title : SOFTWARE ENGINEERING

Course Code : 315323

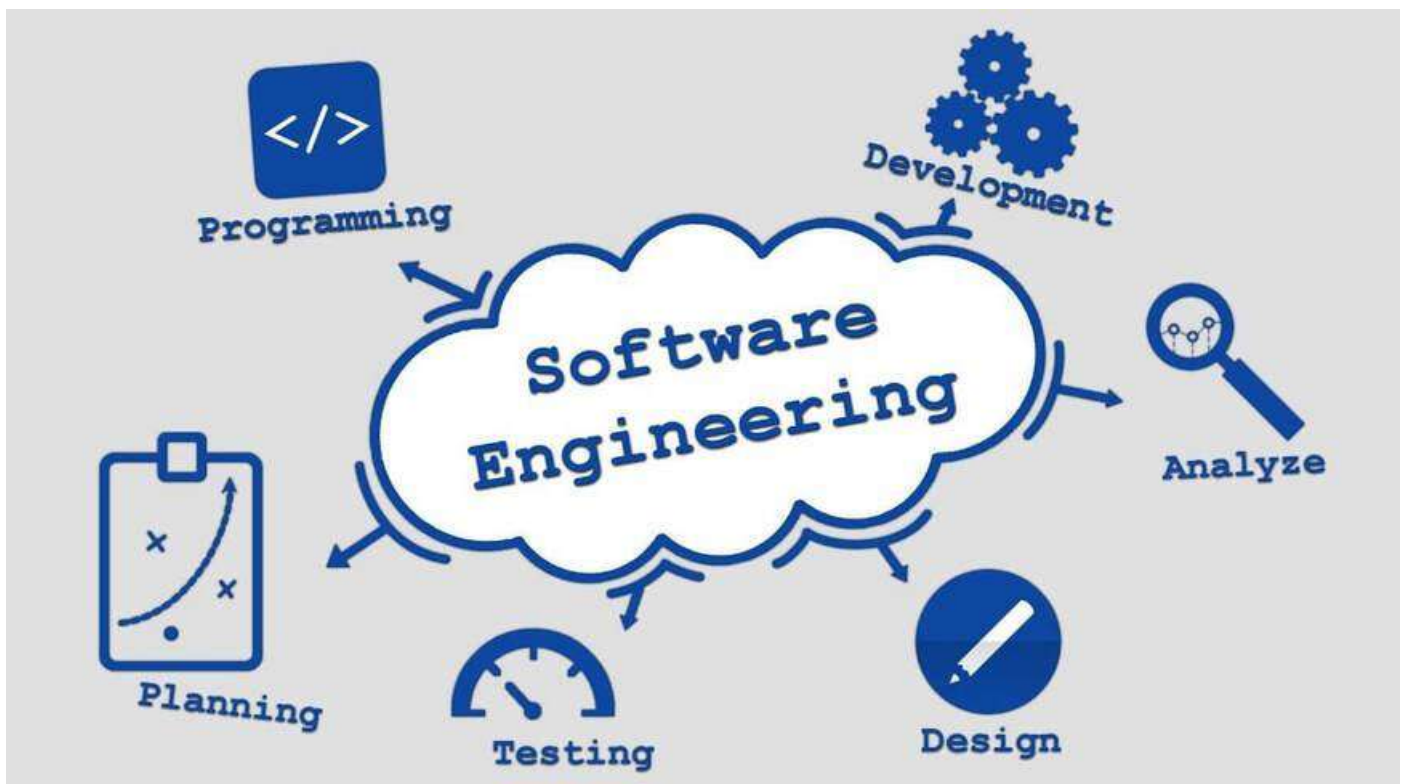
Programme Name : Computer Engineering

Semester : Fifth

Unit - I Software Development Process

Total Marks: 12

Prepared By: Mr.S.P.Chavan



unit - I Software Development Process.→ * Software CharacteristicsDefinition of Software :-

Software is ① instruction that when executed provide desire output, function and performance.

② Data structure that enable the program to adequately manipulate information and.

③ Documents that describe the operation and use of programs

[Software when made for specific requirement is called Software Product. Software is logical rather than a physical system element].

Defⁿ of Software Engg. :- is the application of a systematic, disciplined, quantifiable approach to the development, operation and maintenance of software

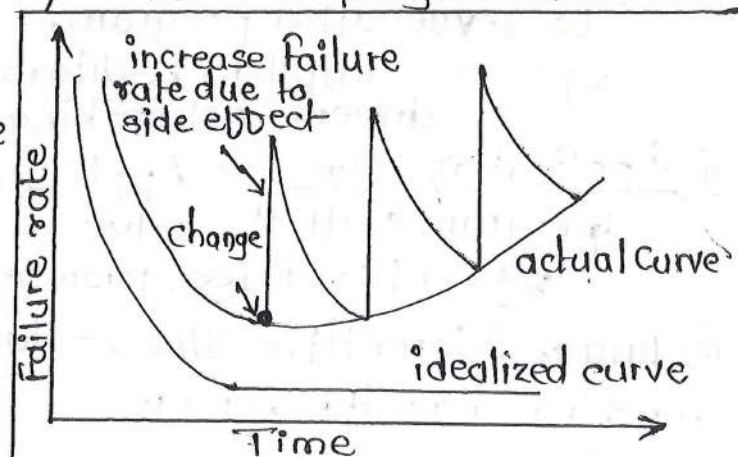
OR Software Engg. encompasses a process, a collection of methods and an array of tools that allow professionals to build high quality computer software.

Characteristics of Software :-

- i) Software is developed or engineered. It is not manufactured in the classical sense
- ii) Software doesn't "wear out". But it does deteriorate!
- iii) Although the industry is moving towards component based construction, most s/w continues to be custom built.
- iv) Software costs are concentrated in engineering. This means that software projects cannot be managed as if they were manufacturing projects
- v) A software component should be designed and implemented so that it can be reuse in many different programs.
- vi) Software is not susceptible to the environmental maladies that cause hardware to wear out

fig.

Failure Curves for s/w.



* Attributes of Software :-

- i) Functionality :- it refers to the degree of performance of the software against its intended purpose
eg. web browser → browses website, store bookmarks, manage history
- ii) Reliability :- It refers to the ability of the s/w to provide desired functionality under the given condition
eg. Database system → data is not lost, queries are executed correctly.
- iii) Usability :- it refers to the extent to which s/w can be used with ease and simple.
eg. mobile App.
- iv) Maintainability :- Software must evolve to meet changing needs
eg. update, fixed or extended over time
- v) Dependability :- Software must be trustworthy
- vi) Efficiency :- Software should not make wasteful use of system resources.
- vii) Acceptability :- Software should not make wasteful use of system resources.
- viii) Portability :- The ease with which s/w developers can transfer s/w from one platform to another, without changes
- ix) Integrity :- it refers to the degree to which unauthorized access to the s/w can be prevented.
- x) Robustness :- it refers to the degree to which the s/w can keep on functioning in spite of being provided with invalid data.

* Types of Software :-

- i) System Software → is a collection of programs written to serve other programs.
eg : → compilers, editors, operating system components, drivers, networking s/w.
- ii) Application s/w → Applⁿ s/w consists of standalone programs that solve a specific business need.
eg. MS-Office, Browsers
- iii) Engg. / Scientific s/w :- Formerly characterized by - number crunching algorithm, engg. & scientific s/w application.

eg. CAD, MATLAB, System Simulation etc.

④ Embedded S/w :- resides within a product or system and is used to implement & control features & function for the end user & for system itself.

eg. - keypad control for microwave oven, washing machine.

⑤ Product line s/w :- Designed to provide a specific capability for use by many different customers.

eg. word processing, computer graphics.

⑥ Web based s/w :- provides interface betⁿ user & internet.

eg. gmail, drive.

⑦ AI s/w :- application within this area includes robotics, pattern recognition, ANN., game playing.

⑧ Open source s/w :- source code which is available to all users.

eg. - Android OS, OpenOffice etc.

⑨ Programming S/w :- used to write, test, debug & develop other s/w programs

eg. Eclipse, Turbo C, VS.

⑩ Personal Comp (PC) S/w :- Used for both personal & official use.

eg. text editor, paint brush.

* Software Engg. A Layered approach.

