SE (Minor Test)

Page No. 1
Date 11 6 21

Name → Subham Sharma
Class → MCA (IInd sem)
Roll no. → 58 (2023475705 4)
Subject → Software Engineering
Sub. code → MCAC-203

Que 1- What is software crisis? What are the causes and how software engineering deals with it?

Ans Software crisis: It is the term used in early days of computation science for the difficulty of writing useful and efficient program in required time.

If we will use same workforce, same tools after the fast increasing complexity of software design and efficiency demands, it will result in some problems. These problem give birth to software crisis.

Causes of Software Crisis:

→ The cost of owning and maintaining software was as expensive as developing the software.

→ Project may take much time than the time decided.

→ The quality of software was very low.

→ Software often did not meet requirements.

→ The average software project overshoots its schedule by half.

At present, software engineering appears to be among the few options to tackle the present software crisis. Software engineering is a solution to this crisis because it is a systematic, disciplined and quantifiable approach. Software crisis may be avoided if we focus on following points while desiging or developing any project

→ Reduce the over-budget of software.
→ Maintain high quality of software.
→ Complete the project in less time.
→ Deliver software on time.

Thus this was all about Software crisis and its causes and how software engineering deals with it.

Answer

Que2. What are different techniques for Require-ments Gathering ?

Ans Requirements gathering is also popularly known as requirements eliciation. The primary objective of requirements gathering task is to collect requirements from the clients or the stakeholder (a stakeholder is a source of the requirements and is usually a person or group of persons who either directly or indirectly is concerned with the software).

Requirement gathering may sound like a simple task. However, in practice it is very difficult to gather all the necessary information from a large number of stakeholders. Following are few basic steps of gathering Requirements.

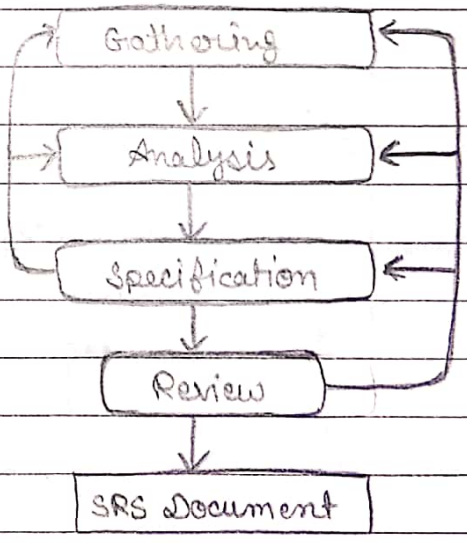
→ Observe existing systems

→ Study existing procedures

→ Discuss with end-users

→ Input and output analysis

→ Analyze what is to be done.



Thus this was all about requirements gathering.

Answer

Que3. Define the functional and Non-functional requirements with help of suitable example.

Ans Functional Requirements :- In software engineering a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behaviour and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Functional software requirements help you to capture the intended behaviour of the system. This behaviour may be expressed as functions, services or tasks or which system is required to perform.

Non-functional Requirements :- A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. For example, how fast does the website load?

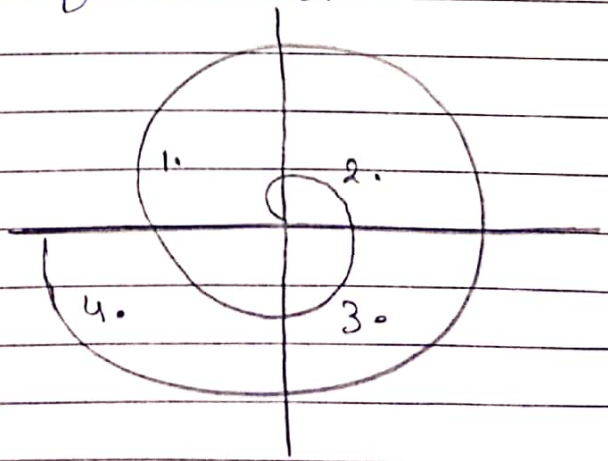
Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. For example, the

website should load in 3 seconds when the number of simultaneous users are more than 10,000. Description of non functional requirements is just as critical as functional requirement.

Thus this was all about Functional and non-functional requirements.

Que 4. How can spiral life cycle model help to improve software development process when compared to the waterfall life cycle model ? Explain.

No doubt spiral life cycle model is much improved and best to use as compared to waterfall life cycle model. because spiral model is one of the most important software Development life cycle models, which provides support for 'Risk Handling' that was almost impossible in watofall model.



In its diagrammatic representation, it look like a spiral with many loops. Each loop of spiral is called a phase of software development process. Following are the main points on which spiral model is considered best as compared to waterfall model :-

→ Waterfall model works in sequential method whereas spiral method works in evolutionary method.

→ Flexibility to change in waterfall model is difficult whereas flexibility to change in spiral model is not difficult.

→ There is low amount risk in spiral model as compared to waterfall model.

Thus, in these ways spiral lifecycle model help to improve software development as compared to waterfall life cycle model.

Answer

Ques. What is Agile Software Development model?

Ans This process model refers to a model that is being used as an umbrella term to refer to a group of development processes. While these processes share certain common characteristics, yet they do have certain subtle differences among themselves. A few popular agile SDLC models are following:-

→ Crystal
→ Atern
→ Feature-driven development
→ Scrum
→ Extreme programming
→ Lean development
→ Unified process

In an agile model, the requirements are decomposed into many small parts that can be incrementally developed. The agile models adopt an incremental and iterative approach.

A central principle of the agile model is the delivery of an increment to the customer after each time box (the time to complete an iteration is called time box).

The agile methods derive much of their agility by relying on the tacit knowledge of the team members about

the development project and informal communications to clarify issues, rather than spending significant ammounts of time in preparing formal documents and reviewing them.

—X—X—X—