

## Chapter - 3

### Requirements Engineering

RE

RE is considered as 1<sup>st</sup> stage in development process of any product.

It is a process of finding requirements, documenting analyzing ~~the~~ services & constraints. on sys.

These requirements tell about the customer needs or functionalities / Services / features. of the system.

Requirements can be capture in two ways

1. User Requirements  
High level abstract stmt.  
of services, can be  
constraint on system.  
written in Natural  
language stmt.

2. System Requirements  
Detailed Stmt. of Services or  
constraints of the system.

→ Provides understanding of the  
domain in development of sys.

Q) System requirements are written in natural lang.  
False.

## # Mental health care patient management service (MHC - PMS)

### ◦ User- & System Requirement -

- Service → generate a report
- User Requirement → generate report of the drugs & cost prescribed by clinic.

System → 1. generate report of drug & costs at the end of month (last working day)

2. Admin generate report at last working day at 17:30 pm

3. Drug informa" should contain about 10mg, 20mg  
(patient info.) No.of frequenc  
& brand,

4 Only authorised users can access.

## # Types Of System Requirements -

### Requirement

#### Functional

what the system do?

#### Non-functional

how the system should behave?

I/P → **System** → O/P

e.g Banking System

User id →

**Banking System**

→ Open a loan account

→ duplicate  
loan acc

Check Balance

→ Balance  
should be  
displayed  
2 ms  
(constraint)

# Requirements

## Functional

(FR)

what the system do?

? IP → **System** → OP

e.g.: Banking

User → **Banking System** → Open a loan account → Duplicate loan account  
id  
Check the balance → Balance should be displayed within 2ms.

**Product feature** ← → **Product Properties**

**User Requirements** ← → **User expectation**

e.g. send ~~the~~ email when new user signs up. → within 1 hr.

ex -

① for an everyday object like a cup would be: "ability to contain tea or coffee without leaking"

① contain hot liquid without leaking upto more than 48°C

② A system loads a webpage when user clicks on a button

② A system should load a page when user clicks on a button within 2ms.

FR - shall or present tense  
past, future

NFR - must / should

## # Functional Requirements -

- \* Statement of services the system should provide, how the system should react to particular I/p & how the system should behave in particular situation.
- \* may state when the system should not do.

- eg:-
1. User shall be able to order an item.
  2. User wants to check cart for item details.
  3. Only admin updates users & item database.
  4. A sales system allow to record customer sales.
  5. System display bill for the items ordered.

## # MHC - PMS: functional Requirements

- o A user shall be able to search the appointments list for all clinics.
- o The system shall generate each day for each clinic, a list of patients who are expected to attend appointments that day.
- o Each staff member using the system shall be uniquely identified by his/her 8-digit employee number.

Requirements: completeness & consistency  
→ difficult to achieve

Completeness - means include detailed descrip<sup>n</sup> of all functionalities.

consistency → No conflicts or contradictions or ambiguity in the interpretation.

- Req. 1. only admin has access to employee database.  
2. Employees can change the salary details.

## # Non-Functional Requirements -

constraint on services / functional

- Quality attribute

- Applied on whole system not on individual feature
- Helps in Design Architecture
- Reflect system properties such as Reliability, Availability, Security & Safety
- IDE, process & prog. language.

