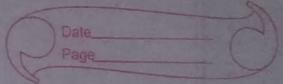


Date

5/2/23

## Software Engineering - Assignment 2.



1 Define waterfall model. Mention the Advantage and Disadvantage of model.

A) Waterfall Model was the first process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase and there is no overlapping.

Requirement

Analysis

System

Design

Implem-

entation

Testing

Deployment

Maintenance

## The sequential phases in Model are.

- ① Requirement Gathering and Analysis: All possible requirements of the system to be developed are captured in this phase and documented in a requirement's specification document.
- ② System Design: The requirement specification documents from first phase are studied in this phase and the system design is prepared. The system design helps in specifying hardware & system requirements.
- ③ Implementation: With inputs from system design, the system is first developed in small programs called units, which are integrated in next phase. Each unit is developed and tested for it's functionality, which is referred to as Unit Testing.
- ④ Integration Testing: All units developed are combined. Post Integration the entire system is tested for faults & failures.
- ⑤ Deployment of System: Once the functional & non-functional testing is done, product is deployed in customer environment.
- ⑥ Maintenance: Some issues come up in client environment. To fix those, Maintenance patches are released.

## Advantages.

- ① Simple and easy to understand & use.
- ② Easy to manage due to rigidity of model.
- ③ Phases are processed and completed one at a time.
- ④ Works well for smaller projects where requirements are very well understood.
- ⑤ Clearly defined stages.
- ⑥ Well understood milestones.
- ⑦ Easy to arrange tasks.
- ⑧ Process & results are well documented.

## Disadvantages.

- ① No working software is produced during life cycle stages.
- ② High amounts of risk & certainty.
- ③ Not a good model for complex & object-oriented projects.
- ④ Poor model for long & on-going projects.
- ⑤ Difficult to measure progress.
- ⑥ Cannot accommodate changing requirements.
- ⑦ Adjusting scope can end the project.
- ⑧ Risk & uncertainty is high.

## •) Distinguish: Verification & Validation.

### Verification

- ① The verifying process includes checking documents, design, code, and program.
- ② It does not involve executing the code.
- ③ Verification uses methods like reviews, walkthroughs, inspections, and desk-checking, etc.
- ④ Whether the software conforms to specifications is checked.
- ⑤ It finds bugs early in development.
- ⑥ Target is application.
- ⑦ It comes before validation.
- ⑧ QA team does verification & make sure document is as per SRS.

### Validation

- ① It is a dynamic mechanism of testing and validation the actual product.
- ② It always involves executing code.
- ③ It uses methods like Black-box, White-Box Testing.
- ④ It checks whether software meets requirements and expectations of a customer.
- ⑤ It can find bugs that verification cannot catch.
- ⑥ Target is product.
- ⑦ Validation is executed on software code.
- ⑧ It comes after verification.

3) what is meant by software prototyping?

A): Software prototyping is similar to prototyping in other industries. It is an opportunity to manufacturer to get an idea of what final product will look like before additional resources, such as time & money, are put into finalizing product. Prototyping gives software publisher the opportunity to evaluate the product.

The Software Prototyping process.

① Identify initial requirements: In this step the software publisher determines what software will be able to do. The software publisher determines what exactly client wants from the product.

② Develop initial prototype: Accept data, validate data, perform calculations, interact with files, produce outputs, Application generators,

③ Review: Once prototype is developed publisher checks what it looks like.

\* Advantage of prototyping:

① The customer get to see the partial product

② New requirements can be accommodated easily.

③ Errors can be detected early.

④ Developed prototype can be reused by developer.

\* Disadvantage:

① Costly w.r.t. time & money.

② Too much variation

③ Poor Documentation due to changes.

④ Customer might lose interest if prototype is not good.

306# - ⑥

(4) What is SRS review? How is it conducted?

- A):
- ① Software documentation is also known as software Requirement specification (SRS) which lists out all the requirements stated by the user in a consistent manner.
  - ② Great SRS leads to great product; we have to keep in mind that goal is to create great products and great software. A great product can be created only from a great specification.
  - ③ SRS originates at the feet of the stakeholders who demand the requirements. They specify desired functions, quality attributes.
  - ④ SRS helps software engineers to better understand the problem they will work to solve. It encompasses a set of tasks that lead to understand what impact the software will have on business.