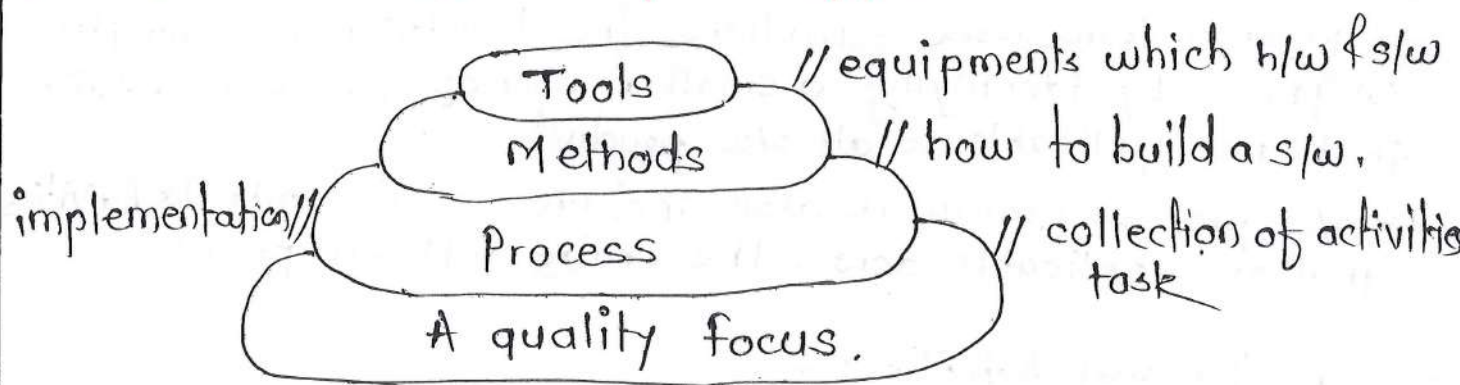


fig. S/w Process Layer.
Engg.

[To develop s/w one will have to go from one layer to another].

1) Quality focus → Base layer of s/w Engg.

→ focuses on maintaining the quality of s/w product.

→ Ensures the s/w is reliable, efficient, maintainable & user friendly.

eg. Error free code, good documentation, Testing at every stage.

2) Process : → This layered provides a Framework for s/w development

→ it defines how the s/w will be developed & maintain.

eg. Water fall, Agile, Spiral model.

3) Methods : → These are technical methods used to develop s/w.

→ Help in analysis, design, coding, testing & maintenance.

eg. DFD, UML, Structured programming.

4) Tools : → used to support & automate the SE, methods & processes.

→ Reduce manual work & increase productivity

eg. Visual Studio, Git hub, Selenium, CASE tools.

⊛ Software Development Framework : -

→ process framework establishes the foundation for a complete s/w process by identifying a small number of framework activities that are applicable to all s/w products

→ s/w process framework also include set of Umbrella Activities that are applicable across the entire software process.

→ Basic Framework Activities : -

1) communication : - This framework activity involves heavy comm & collaboration with customer and requirement gathering & other activities.

Umbrella Activities :-

- i) Software Project tracking & control :- it allows s/w team to access progress against project plan & takes necessary action to maintain schedule.
- ii) Risk management :- Assess risk that affect performance and quality of project.
- iii) Software quality assurance :- define & conduct activities to ensure s/w quality.
- iv) Formal technical Review :- uncover and remove errors before shifting to next level activity.
- v) Measurement :- Defines & collects process, projects & product measure that assist the team in delivering s/w that meets customer needs.
- vi) Software Configuration Management :- manage the effect of change throughout s/w process.
- vii) Work product preparation & production :- includes the activities required to create work products such as models, documents.

1.4. Software Process Model.

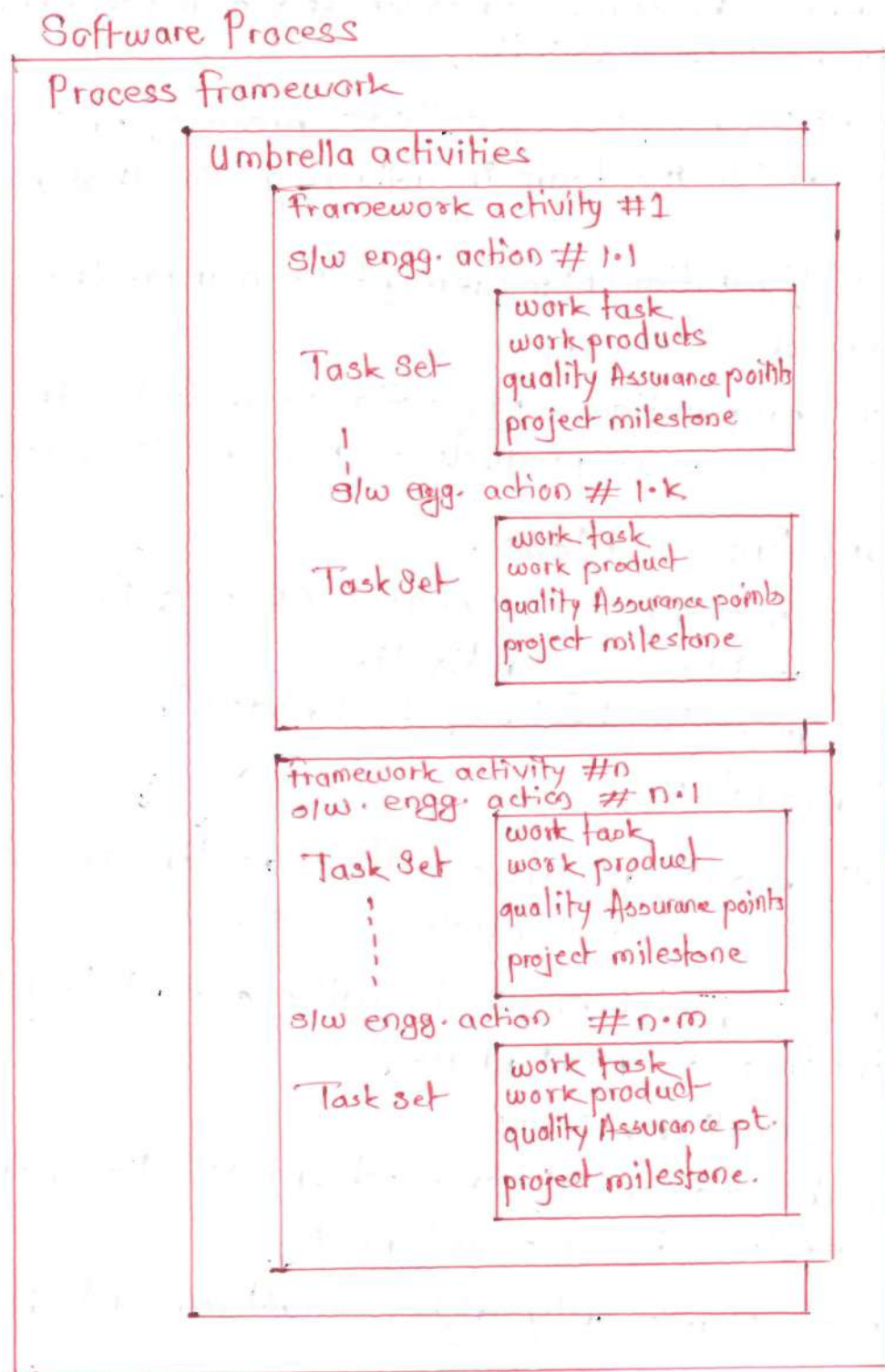
- it describe @ sequence of phases (R, D, C, T).
- (i) Flow of activities
 - (ii) role, tools & techniques.

⊛ Waterfall model :-

- Water fall model is also called as linear-sequential model or classic life cycle model.
- in this model, each phase is fully completed before the beginning of next phase.
- used for small projects
- feedback is taken after each phase to ensure that the project is on right path.
- Testing part starts only after development is complete.