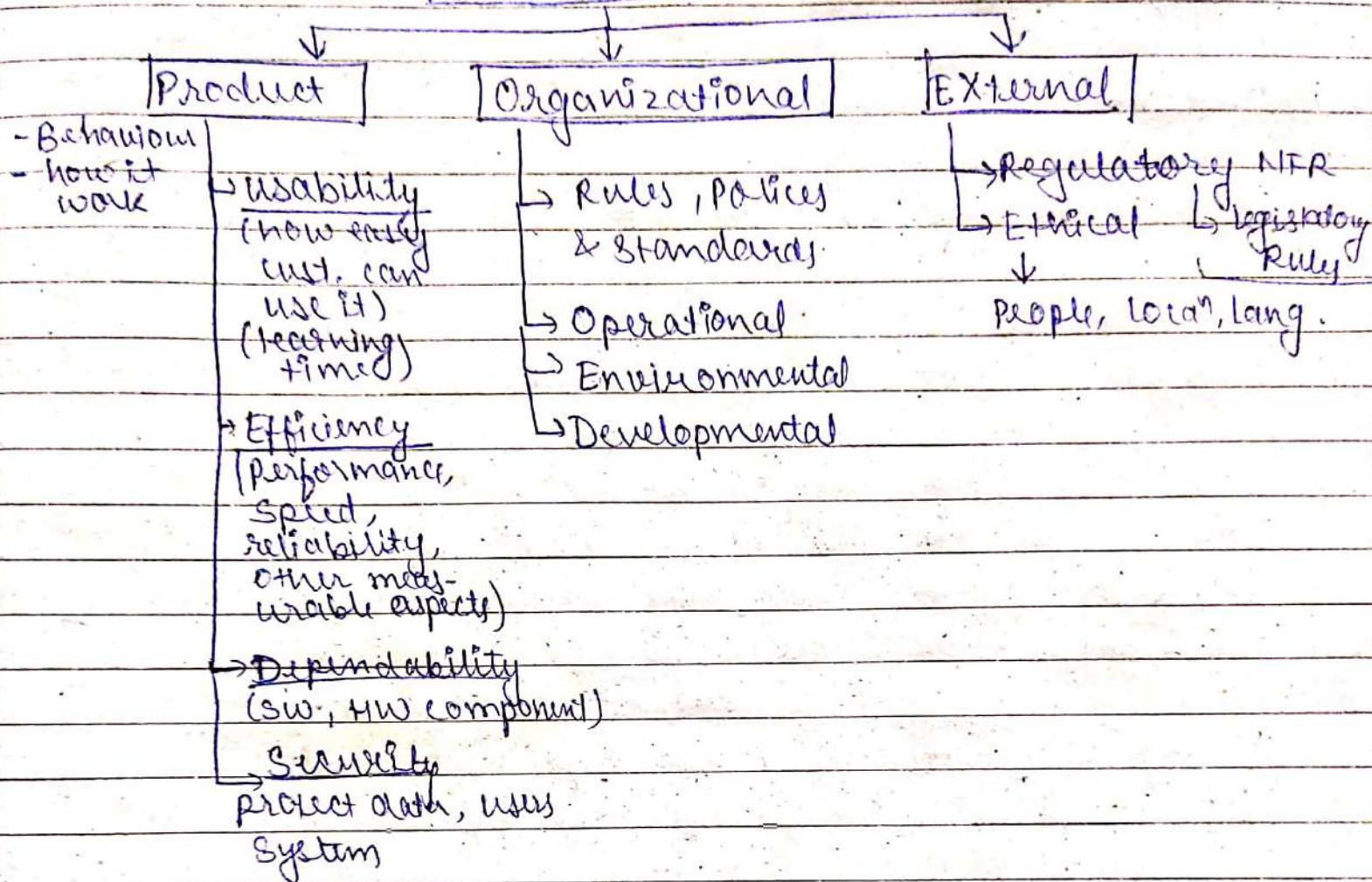
✓ These requirements should be measurable/quantifiable  
→ Avoid using words such as good, better, best, all, some, few, many. (adjective parameters)  
should

✓ We use should/must in NFR. (Shall can also be used)

1. User should be authenticated while Login.
2. System should be portable.
3. System should handle nearly 2K user without affecting performance.
4. User should change password after 1<sup>st</sup> login.
5. Privacy of info. such as export of technology, intellectual properties, document should be edited once in a year.

## # Types Of Non- Functional Requirements -

Non-Functional  
req.



Example: —

1. ATM system should be available b/w 5 am to 17:30 pm  
(availability)
2. User is authenticated using PIN provided by BANK  
(Security) ↓
3. User should be allowed to login only thrice.  
Acc. block after 3 trials.
4. Downtime of system should not be more than 5 sec. in a day. (Reliability / Availability)
5. User should be able to international banking without assistance (help) (usability).

As NFR affect the architecture of the system therefore, these NFR should be written at the early stage in the Requirement Phase Only.

