06-10-20

Software Engineering

VINCE JOSEPH
S3 MCA
Roll : 59

1.) The essential attributes of a good s/w are:

a) Maintainability

s/w should be written in such a way ~~to~~ so that it can ~~be~~ evolve to meet the changing needs of customers. s/w changes are inevitable so maintainability is a prime attribute.

b) Dependability

includes various characteristics, ~~a~~ dependable s/w should never cause any physical or economical damage at the time of s/m failure. includes range of characteristics like security, safety, reliability etc.

c) efficiency

s/w application should overuse s/m resources like mry and processor cycle.

d) usability

Every s/w application must have enough UI resources and documentation.

2.) FURPS categories

FURPS is a requirement ~~gathering~~ categorising technique. Developed by Hewlett packard.

FURPS includes

i) functionality : implies what an application should do.

ii) usability : implies how a s/w would look like.

iii) Reliability : implies how reliable a s/m should be. ie s/m will fail how often etc.

iv) performance : implies how efficient the s/m should be. (speed, mly usage, disk usage)

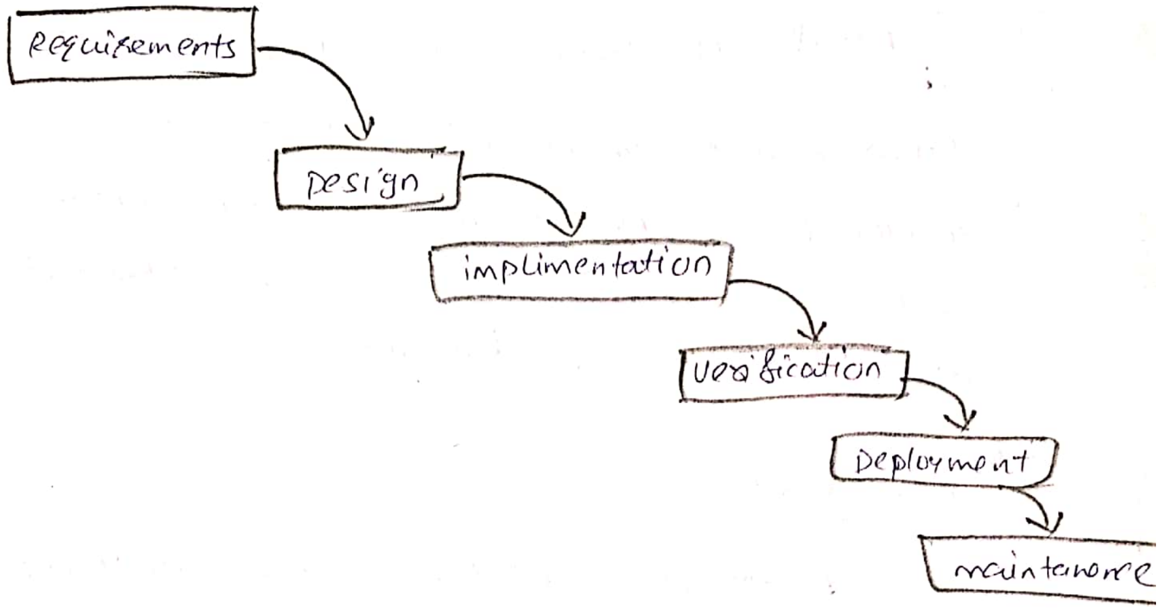
v) supportability : implies how easy it is to support the application.

6) Pure predictive model

A pure predictive model assumes that each stage of development is finished completely and correctly before the next stage begins.

The best example for a pure predictive model is water fall model.

Diagram



As generaly the stages mentioned above can be completed only when the immediate ~~Pr~~ predecessor stage is completed already.

5.) ~~Q~~ Qualities of ~~a~~ good requirements.

~~A~~ Requirements are the features that an application must provide. Also requirements give insight to developers at begining and guides them at developing stage and helps to verify features at finished stage.

The following are some of the properties/ characteristics of ~~a~~ good requirements

1.) Clear

Good requirements must be / should be clear, easy to understand and concise for a developer. It may contain some abbreviations / technical terms which is understandable for both customer and developer.

2.) unambiguous

unambiguity of requirements is generally harder to achieve. Ambiguity in software requirements means that a single reader can interpret the requirement in more than one way or multiple readers come to different interpretations.

3.) Consistency

Requirements must be consistent with each other. They shouldn't contradict each other and also don't provide so many constraints that make the problem unsolvable.