- 2) planning :- plan for the sw Engg. work
 - describe technical task to be conducted.
 - Dead line is fixed.
- 3) Modeling :- creation of models for developer & customer to better understand sw requirement & design.
- 4) Construction :- code generation & testing.
- 5) Deployment :- final sw deliver to customer
 - feedback based on evaluation.

fig. Software Development (Process Framework)



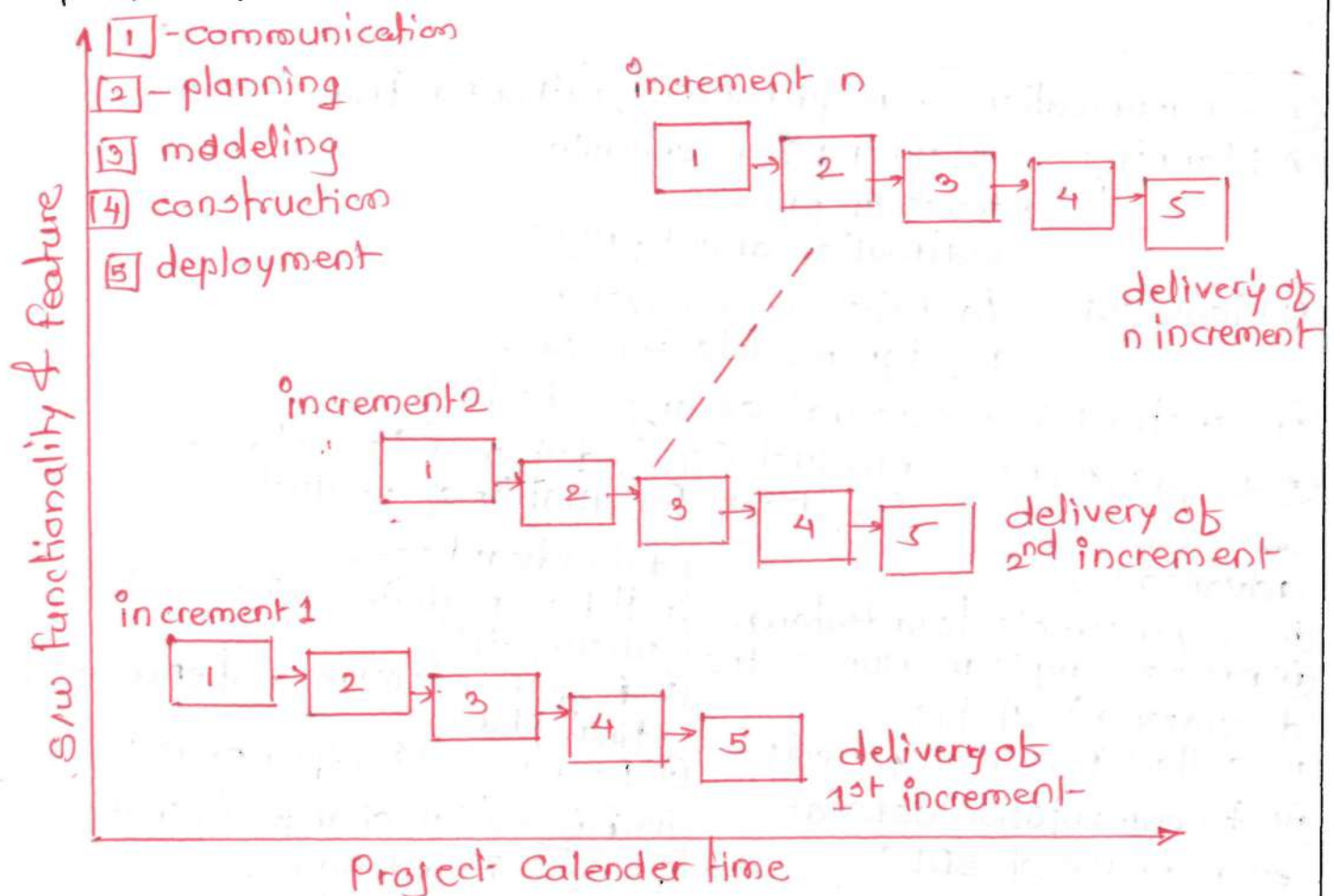
1.5 Incremental Process Model :- RAD Model.

① Incremental Model :-

- is also called as iterative enhancement model.
- in this model s/w is built in incremental fashion
- Project is divided into small subset called as increments
- combines elements of linear sequential model with iterative philosophy of prototyping.
- each linear sequence produces deliverable increments of the s/w.
- in this model the product is designed, implemented, integrated and tested as series of incrementals builds.
- each build consist of code pieces from various modules interacting to provide a specific functional capability.
- incremental development is useful when staffing is unavailable for a complete implementation.

eg. Word Processing S/w.

file management editing & document production — 1st
spelling and grammar checking — 2nd.
page layout — 3rd.



Commⁿ
project initiation,
requirement
gathering.

→ Planning
Estimating,
Scheduling
tracking.

→ Modeling
analysis
design

→ Construction
code
test

→ Deployment
support
feedback.

Fig. Water fall model.

- ① communication → requirement gathering from customer
- ② Planning → planning for schedule,
→ track on process
→ estimation about project
- ③ Modeling → Architecture of system
→ Design models for s/w.
- ④ Construction → actual coding & testing.
- ⑤ Deployment → product (s/w) delivered to customer.
→ feedback for quality of product.

advantages

- ① easy & simple to understand.
- ② provide simple measure of the development status.
- ③ suited for small projects
- ④ Documentation done at every phase of SDLC

Disadvantages.

- ① it has high amount of risk & uncertainty
- ② Require customer patience until final phase
- ③ Poor model for long projects
- ④ Difficult in change after the process is underway.

advantages :-

- generates working s/w quickly & easily.
- more flexible & less costly.
- easy to test & debug.
- customer can respond to each build.
- lowers initial delivery cost.
- easy to manage risk.

Disadvantages :-

- needs good planning & design
- needs clear & complete defn of whole system.
- Total cost is higher than than waterfall model.

* Waterfall model.

- 1) linear sequential model uses systematic processes through analysis, design, coding, testing, & support
- 2) Oldest model for development
- 3) sequential in nature
- 4) all phases are clearly defined
- 5) Real projects rarely follow sequential
- 6) difficult for customer to state all requirements.
- 7) working model is available only in latter part of development
- 8) Team size is large
- 9) Backtracking is difficult
- 10) diagram

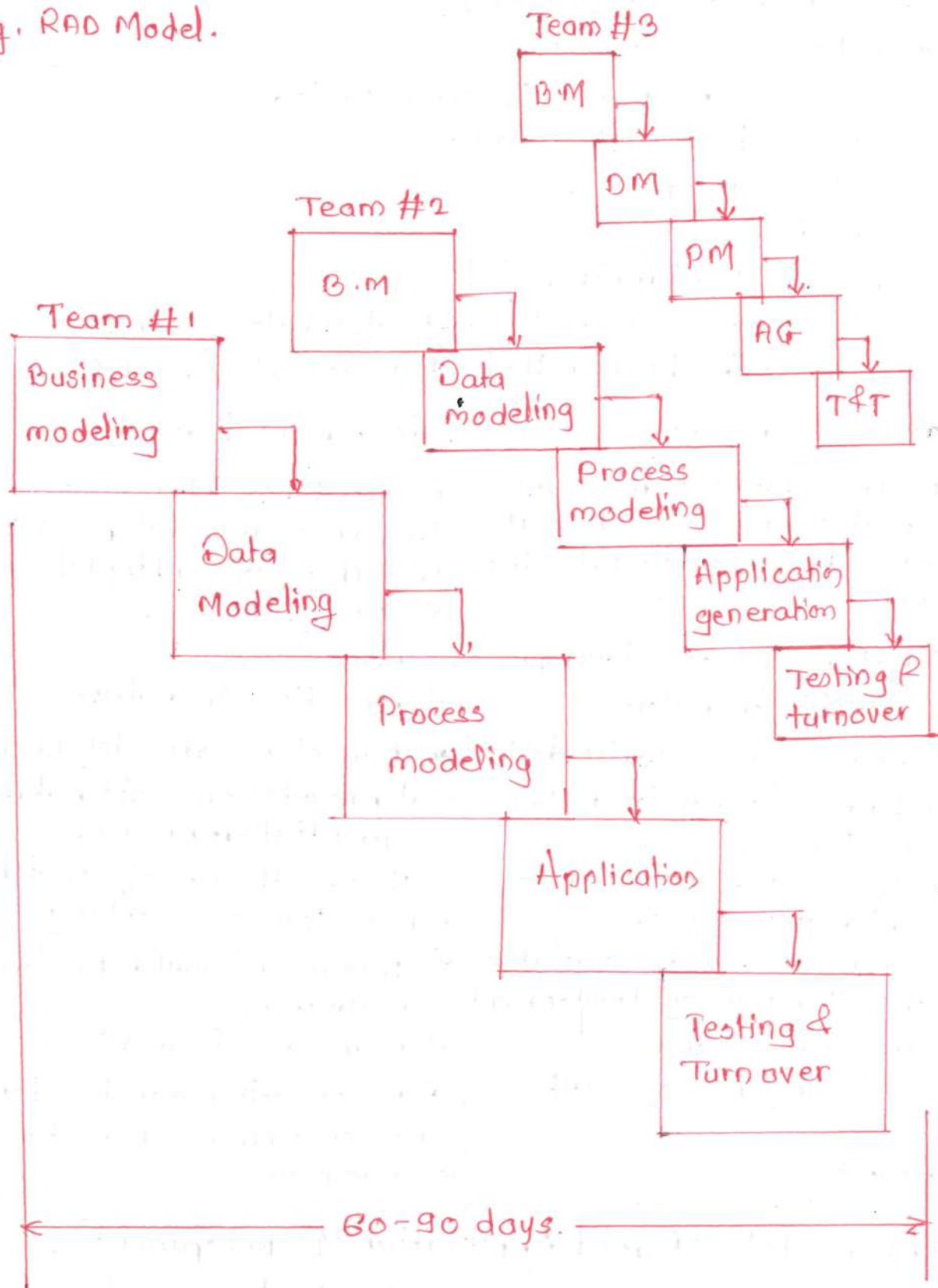
incremental model.