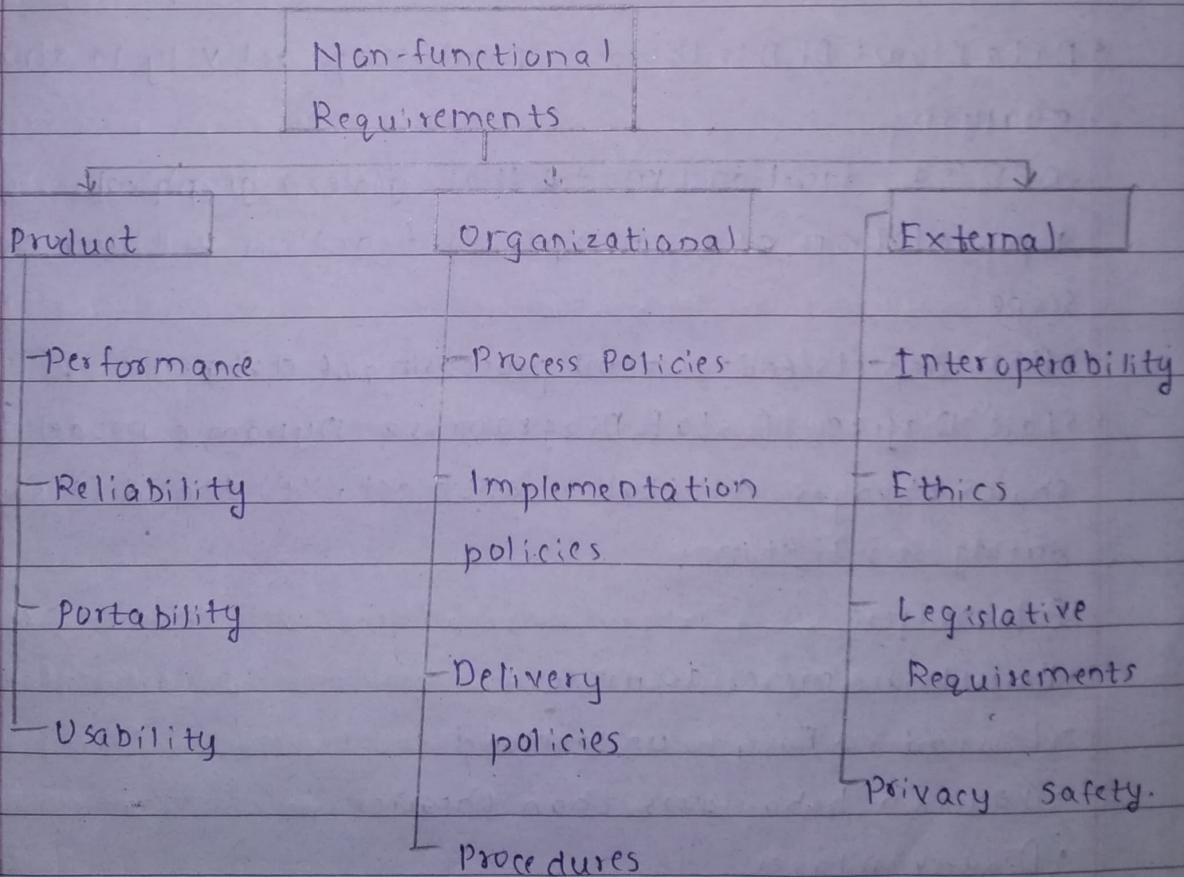
SRS can be conducted in 2 ways:

- ① First, the SRS could be written by client of a system. Second, the SRS could be written by developer of a system. The two methods create entirely various situations and establish different purposes for document altogether.
- ② The first case SRS is used to define the needs and expectation of users.
- ③ The second case: SRS is written for various purposes & serves as a contract between customer and developer.
- ④ A good SRS document should be Concise, structured, verifiable and Traceable.