## # Metrics -

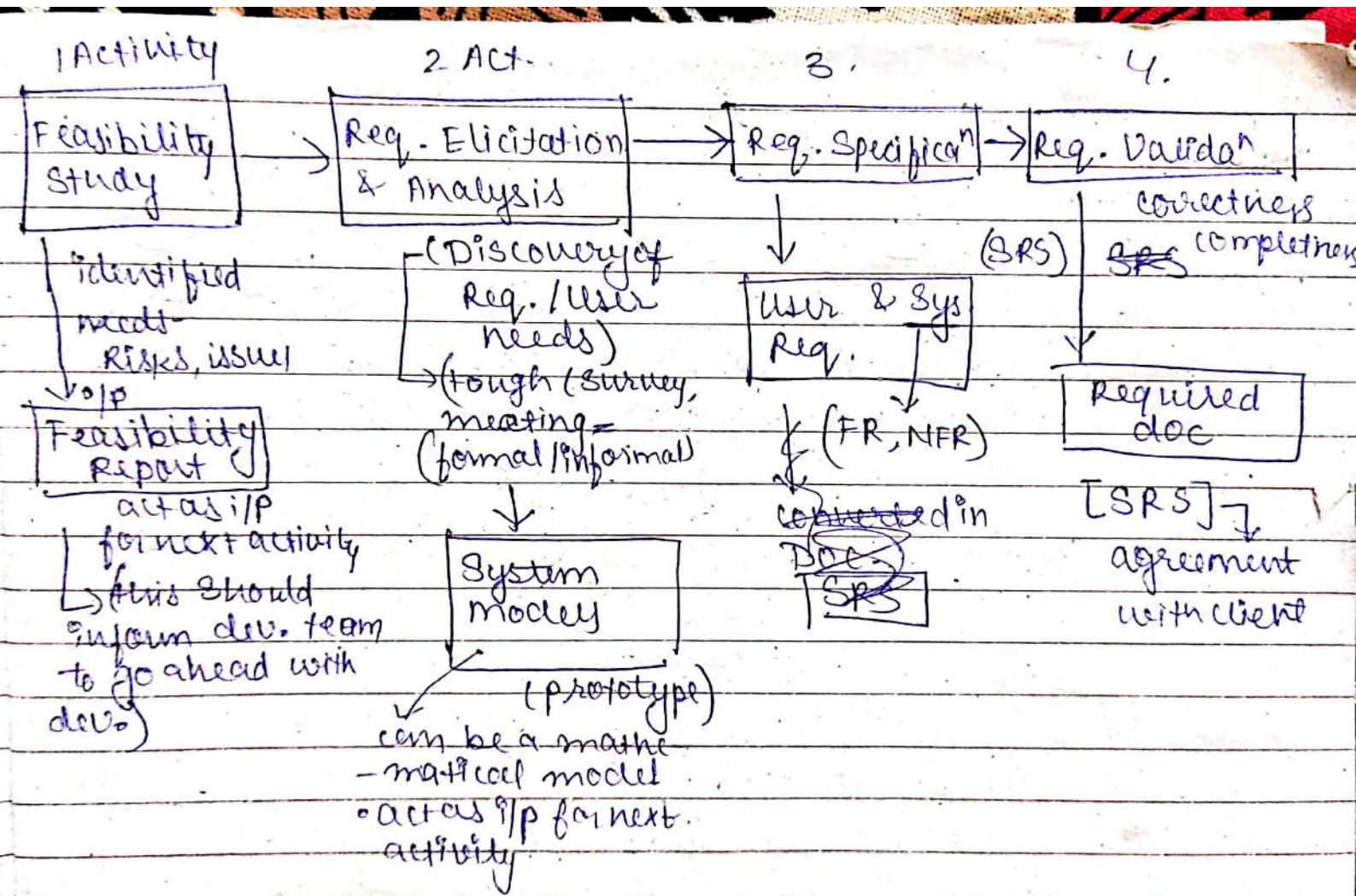
→ Metrics for specifying ~~NFR~~ NFR

Property	Measure
Speed	Processed transaction / sec, user / event response time.
Size	Mbyte / No. of ROM chips (capacity)
Ease of use	Training Time / No. of help frames
Reliability	Mean time to failure, PE of availability
Robustness	Time to restart after failure % of event causing failure.
Portability	% of target dependent stmt. No. of target systems.

## # Requirement Eng. Process -

↳ how & req. gathered, analysis is done & written in standard formal & validation is done.

There are 4 Activity -



## Chapter 6

SCRUM:- It is a type of Agile Framework.

- Scrum is an agile process most commonly used for product development, especially software development.
- In Scrum, projects move forward via a series of iterations called sprints.
- Each sprint is typically 2 to 4 weeks long.
- At the start of each sprint, the team selects some amount of work from the product backlog & commits to completing that work during the sprint.
- At the end of each sprint, the team produces a potentially shippable product increment i.e. weekly, high-quality software.
- It is a framework within which people can address complex adaptive problems while productivity & creativity of delivering product is at highest possible values.

## \* Salient features of Scrum :-

- Scrum is lightweighted framework.
- Scrum emphasizes self-organization.
- Scrum is simple to understand.
- Scrum fast framework help the team to work together.

Input from  
Customers Team

↓  
Product Owner | Team



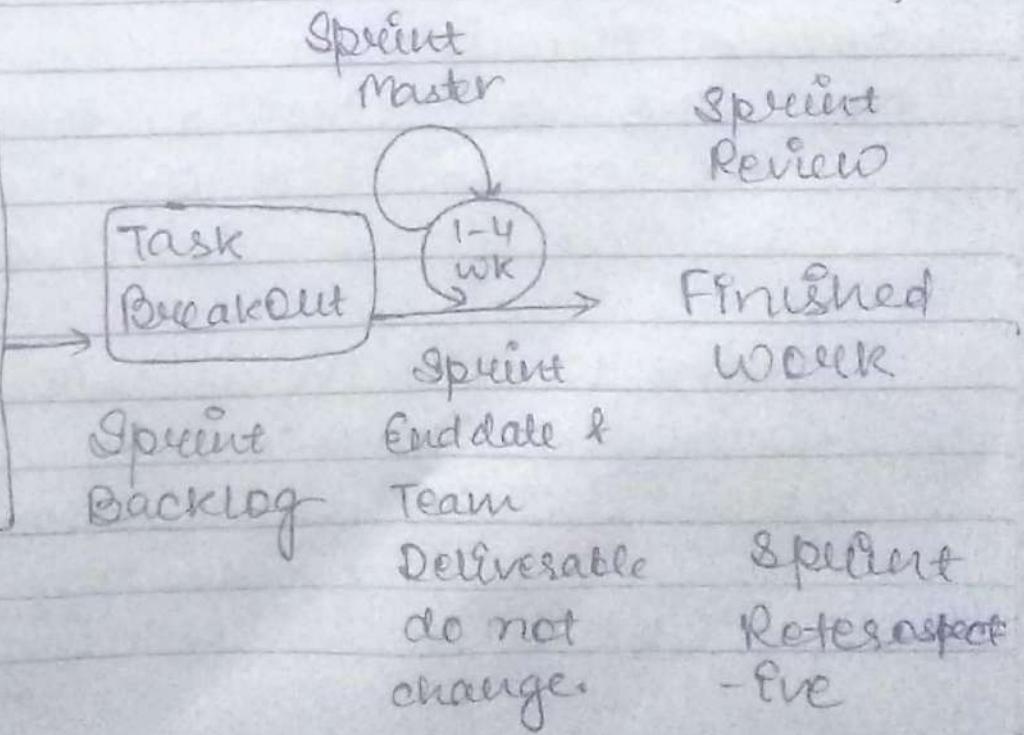
Prioritized list of  
what is required:  
features, bugs