- 1) combines elements of the linear sequential model with the iterative philosophy of prototyping.
- 2) New model.
- 3) iterative in nature
- 4) few steps are defined.
- 5) it provides on rigid nature of sequential approach.
- 6) This method is of great help when low on staffing.
- 7) This model could be time consuming.
- 8) Team size is small.
- 9) Backtracking can be done for uncovered errors in next increment.
- 10) Diagram.

* RAD model (Rapid Application Development).

- is a modern s/w process model that depends on a short development cycle.
- it was first proposed by IBM in 1980's.
- is a high speed adaption of waterfall model.
- one of the distinct feature of RAD model is the possibility of cross life cycle activities which will be assigned to teams, team #1 to team #n.
- it distribute analysis, build & test phases into series of short.

Fig. RAD Model.



→ RAD model contains following phases.

1) Business Modeling :-

- Understand business functions and data flow.
- Defines scope and objectives.

2) Data Modeling :- Analyze and define data objects and relationships.

3) Process Modeling :-

- Define how data is processed and transformed.
- create workflow and process diagrams.

4) Application Generation :-

- Build prototypes using automation tools & reusable components
- iterative user feedback is incorporated.

5) Testing & Turnover :-

- minimal time for testing due to continuous integration.
- finalize & deploy the product.

Advantages :-

- i) fast delivery of functional prototypes.
- ii) High customer satisfaction due to continuous involvement.
- iii) Suitable for scalable component based systems.
- iv) easy to incorporate changes mid process.

Disadvantages :-

- i) not ideal for large, complex systems.
- ii) user commitment is crucial throughout.
- iii) Highly skilled developers needed with modeling skills.
- iv) it also doesn't work in high technical risk.
- v) user involvement throughout life cycle.

1.6 Evolutionary Process Model: Prototyping, Spiral model.

- [• overall design of system is not so good,
- occupy large memory space,
- delivery Date is not decide
- Quality of slw is poor
- Poor modularization.
- not suitable for complex product.]

These drawbacks are overcome in evolutionary process model.

⇒ main objectives of Evolutionary Process Model

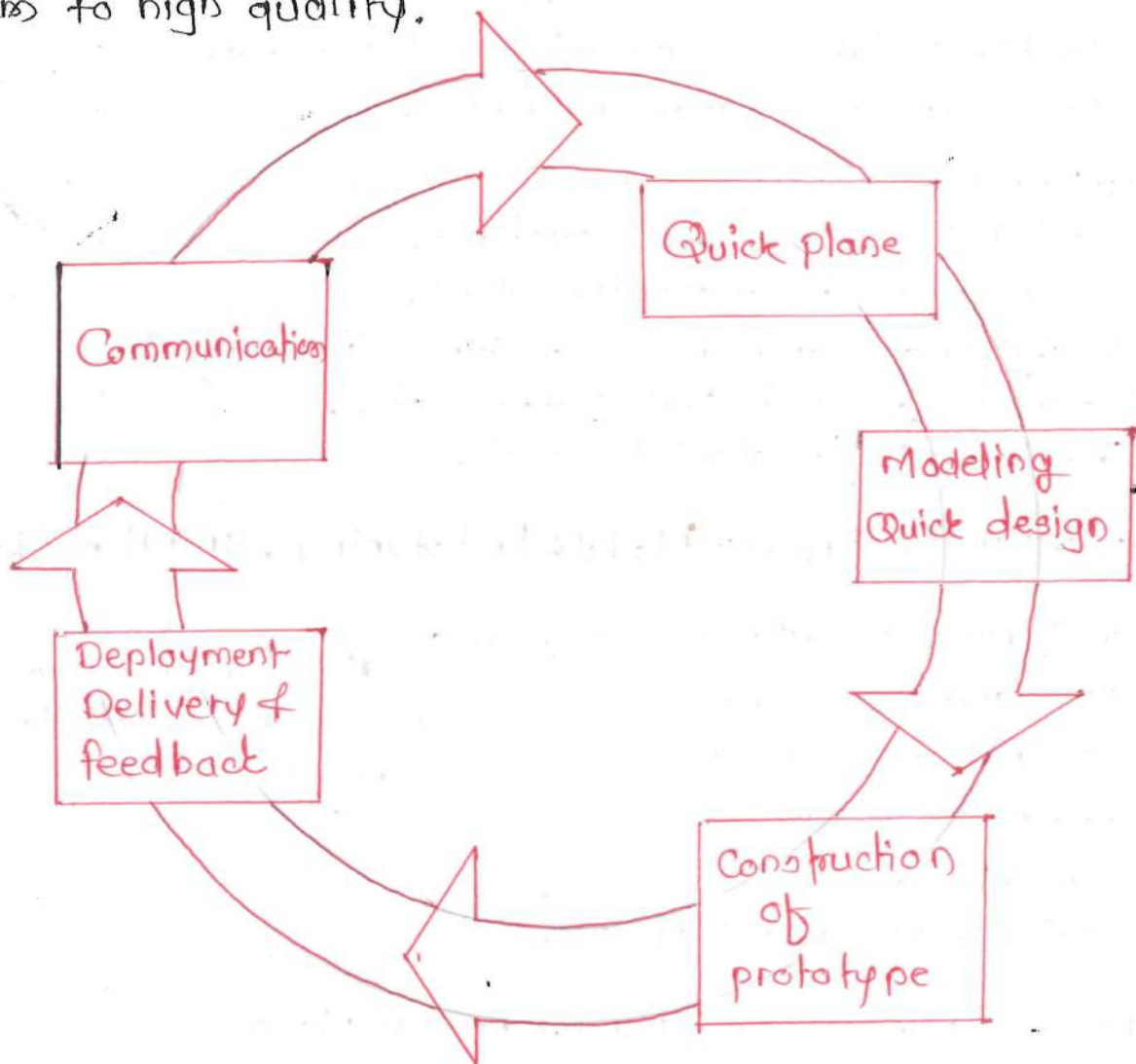
- ⇒ suitable for complex system
- it makes straight line path betⁿ customer requirement & business requirement & unrealistic product.
- it is iterative model.

① Types of EP Model -

- i) Prototyping Model
- ii) Spiral Model.

Prototyping Model :- ① follows an evolutionary and iterative approach.

- ② in this model initially requirement gathering is done.
- ③ Developer & customer define overall objective and identify areas needing more requirement gathering
- ④ quick design is prepared.
- ⑤ Design represents what will be visible to user in input & o/p format
- ⑥ From design prototype is prepared.
- ⑦ customer evaluates the prototype to refine the requirement initial prototype satisfy customer requirement.
- ⑧ when a working prototype is build, the developer uses existing program to through away the prototype & rebuild the systems to high quality.



- ① The sequence of event for prototyping are :-

- 1) Communication :- Begins with commⁿ by meeting with stakeholders to define the objective, identify requirements.
- 2) Quick Plan :- A quick plan for prototyping & modeling occurs.
- 3) Modeling Quick Design :- it focuses on a representation of aspects of system that will be visible to end users.