### 5) Non-functional requirements of a software.

A): Non-functional requirements can be again categorized as followings:

- ① Product Requirements: States the behaviour of product - performance, reliability, portability & usability.
- ② Organizational requirements: specify the policies & procedures followed in customer's & developers organization - process policies, implementation policies & delivery policies & procedures.
- ③ External requirements - state the external factors of the system and the development process - interoperability, ethics.



## (6) Define Behavioural Modelling.

- A)= ① Behaviour Model is specially designed to make us understand behaviour and factors that influences behaviour of a system. Behaviour of a system is explained and represented with help of a diagram. This diagram is State-Transition Diagram.
- ② It is a collection of states & events. It usually describes overall states that a system can have & events for which changes are made.

③ There are 2 types of Behavioural Models:

- (a) Data Flow Diagram
- (b) State Diagram.

• Data Flow: DFD is the core modelling activity in structured analysis.

- DFD is a functional model that gives a graphical representation of an enterprise's function within a defined scope.

- DFD depicts end-to-end processing of data.

• State Diagram: A state Diagram is a Dynamic model that shows changes in state of an object, i.e. the diff. states during it's lifetime.

• Advantages:

- Results are more accurate

- Less cost & Easy to understand

- Focus on behaviours than theories.

• Disadvantages:

- No principle

- Not Fully Automated

- NO Maximum Productivity.

7) Draw an ER diagram for university information system. Specify at least four cardinality and modality relationship.



8. Explain the following.

(i) SCM repository: SCM (source code Management) is used to track modifications to a source code repository. SCM tracks a running history of changes to a code base & helps resolve conflicts when merging updates from multiple contributors. In computer Software Engineering, software configuration management (SCM) is any kind of practice that tracks and provides control over changes to source code. Software developers sometimes use revision control software to maintain documentation & configuration files as well as source code.

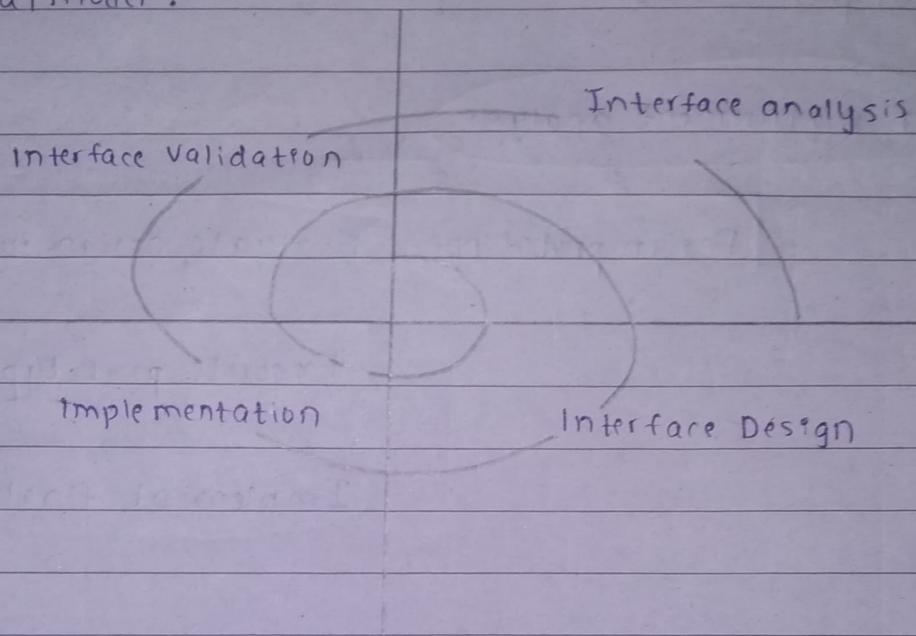
(ii) SCM process: SCM process is a tool that keeps necessary change has been implemented adequately to appropriate component. The SCM process defines number of tasks:

- ① Identification of objects
- ② Version control.
- ③ Change control.
- ④ Configuration Audit.
- ⑤ Status Reporting.