Product  
Backlog



Team Selects  
Starting at top  
as much as it  
can commit to  
deliver by end  
of sprint

Sprint  
Planning  
Meeting



- 1). Apply Scrum for Chat Application.  
Draw the figure then write the  
For the development of chat application,  
let's suppose the user requirements are:
- 1) • The application should be able  
to verify the end users who  
are using the chat application.
  - 2) • The application should be able  
to ~~enect~~ maintain the  
privacy of the end users.
  - 3) • The application should be able  
to secure the msg being sent.
  - 4) • The msg content should be  
properly encoded & decoded to  
maintain the security.

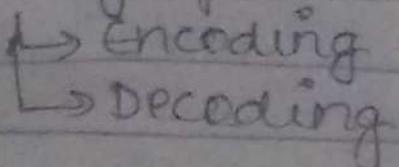
#### \* Sprint Planning :-

Now, the development team will plan  
the sprint out of the requirements  
gathered from customers:

Sprint 1: User Verification

Sprint 2: Privacy

Sprint 3: Security of content



### \* Sprint Backlog:-

Now, the development team assigns the end date & duration for each sprint.

Sprint 1 : 03 Weeks

Sprint 2 : 02 Weeks

Sprint 3 : 04 Weeks

Now, the team deliverables should do not change & Finished work is augmented together once all sprints are over.

In bet<sup>n</sup> each sprints, daily stand up meeting & sprint reviews are carried out.

## CHAPTER-2

### AGILE SOFTWARE DEVELOPMENT

- It allows changes during the development of software and allows quick delivery of changes.

Quiz: Agile software development allows changes during development. True

#### \* Topics:

- Agile Methods:
  - Agile Software Engineering principles
  - Agile Manifesto
  - Characteristics of Agile Software.
- Plan-driven & agile development:
  - Difference b/w PDA & ASD.
  - Applicability.
- Extreme programming : (XP)
  - Life Cycle Activities
  - Key practices

#### \* Agile Software Development (ASD):

- 'Agile' means adaptiveness or reacting to change.

- ASD means we are adjusting to an environment, identifying uncertainty & then figure out how to adopt it in the software development.

- There can be two cases, environment is changing or customer needs can be changing.
- Thus, Agile Software is defined as set of frameworks, values/best practices based on Agile Manifesto.
  - ↳ doc produced by Industrialists that provides best practices based on the problem definition.

• Why is ASD used? (traditional)  
 There were many issues in the classical models, to overcome those ASD are used.  
 Issues are :

- 1). Rigid Documentation of SRS, SDD
- 2). Longer duration of the project completion.
- 3). Rework (possibility of the rework).
- 4). Not availability of working software.
- 5). No feedback from customer till end.

### \* Agile Manifesto :

↳ doc that includes set of values/best practices that are to be considered.

4 values :

- 1). Individuals & Interactions over process & tools.
- 2). Working Software over comprehensive documents.
- 3). Customer collaboration over contract negotiation.
- 4). Responding to change over following a plan

## ① Individuals & Interactions over process & tools & traditional approach

- In order to satisfy the need of customer, there must be interaction b/w the product developer & the customer.
- So, ASD uses a lot of communication to help the product make more effective, using different techniques E.g: informal meetings, collective surveys etc.

## ② Working Software over comprehensive documents

- If the customer can use the product in intermediate stage, it helps the development team to get the feedback. So, instead of planning everything in the beginning, ~~ASD~~ it tells us to execute the current iteration then move to the next stage or else there may be a chance of wastage of plan.

## ③ Customer collaboration over contract negotiation

- Customer's R/p is considered as a very imp. part in the development of the product.  
So, consider it for every activity whenever required.

## ④ Responding to change over following a plan

- One ~~is~~ instead of sticking to plan decided in the early stage, go for frequent plans.

\* So, ASD mainly focus on coding rather than design (Limited).

(i) Quick delivery

(ii) Test first approach → write the test cases or test plans before development.

Video 2:

\* Example:

• Company ABC is designing web browser for recent release of OS.

• Deadline - 10 months

• Team A

- using waterfall model

Team B

- using Agile Software Approach.