4) Construction of Prototype :- Design leads to the construction which will be deployed and evaluated.

5) Deployment, Delivery & Feedback :- User get actual system & Developer get feedback from customer.

Advantages :-

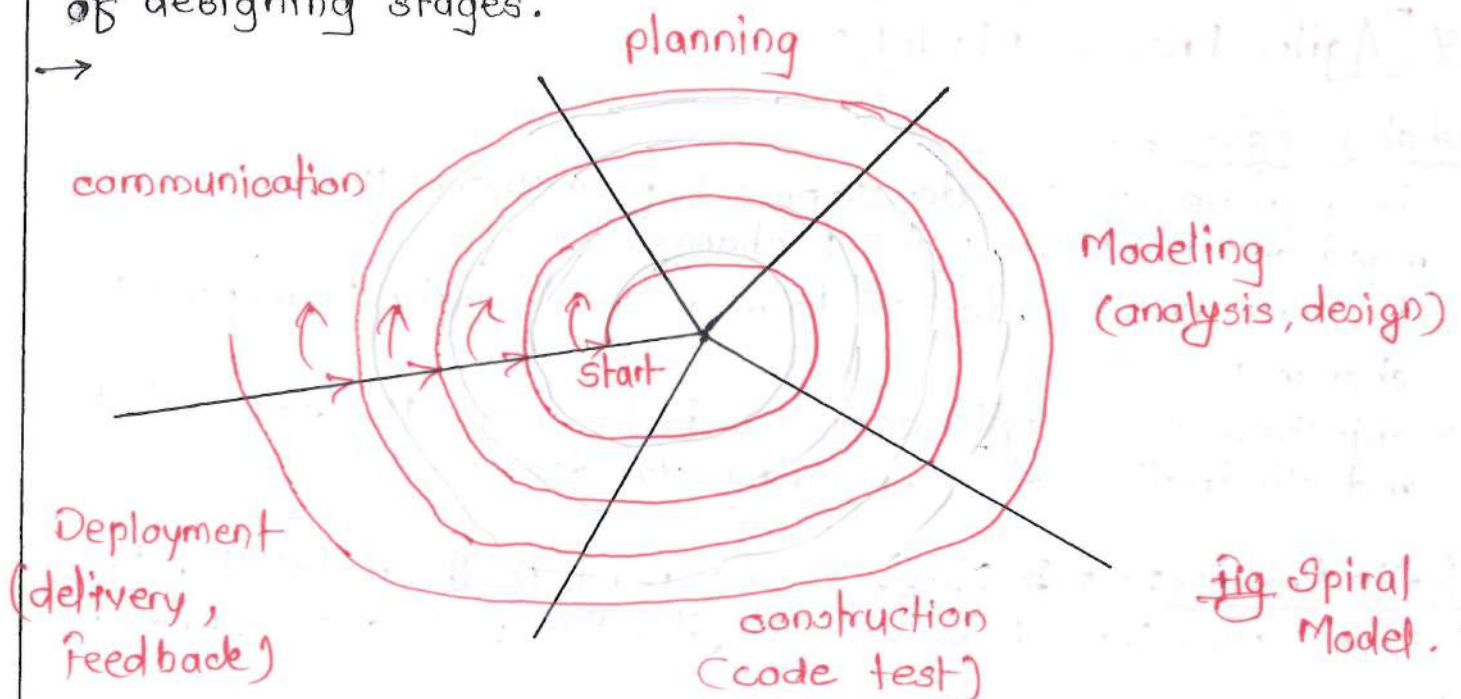
- i) Users are actively involved in development.
- ii) users get a better understanding of the system.
- iii) errors can be detected much earlier.
- iv) Quicker user feedback is available leading to better solutions.
- v) missing functionality can be identified easily.
- vi) difficult functions can be identified.

Disadvantages :-

- i) may increase the complexity
- ii) incomplete or inadequate problem analysis.
- iii) incomplete application may cause application not to be used as the full system was designed.
- iv) difficult to implementing & repairing way of system.

② Spiral Model :-

- model was invented by Dr. Barry Boehm in 1988. it follows an evolutionary approach.
- This model is able to produce rapid development in more complicated versions of software.
- Software development takes place in a series of developed releases
- In initial stage iterations, the release or model might be present on paper or may be a prototype model. These are kind of designing stages.



- In later stages, software system is actually produced.
- every framework activity represents one section of spiral path.
- As development process start, the sw team perform activities that are indirected by path around spiral model in clockwise direction.
- it begins at the center of spiral model & that is risk,
- project manager adjust the number of iteration/repetitions which are planned for completing the software.
- main feature of model is that it can be apply throughout the computer software's life.
- it continues & makes multiple ~~the~~ integrations till concept development get completed.

Advantages :-

- Risk Handling
- Good for large projects
- flexibility in requirements
- customer satisfaction
- it focuses on early error detection & design flaws.

Disadvantages .

- costly model to use,
- Not suitable for low risk Projects
- Risk analysis in this model requires highly specific expertise.
- Difficulty in time management.

1.7. Agile Process Model :-

What is Agile ? .

- is a group of sw development framework that encourage rapid and flexible response to change.
- term 'Agile' is adapted to mean 'the ability to respond to changes'.
- objective of an agile approach is to ensure the early detection and elimination of risks during sw development.

Agile Process : → it helps teams react to the instability to building sw. through incremental, iterative work cycles, known as sprints.

- Agile is s/w development methodology to build s/w incrementally using short iteration of 1 to 4 weeks so that development process is aligned with the changing business needs
- focus on the rapid development of the s/w product by considering current market requirements & time limits.
- Agile s/w development solves the problems of long time & heavy documentation processes.
- Agile s/w development saves man power, cost, documentation, & most importantly time.

Agile Principles :- (12 principles).

- i) Customer satisfaction through early & continuous delivery.
- ii) Welcome changing requirements, even late in development.
- iii) Deliver working s/w frequently.
- iv) Business people and developers must work together daily.
- v) Build projects around motivated individuals.
- vi) Face to face conversation is the best form of commⁿ.
- vii) Working s/w is the primary measure of progress.
- viii) Sustainable development.
- ix) Continuous attention to technical excellence & good design.
- x) Simplicity - the art of maximizing the amount of work ^{not done}.
- xi) Self organizing teams produce the best architecture & design
- xii) Regular reflection and adaptation.

★ Feature of the agile Software Development Approach

- ① Modularity :- is a key element of any good process. it allows a process to be broken into components called activities.
- ② Iterative :- Agile s/w processes acknowledge that we get things wrong before we get them right. Therefore, they focus on short cycles.
- ③ Time bound :- iteration become the perfect unit for planning the s/w development project. We can set time limits on each iteration & schedule them accordingly.

- ④ Parsimony :- Agile s/w processes focus on parsimony, i.e. they require a minimal number of activities necessary to mitigate risks & achieve their goals.
- ⑤ Adaptive :- During an iteration, new risk may be exposed which require some activities that were not planned. The agile process adapts the process to attack these new found risks.
- ⑥ Incremental :- an agile process does not try to build the entire system at once. Instead, it partitions the nontrivial system into increments which may be developed in parallel, at different times and at different rates.
- ⑦ Convergent :- Convergence states that we are actively attacking all of the risks. As a result, the system becomes closer to the reality.
- ⑧ People Oriented :- agile processes favor people over process and technology. They evolve through adaptation in an organic manner. Developer empowered raise their productivity, quality & performance.
- ⑨ Collaborative :- Agile processes foster commⁿ among team members.

④ Perspective process model	Agile Process Model.
1) Traditional approach.	1) Recent approach.
2) waterfall, incremental, RAD, prototype & spiral	2) uses concept of Extreme programming (XP).
3) product oriented process	3) people oriented process
4) it changes from analysis → Design → code → Test	4) it focus on all aspects of SDLC at any given time.
5) Detection & fixing during system & regression testing at the last phase of project	5) Early detection & fixing in each sprint followed by stabilization.
6) lesson learned from previous release implemented in next release	6) lesson learned from previous sprint will be implemented in the next sprint
7) no risk identification	7) early identification & mitigation in every sprints
8) Retrospection after every place.	8) after every sprint in Retrospection meeting.