4.) prioritized

for a user ~~point of view~~, they may have much more no. of requirements ~~to of~~ than the actual needed and important ones. so it is nice to cut ~~them~~ the nice - to have features from the design. Because otherwise, the time, cost of developing a s/w will rise. So a developer needs to prioritize the requirements. If you've assignad costs and priorities to the requirements, then you can defer the high-cost priority requirements until a later release. A method called MOSCOW method is used to prioritize requirements

MOSCOW stands for
must - M should - S could - C won't - W

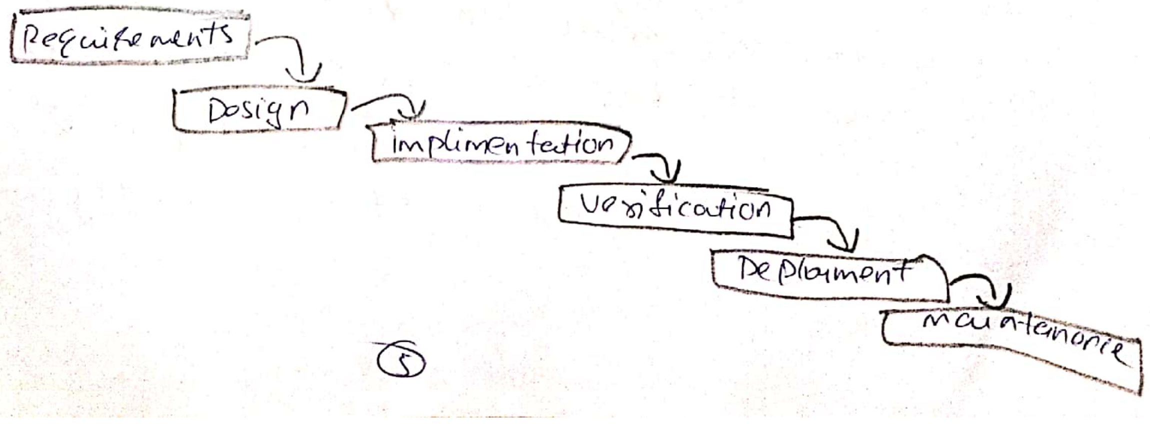
6.) verifiable

Requirements must be verifiable at run time, otherwise we can't ~~assume~~ guarentee that the specific feature has implemented. Being verifiable means the requirements must be limited and precisely defined.

7.) varieties of waterfall model

waterfall model is a pure predictive model. means that next step is started only when current step is completed.

i) waterfall model

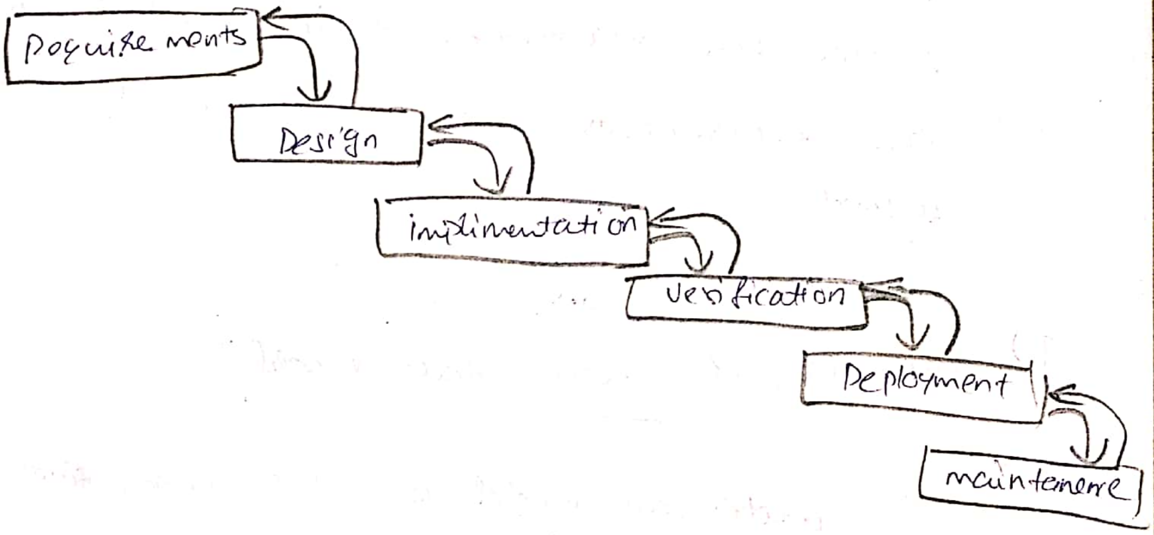


ii) waterfall with feedback

waterfall model will be 100% success if you can carry out each step perfectly.