- Requirement Analysis → 1m

- Design → 2m

- Implementation → 4m

- System Testing → 3m

- divided the requirement

↳ Iterative

- deliver working software (1m)

At any point of time, if there is a change in requirements then team A has to start the work from beginning while team B can continue. So, there is an overhead of 'Efforts'.

So, Agile Softwares are preferred.

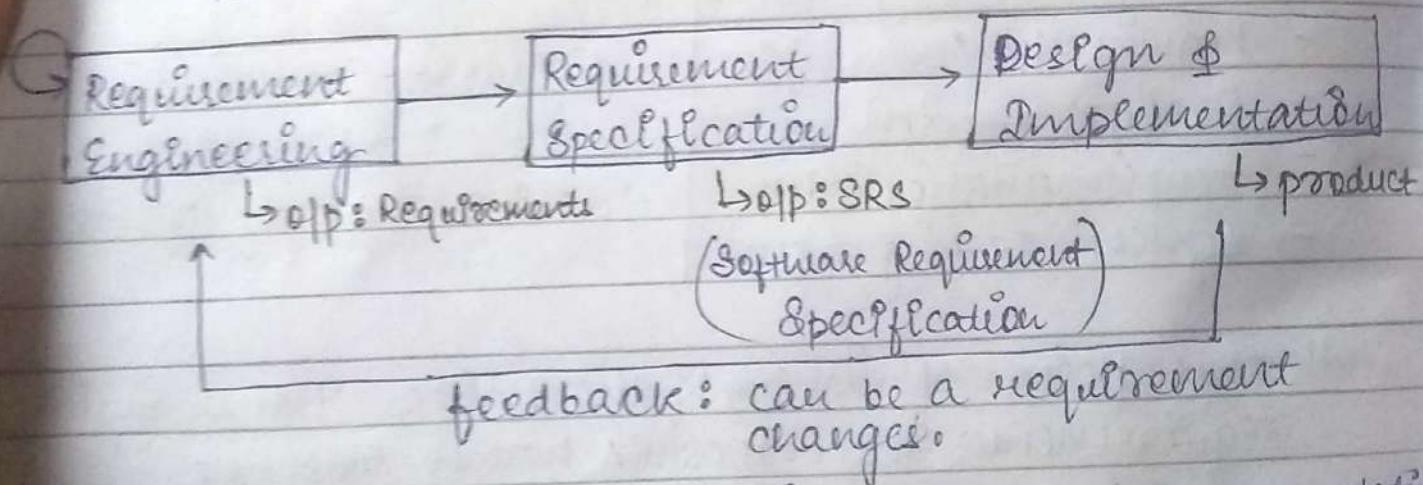
## Video 3: Plan - Agile - Development

### \* Differences between Plan driven & Agile Development :

In plan-driven models, like waterfall model, spiral model, prototype model the output of one stage acts as the input of the other stage.

Quiz: Plan driven model is also known as Agile software development. False.

\* Plan driven: In plan driven, we have different stages, such as

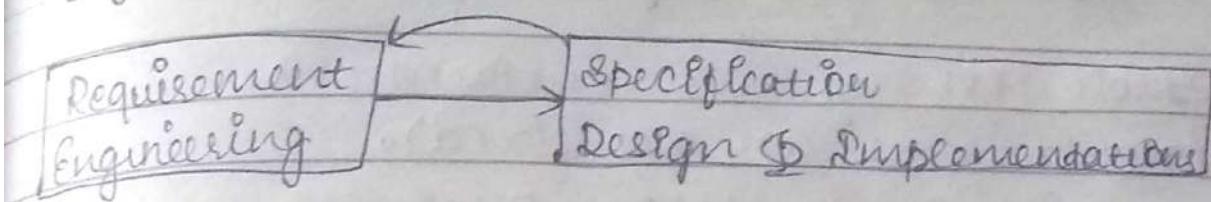


feedback: can be a requirement changes.

For Requirement Engineering & Design & Implementation, iterations can happen at any of the activities. Across requirement specification there cannot be any iteration.

Thus, for each of the stages, there must be a planning.

## \* Agile Model :



- Stages are interleaved (overlapped).
- There can be a feedback from one stage to other stage.
- Iterations occur across different stages.
- Thus, if any changes in the requirement is to be done, it goes back to requirement engineering phase & the requirements are redefined.
- Limited documentation is considered.

Based on the different conditions, different Agile methods are used. Two of them are:

XP (Extreme Programming), Scrum

## \* Applicability of Plan-driven process

- 1). suitable for stable environment.
- 2). suitable for large product & large team size.
- 3). suitable for development of critical systems.  
↓  
<safety, security & reliability> are most important.
- 4). suitable for experienced personnel available at beginning.
- 5). structure & order of process is most important.

## \* Applicability Of Agile Process:

- 1). suitable for dynamic environment.  
(requirements are not fixed).
- 2). not suitable for large products.
- 3). not suitable for critical systems.
- 4). requires experienced personnel throughout the development process.

## Video 4 : XP (Extreme Programming)

## \* Agile Development Techniques

1. Extreme Programming (XP) - used where
  - customer involvement
  - exhaustive testing
  - doc → user stories
  - test-driven approach
2. Scrum - Project management tool / framework controlling iterations.
3. Dynamic Systems Development Method (DSDM) :- used for combining agile & iterative development.
4. Feature-Driven Development (FDD) :- Client centric & practical approach.