(g) customer feedback at the end of project

(g) at the end of every sprint.

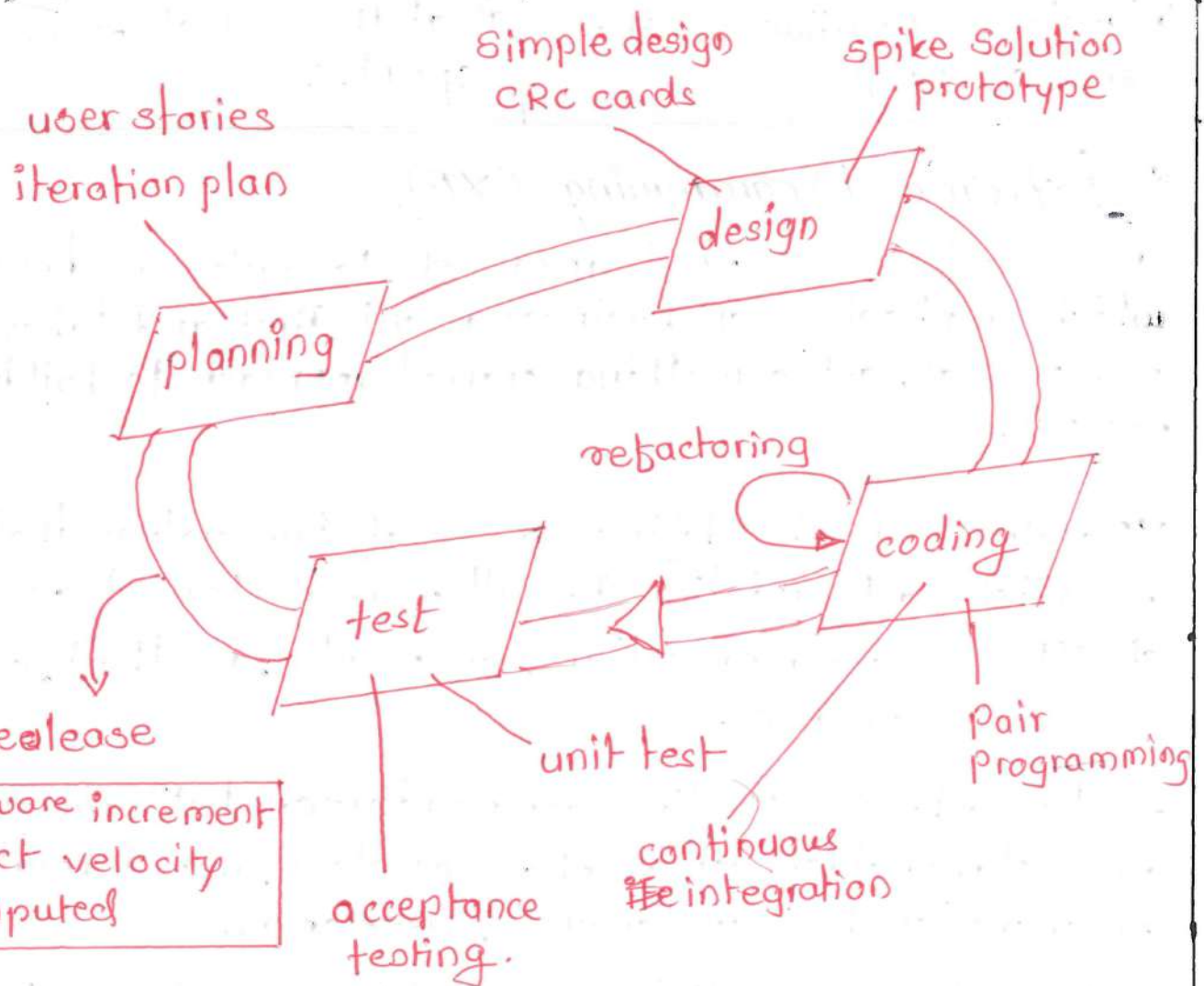
⑧ Extreme Programming (XP)

The most widely used approach to agile s/w development, which emphasizes on business result first and takes an incremental, get something started approach to building the product.

XP value - Beck defines a set of five values that establish a foundation for all work performed as part of XP — communication, simplicity, feedback, courage & respect.

- to achieve effective communication betⁿ s/w engg. & other stakeholder, XP emphasizes close, yet formal collaboration betⁿ customers & developers.
- To achieve simplicity, XP restricts developers to design only for immediate needs, rather than future needs.
- Feedback is derived from 3 sources
 - implemented s/w itself
 - the customer
 - other s/w team members.
- Courage (discipline) agile XP team must have the discipline to design for today, recognizing that future requirements may change
- Agile team must maintain respect among its members

⇒ The XP Process :- Extreme Programming uses an object oriented approach & encompasses a set of rules & practices that occur within context of four framework activities - planning, design, coding, & testing.



planning :- This activity begins with listening - a requirement gathering activity that enables the technical members of XP team to understand the business context for the s/w.

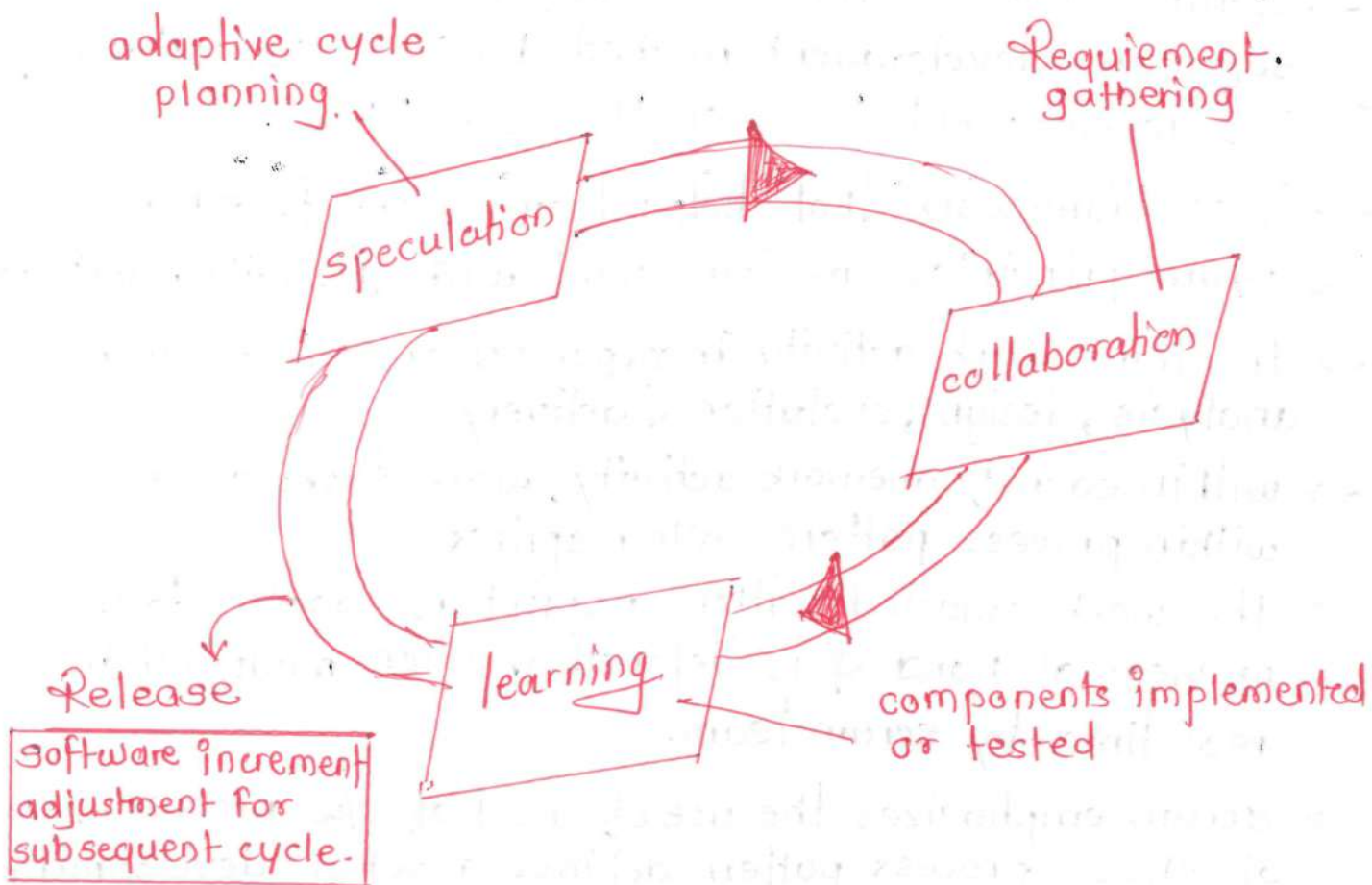
Design :- XP design follows the KIS (Keep it simple) principle. A simple design is always preferred over more complex representation.