SCM recommends requirements keeps on changing so we should keep on upgrading systems based on current requirements and needs to meet desired outputs. Changes should be analyzed before they are made to an existing system.

## g. Short notes on User Interface Design Process.

A) The UI Design process is an iterative process represented by a spiral model:



The UI design process is categorized into 4 major activities:

### 1. Interface Analysis:

- Interface Analysis means evaluating UI Design without users in order to:
  - Understand the end-users who will interact with system by UI.
  - The user analysis activity refers to skill level and business needs of end users who will interact with system.
  - Once requirements are gathered, task analysis is conducted.
  - The environment analysis focuses on questions like - where will the interface be located physically.

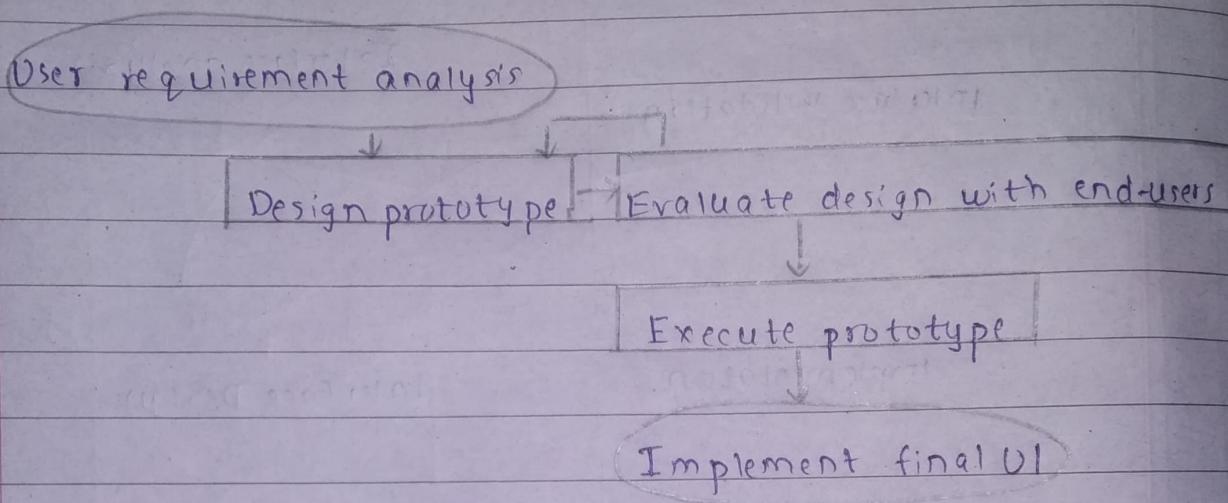
### 2. Interface Design:

- This phase defines a set of interface objects and actions that allow a user to perform all tasks that are defined in user requirement analysis phase.

3067 (12)

- It is an attempt to confirm that designed tasks meet every usability goal defined in user requirements.

### 3. Implementation.



### 4. Interface validation.

This focuses on:

- The ability of UI to implement every user task correctly.
- The degree to which the UI is easy to use & learn.
- User's acceptance about interface as an efficient tool in work.

(10) Discuss in detail about the design process in the software development process.

- A) -
- ① A good User Interface Design can make a product to be accepted or rejected in the market.
  - ② If end-users feel it difficult to learn & use then even an excellent product could fail.
  - ③ The main objective of the Software Design phase is to design proposed system using all information collected during preliminary investigation.
  - ④ Every end user has different level of capabilities. Some people may see or hear better than others, some may be color blind or some may be good in handling the system better than other.
  - ⑤ Design phase includes full:
    - Design of all types of inputs of proposed system.
    - Design of all types of outputs of proposed system.
    - Design of procedures which convert input to output.
    - Design of flow of information.
    - Design of info. store within a file & database volume.
    - Design of collection of inputs using forms.
    - Design of program specification.
    - It determines hardware cost, speed of software, error rates, changes to made in structure.
    - Also design standards for testing, documentation, etc.