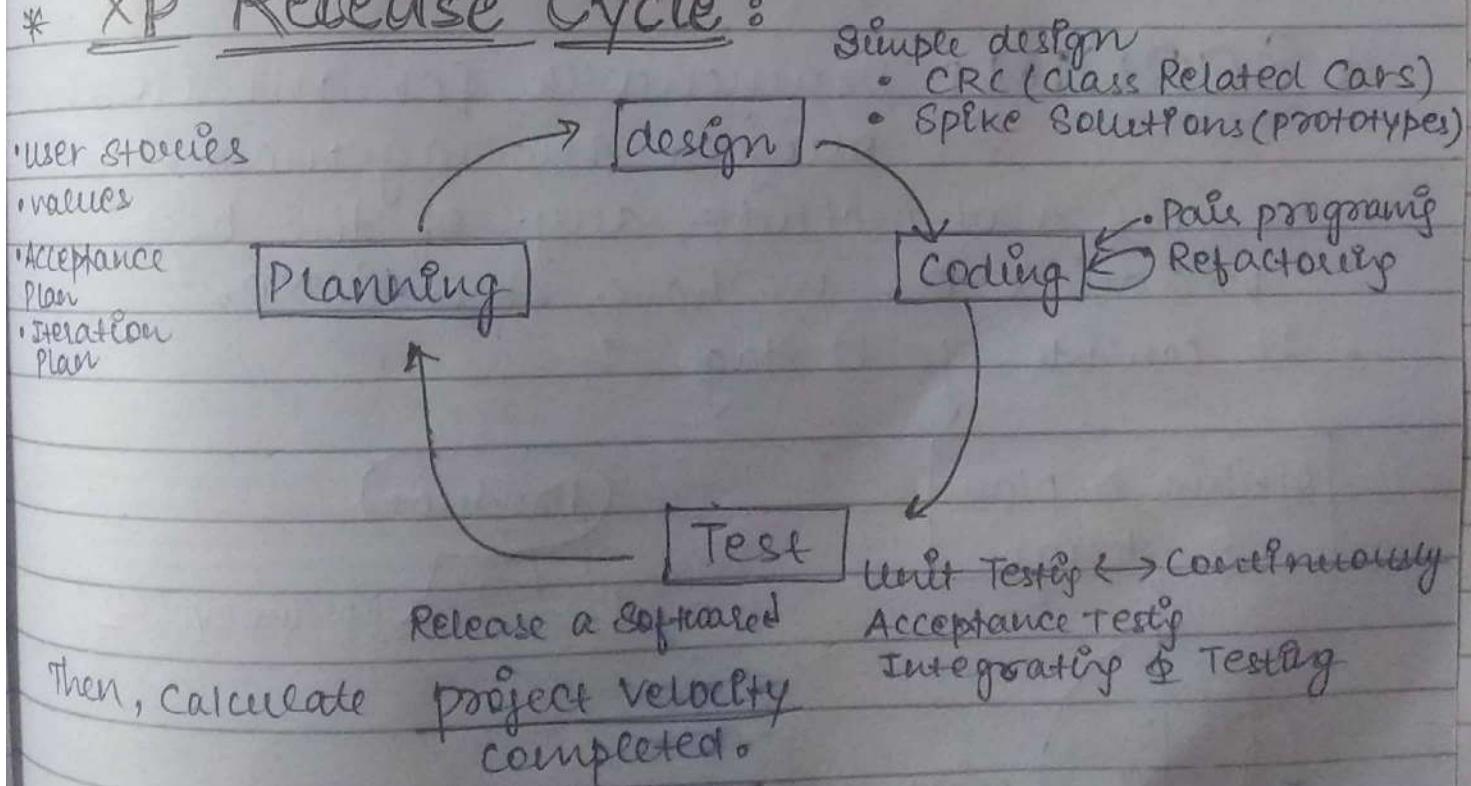
## \* Extreme Programming (XP) :-

- It is a lightweight agile technique.
- It can be used for Object Oriented Applications.
- Many new versions can be developed per day.
- Test driven approach for each of Iterations.
- It is characterized by quick delivery of actions.

### \* Key practices throughout its life cycle :-

1. User Stories → <sup>User Requirements</sup> Requirement Specification
2. Refactoring → change in req. → change in struc. (code).
3. Pair Programming → Two people do the coding
4. Test-first development → Test cases & planning will be paired before actual development starts.