Coding :- After Design, the team does not move to code, but rather develops a series of unit tests. Once code is complete, it can be unit tested immediately.

Testing :- The unit tests that are created should be implemented using a framework that enables them to be automated.
XP acceptance tests are also called customer test. They are specified by customer & focus on overall system features & functionality that are visible.

* Adaptive Software Development (ASD)

- has been proposed by Jim Highsmith as a technique for building complex s/w & systems.
- ASD focus on human collaboration & team self organization
- ASD life cycle consist of 3 phases
 - ① speculation ② collaboration ③ learning.



1) speculation →

The project is initiated & adaptive cycle planning is conducted.

→ Adaptive cycle planning uses project initiation infoⁿ the customer's mission statement, project constraints & basic requirements that will be required for the project.

2) Collaboration → is used by motivated people such a way that it multiplies their talent & creative.

→ collaboration encompasses communication & teamwork but it also emphasizes individualism.

③ learning :- ASD teams learn in 3 ways

- Focus groups
- technical reviews
- project postmortems.

④ Scrum :-

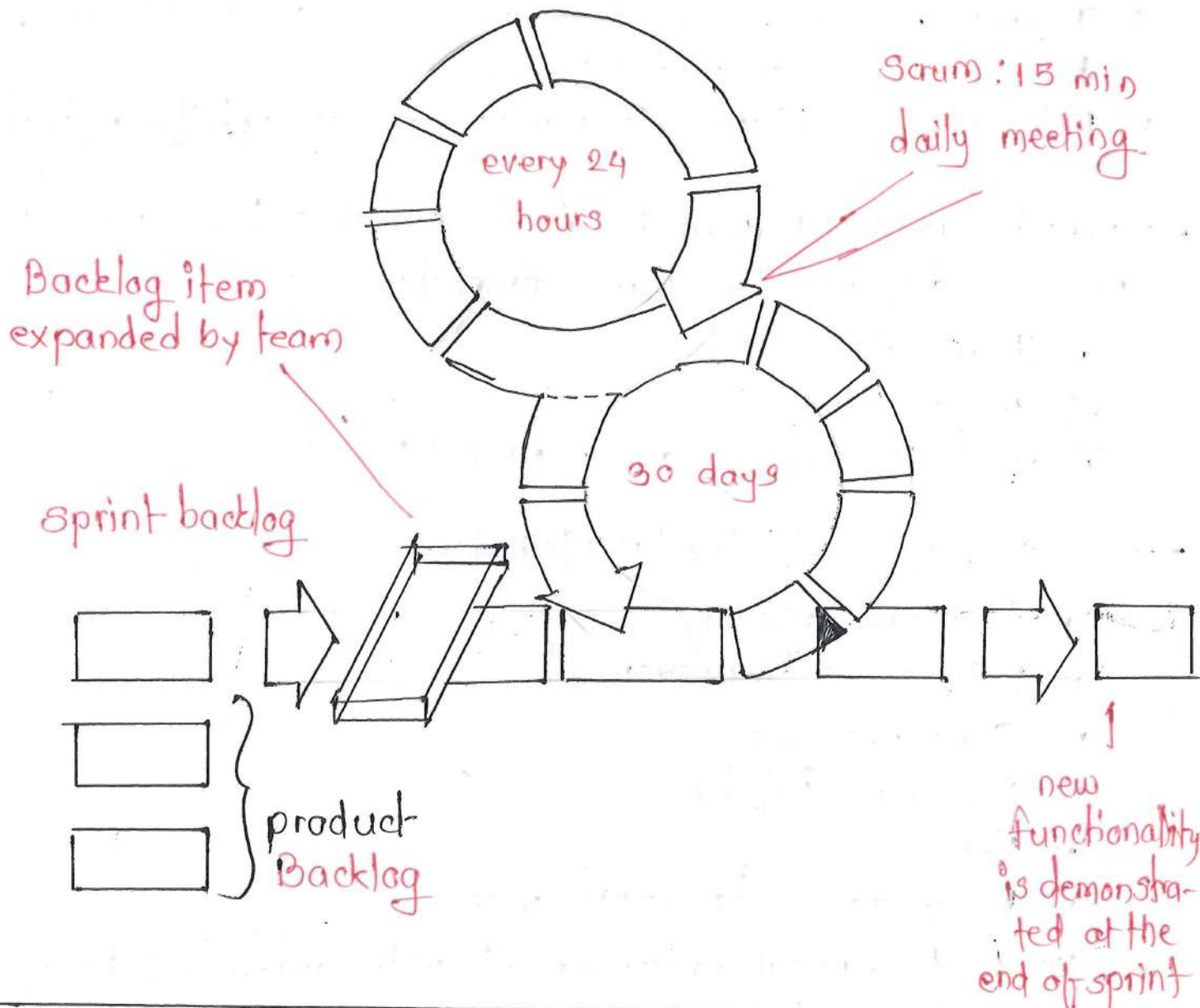
- scrum (name derived from rugby match) is an agile software development method by Jeff Sutherland & his development team in the early 1990's.
- it is framework that helps teams work together
- Scrum principles are consistent with the agile manifesto
- The framework activity incorporates are : requirements, analysis, design, evolution & delivery
- within each framework activity, work tasks occur within process pattern called sprints
- The work conducted within a sprint is adapted to the problem at hand & is defined & often modified in real time by scrum team.
- scrum emphasizes the use of a set of s/w process. Each of these process pattern defines a set of development actions :-
 - ① Backlog - a prioritized list of project requirements or features, items can be added to the backlog at any time. product Manager handles backlog & updates.
 - ② Sprints - consist of work units that must be fit into a predefined timebox (typically 30 days).
 - ③ Scrum meeting - are short (15 min) meetings held daily by the scrum team.
 - ④ - what did you do since the last team meeting?
 - what obstacles are you encountering?

- what do you plan to accomplish by the next team meeting?

a team leader called a Scrum master, leads the meeting & assesses responses from each person.

① Demos - deliver the software increment to the customer so that it can be demonstrated & evaluated by the customer

Scrum Flow Process



★ Crystal :-

→ in 1991, IBM asked Alistair Cockburn to develop the methodology for object-oriented projects.