But in realtime it is very hard. Because that model doesn't allow to go back to earlier steps. so if we fail at one step, later steps will be wrong too.

So there comes the waterfall model with feedback. it allows us to go back one step ahead from current stage. Also it is harder to go further backwards in this model than 1 stage.



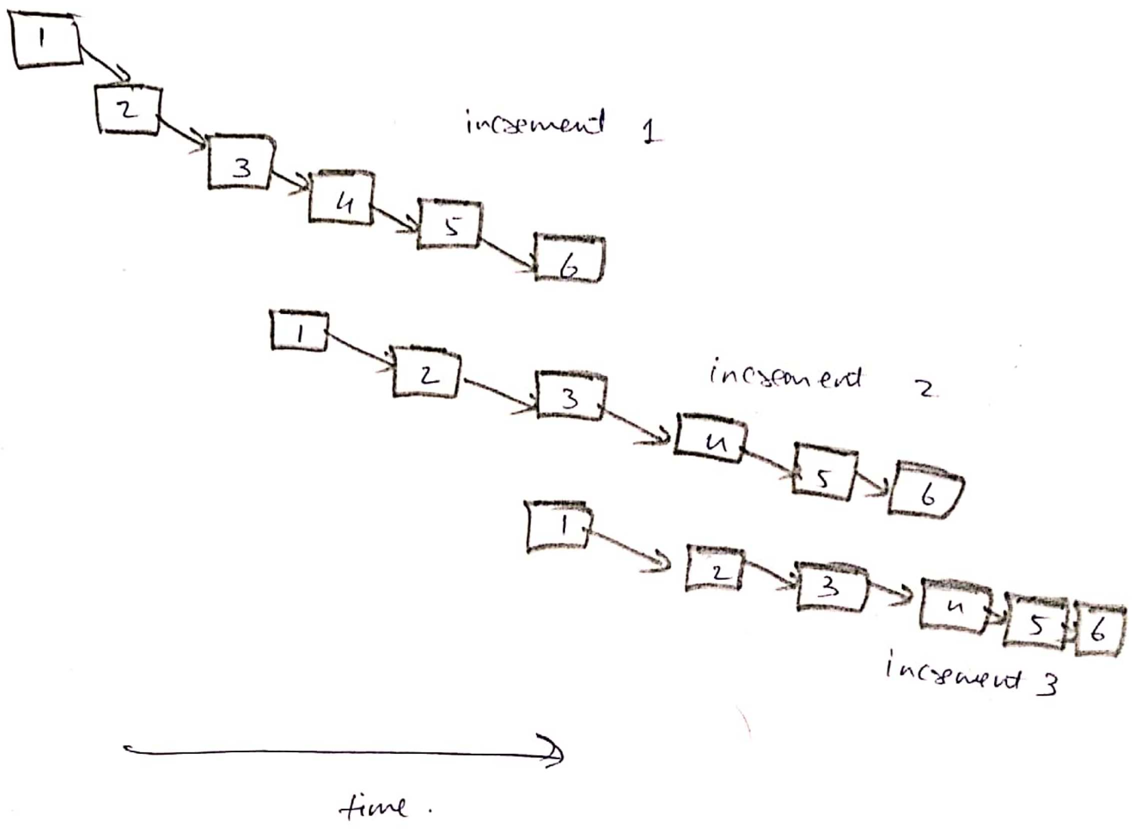
iii) incremental waterfall

This method uses a series of separate waterfall cascades. Each cascade ends with the delivery of an usable application called the increment. Each increment includes more features than the previous one, so you' building the final application incrementally.

During each increment, we get a clear idea of what application should look like.

Scanned with CamScanner

The different increments overlap in time. This model is somewhat adaptive over long timeline.



3.) The incremental approach is a method where model is designed, implemented and tested incrementally until the product is finished. it involves both development and maintenance. The product is ~~[illegible]~~ defined as finished when it satisfies all its requirements.

But in iterative design, it is based on a cyclic process of prototyping, testing, analyzing, and refining a product or process. Based on the result of testing the most recent iteration of a design, changes and refinements are made.