## \* XP Release Cycle :



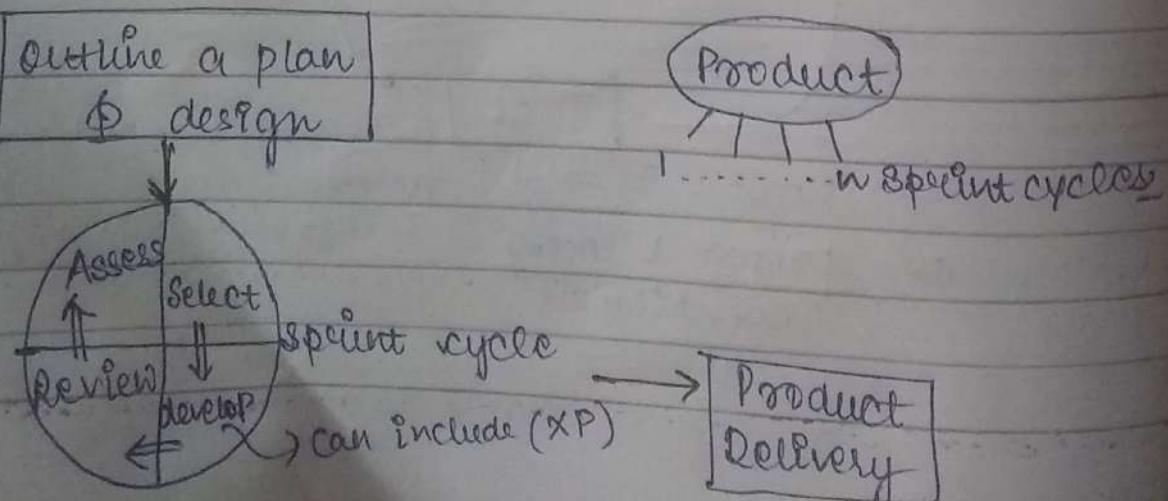
Quiz: Refactoring is a key practice in Extreme Programming. True

### \* Agile Project Management :-

- Project management
  - monitoring the workflow of the project
  - product checking if the product is delivered in time or not.
- Project manager takes care of this
- In case of Agile Project management, we need to manage increments i.e. we need to check after some increments, so that the whole product is delivered in time.
- For this, we go for Scrum.

### \* SCRUM: Framework for monitoring project management.

- It does not include any of the best practices as we have seen in XP.
- It consists of 3 stages :-



Quiz: How many phases are there in Scrum?

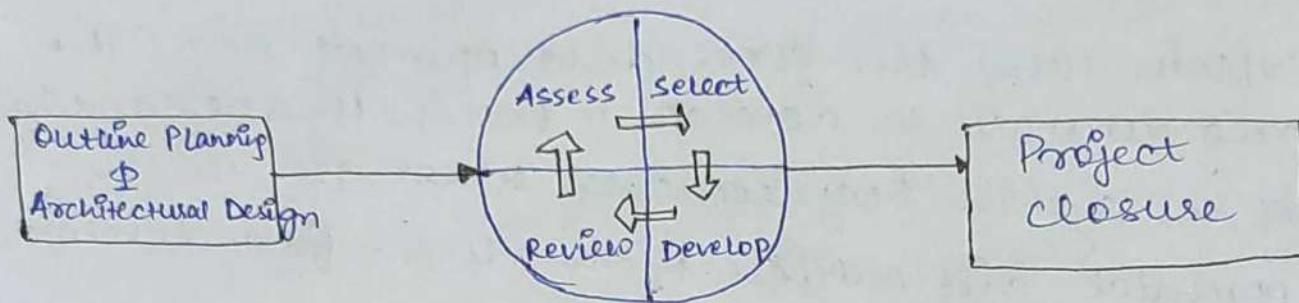
Three

Quiz: Sprint Cycle runs for 2-10 weeks.

True

## Chapter 2: Agile Software Development

Q). Draw the scrum process and list the key characteristics of this process.



- Scrum is an agile method that focuses on managing iterative development rather than specific agile practices.
- It is a framework for monitoring project management.
- It does not include any of the best practices as seen in Extreme Programming.
- There are 3 phases in Scrum:-
  - <i></i> The initial phase is an outline planning phase where you establish the general objectives for the project & design the software architecture.
  - <ii></i> This is followed by a series of sprint cycles, where each cycle develops an increment of the system.

<iii> The project closure phase wraps up the project, completes required documentation such as system help frames & user manuals & assesses the lessons learned from the project.

3). Explain why test-first development helps the programmer to develop a better understanding of the system requirements. What are the potential difficulties with test-first development?

- Test-first development helps with understanding the requirements to discover what is intended.
- Writing tests are sometimes impossible because the system requirements are impossible.
- Writing tests before code clarifies the requirements to be implemented.
- In test-first development, each scenario is assessed & broken down as tasks. Each task represents the feature of the system.
- Thus, test-first development helps the programmer to develop a better understanding of the system requirements.

\* Difficulties with Test-first development:-

- a). Programmers prefer programming to testing & sometimes they take short cuts when writing tests. For example, they may write incomplete tests that do not check for all possible expectations that may occur.

- b). Some tests can be very difficult to write incrementally.  
E.g.: In a complex user interface, it is often difficult to write unit tests for the code that implements the 'display logic' & workflows b/w screens.
- c). It is difficult to judge the completeness of a set of tests. Although you may have a lot of system tests, your test set may not provide complete coverage.

4). Principles of Agile Methods :-

## Agile Manifesto:

↳ doc that includes set of values/best practices that are to be considered.