So, he decided to interview project teams and find out their

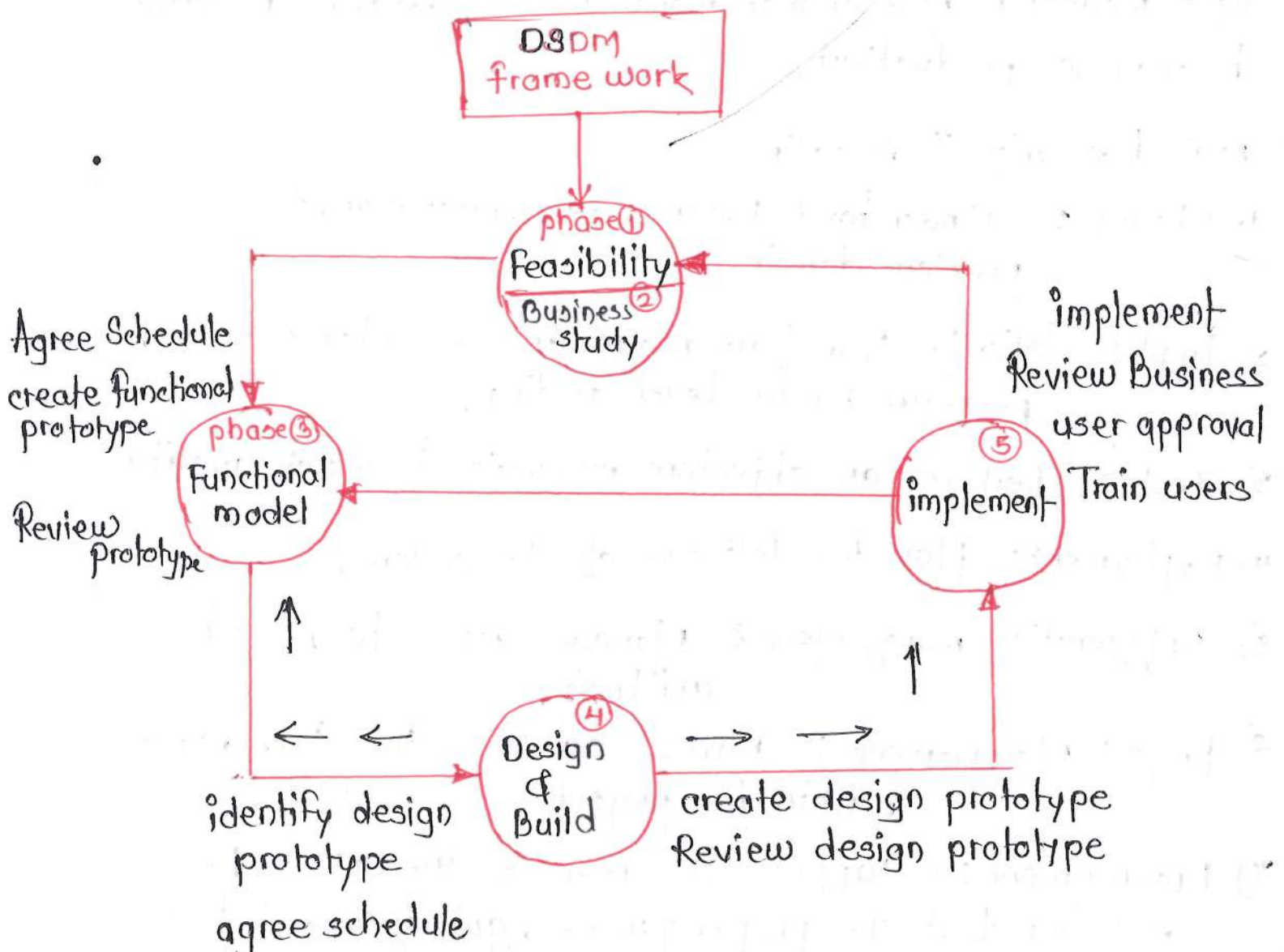
- After doing a research he came to conclusion that successful teams shared the same patterns & technique without even using any specific project methodology.
- finally, he used his findings to construct a family of methodologies & named it crystal.
- Crystal Method is an agile s/w development approach that focuses primarily on people & their interactions when working on a project rather than on processes & tools
- is based on two fundamental assumptions.
 - (a) Teams can streamline their processes as their work & become more optimised team.
 - (b) Projects are unique & dynamic & require specific method
- which approach will be most suitable for your projects depends on three dimension.
 - a) Team size
 - b) Criticality
 - c) What the priority of the project is.
- 7 properties of Crystal Method.
 - i) Frequent delivery
 - ii) Reflective improvement
 - iii) Communication
 - iv) Personal Safety
 - v) Focus
 - vi) Easy access to expert users
 - vii) A technical environment with automated tests, configuration, management & frequent integration.

① Dynamic System Development Method (DSDM)

- it is an agile project delivery framework, initially used as s/w development method.
- Released in 1994 to provide discipline to rapid RAD method.
- DSDM philosophy is borrowed from modified version of Pareto principles.
- DSDM life cycle defines 3 different cycles, preceded by 2 further life cycle activities

1) Feasibility Study - it establishes the essential business necessities and constraints

② Business study - it establishes the use and knowledge necessities.



③ Functional model iteration — it produce a collection of progressive prototype that demonstrate practically for the client.

④ Design & build iteration → it revisits prototypes design during functional model iteration

⑤ Implementation — it places the latest s/w investment into operational environment.

* Agile Unified Process :-

is simplified version of Rational Unified Process (RUP) developed by Scott Ambler

AUP applies agile technique including test driven development, agile modeling, agile change management & database refactoring to improve productivity.

AUP has only 7 disciplines.

1. Model :- Understand bussiness of organization, the problem domain

2. Implementation :- Transform model into executable code & perform basic level testing.

3. Test :- Perform an objective evaluation to ensure quality.

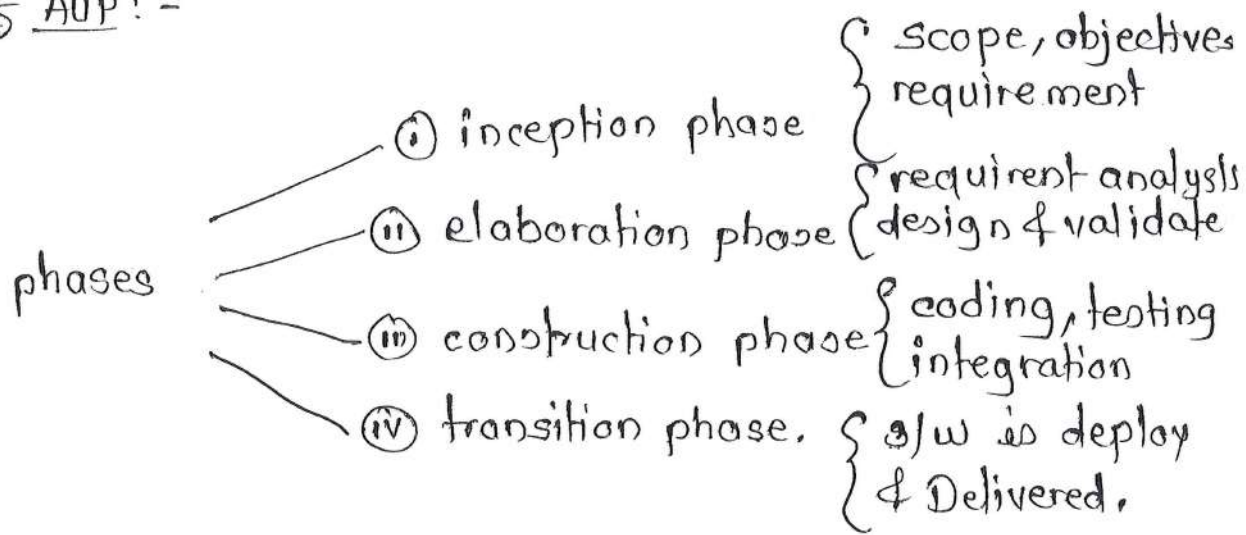
4) Deployment :- plan for delivery of the system.

5) Configuration management :- Manage access to project artifacts.

6) Project Management :- Direct activities that take place within the project

7) Environment :- Support the rest of the efforts by ensuring that the proper process, guidance and tools are available for team.

phases of AUP! -



Advantages

- i) Agile modeling → provide teams to create & maintain effective models of system.
- ii) Test-driven Development → it involve writing test cases before code
- iii) Continuous Integration :- involving integrating code changes frequently & automatically.
- iv) Refactoring :- @ involves improving design & code quality of s/w without changing its behavior.
- v) Iterative Development :- involves delivering working s/w in small, increment releases.

disadvantages :-

- i) Lack of Documentation
- ii) Limited Scalability.
- iii) Dependency on team Experience
- iv) Lack of formal Process.

AN
4/10
05/05/25

Question Bank

Unit-I Software Development Process

2 Marks Questions

1. Enlist and explain software characteristics (any two).
 2. List any four types of software.
 3. Enlist and explain any two characteristics of software.
 4. Define Software Engineering.
 5. Enlist any four attributes of a good software.
-

4 Marks Questions

1. State and explain with examples four broad categories of software.
2. State and describe characteristics of Software Engineering.
3. Explain Software Engineering as layered technology approach.
4. Explain Process Framework with suitable diagram.
5. Explain Adaptive Software Development (ASD) with respect to:
(i) Focus of ASD (ii) Steps in ASD.
6. Distinguish between Prescriptive Process Model and Agile Process Model.
7. Describe any four characteristics of Agile Process.
8. Describe phases in Spiral Model. Draw a neat labelled diagram showing: Customer Communication, Planning, Risk Analysis, Engineering, Construction & Release, and Customer Evaluation.
9. Explain RAD Model with respect to: (i) Diagram (ii) Phases of RAD Model (iii) When to use RAD Model.
10. Explain Dynamic Systems Development Method (DSDM).