4 values:

- 1). Individuals & Interactions over process & tools.
- 2). Working Software over comprehensive documentation.
- 3). Customer collaboration over contract negotiation.
- 4). Responding to change over following a plan

## ① Individuals & interactions over process & tools;

+ traditional

- In order to satisfy the need of customer, there must be interaction b/w the product developer & the customer.
- So, ASD uses a lot of communication to help the product make more effective, using different techniques E.g: informal meetings, collective surveys etc.

## ② Working Software over comprehensive documents

- If the customer can use the product in intermediate stage, it helps the development team to get the feedback. So, instead of planning everything in the beginning, ~~ASD~~ it tells us to execute the current iteration then move to the next stage or else there may be a chance of wastage of plan.

## ③ Customer collaboration over contract negotiation

- Customer's input is considered as a very imp. part in the development of the product.  
So, consider it for every activities whenever required.

## ④ Responding to change over following a plan

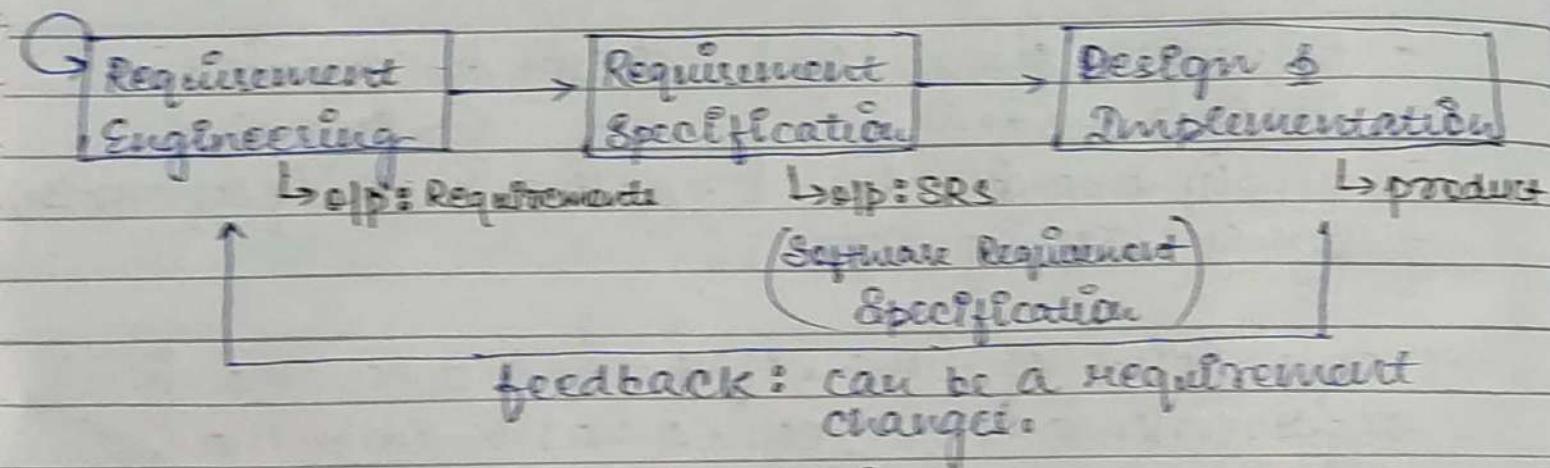
- ~~One~~ Instead of sticking to plan decided in the early stage, go for frequent plans.

## \* Differences between Plan driven & Agile Development:

In plan-driven models, like Waterfall model, spiral model, prototype model the output of one stage acts as the input of the other stage.

Q: Plan driven model is also known as Agile software development. False.

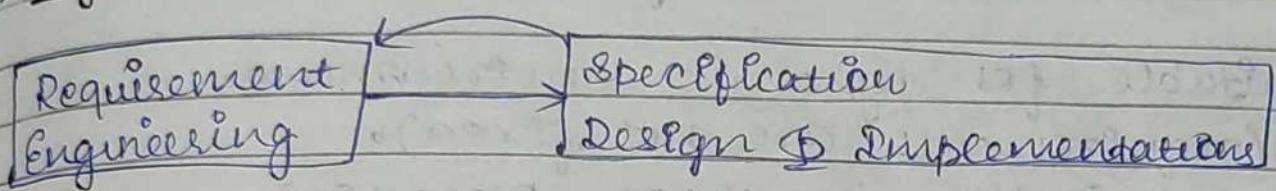
\* Plan driven: In plan driven, we have different stages, such as



For Requirement Engineering & Design & Implementation iterations can happen at any of the activities. Across requirement specification there cannot be any iteration.

Thus, for each of the stages, there must be a planning.

## \* Agile Model :



- Stages are interleaved (overlapped).
- There can be a feedback from one stage to other stage.
- Iterations occur across different stages.
- Thus, if any changes in the requirement is to be done, it goes back to requirement engineering phase & the requirements are redefined.
- Limited documentations is considered.

\* Based on the different conditions, different Agile methods are used. Two of them are:

XP (Extreme Programming), Scrum