Unit III: Software Requirement Engineering
(weightage - 14 marks)

① Explain Core principles of software Engineering. (4/6m)
or
Enlist Core principles of software. (4m).

⇒ Principle 1: Think
- Thinking before doing usually leads to better result.
- When you think about something you are likely to do it correctly.
- You also learn how to do it correctly next time.
- If you think about something and still do it wrong, it becomes a valuable experince.
- Good thoughts bring valuable result.

Principle 2: The Reason all it's exists.
- Software is made useful for its users.
- Before deciding what the software will do or how it's made, ask "Does this really help the user?".
- If answer is no, then don't include it.
- Before speciting a system requirement, functionality before determining the hardware platform, first determine whether it adds value to system.

Page No.:
Date: / /

Principle 3 : Keep it Simple and stupid.
- All design should be simple but no simpler.
- Simple design are easier to understand and maintain.
- Keeping it simple doesn't mean losing important features.
- Usually, the best design are simplest ones.

Principle 4 : maintain the vision
- A clear vision is essentential for suessful software project
- Changing the main plan weakens even a good system.
- Having a strong leader to maintain and enfore the vision, leads to sucess.

Principle 5 : what You Produce, other will consum
- when you make something, think about how others will understand it.
- Lots of people might end up using what you make.
- When you plan, think about the people who will built it. when writing, think about ~~how~~ those who will work on it later.
- Remberd, Some one might need to FIX Your work, so think them as users too.

Principle 6: Be open to the future.
- Software that last longer is more valuable
- It should be able to handle changes.
- Design software to be flexible from to start.
- To do this sucessfully, these systems must be ready to adapt changes.
- Always ask "what if" and prepare for different possibilities.

Principle 7: Plan Ahead for Reuse.
- It leads to creating of new products.
- Reuse saves times and effort.
- The reuse code and design has a major benefit of using object-oriented technologies.
- Planning ahead for reuse reduces the ~~cost~~ cost and increases the value of both the reusable components and systems in which they are incorporated.

(2) State and explain communication principles of Software Engineering. (6M/4M). imp

⇒ Principle 1: Listen Carefully
- Pay attention to what the speaker is is saying instead of thinking about your response right away.
- If you don't understand something, ask pdietly for clarification, but try not to interrupt to much.

- Don't show disagreement or annoyance with your body language while the person is talking, like rolling your eyes or shaking your head.

Principle 2 : Face-to face communication is best
- Talking in person is better when there's something to see along with the talk.
- For example, something might draw or show a document to help everyone understand.
- Pictures make it easier to talk about thing and understand each other.

Principle 3 : Prepare before you communicate
- Spend the time to understand the problem before you meet with others.
- If necessary, perform some research to understand bussiness domain.
- If you have responsibility for conducting a meeting, prepare an agenda in advance of meeting.

Principle 4 : Someone should facilite activity
- Every communication meeting should have a leader (facilicator).
- To keep conversion moving in productive direction.
- To mediate any confict that does not occur.
- To ensure the other principle are followed.

Principle 5 : Take notes and document decesion
- Taking notes during meeting to remember what's said.
- Write down decesions made so nobody forgets.
- Keeping notes helps everyone stay on track.
- Documenting decesions ensures everyone knows what's been agreed upon.

Principle 6 : If something is unclear, draw pictur
- Yerbal communication goes on so fast.
- A sketch or drawing can often provide clearity when words fail to do the job.

Principle 7 : Negotation is not a contest or a game.
- Negotation is about finding solutions where everyone benefits.
- Sometimes you need to negoticate on tasks, priorities or deadlines.
- Even with shared goals & negotations means both sides may need to compomise.
- Sucessful negotation happens when everyone is willing to give a little to find agreement.

(3) Describe four principles of good planning.
or Planning practices principles in SEN. (4M)

⇒ Planning is an activity that includes the set of management and technical practices that software has to follow in definite direction.

✓ Principle 1: To understand scope of project.
- Know exactly what tasks are involved.
- Figure out what's not part of project too.
- Be clear about what you're aiming to accomplish.
- Make sure everyone understands what's included and what's not.

✓ Principle 2: Involve customer in planning activity.
- Work together with the customer.
- Understand what they want and need.
- Make sure that, their ideas are part of plan.
- Keep them informed and involved throughout planning process.

Principle 3: To consider risks as you define the plan.
- Think about what could go wrong.
- Identify potential problems or uncertainties.
- Come up with ways to deal with them.
- Be ready to adapt if things don't go as planned.

Page No.:
Date: / /

Principle 4: Estimate Based on Available Information.
- Use what you already know.
- Make guesses based on facts.
- Figure out how long things will take.
- Think about what resources and money you'll need.

Principle 5: Track the plan Frequently and make Adjustment when necessary.
- Keep an eye on how things are going.
- Check progress regularly.
- If things aren't getting or going according as planned, make changes.
- Make sure if plan stays in line with vision maintaining.

Principle 6: Be realistic
- Set goals you can actually reach.
- Condiser what you can realistically accomplish.
- Think about any limitations or thang challenges.
- Make sure with goals match what is possible with resources you have.

(4) State and describe any four deployment principles. (4m). 1 time.

⇒ -A software deployment process is structured systematic approach to implementing software solutions in business environment
- it is one of the stages in the software development process, which includes development, testing, deployment and ongoing operations
- It involves installing, configiguring, and testing software application to prepare it for operating in a specific enviroment.

Principle 1: Planning and Assessment.
- The first step in software deployment process is to carefully plan and assess the organization's needs and objectives.
- This involves identifying the specific problems or challenges that the software is intended to address.
- During the planning and assessment phase, it is also important to consider any potential constraints or limitations, such as budget, resources or infrastructure requirement.

Principle 2: Development and Configuration
- Once the organization's needs and objectives have been identified, the next step is to develop or configure the software solution.

- During this development or configuration phase, it is essential to ensure that the software meets the organization requirements and that it is compatible with existing systems and infrastructure.

Principle 3 : Testing and Quality Assurance
- Before deploying the software in a live environment, it is crucial to subject it to rigorous testing process.
- This ensures that the software functions as intended and that it is free of any defects or bugs.
- The testing phase is essential because it helps to identify and resolve any issues before the software is deployed, minizing the risk of problems or downtime.

Principle 4 : Deployment
- Once the software has been tested and approved, it is time to deploy it to its end users.
- This involves installing and configuring the software on the appropriate hardware and infrastructure, or deploying it directly to end users device.

Principle 5 : Monitoring and maintaience
- The software deployment process does not end once software is running in production

(5) Define SRS(Software Requirement Specification) (2m)

⇒ - Software Requirement Specification is SRS.
- It is a document that explains what software system needs to do.
- It covers things like what software should accomplish, how it should work, and what it should look like.
- This document helps everyone involved like customers and developers.
- It understand software goals and features clearly.

(6) State the need of SRS and characteristics of SRS. (4m)

⇒ Need of SRS. (2m)
- An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost.
- A good SRS defines how an application will interact with system hardware, other programs and human users in wide variety of real-world situations.
- The requirement process helps to clarify needs.
- Users do not always know their needs, must analyze and understand the potential.
- SRS errors are expensive to fix later.
- Good SRS can minimize changes & error.
- SRS provide a reference for validation of final product.

Charachtertics of SRS. (2m)

- Complete: The SRS should be complete i.e. All software requirements should be mentioned in SRS.
- Correct: It should be correct i.e it should be according to needs of customer
- Clear: It should be clear i.e The requirements should be clearly declared.
- Accurate: It should have accuracy. If this is not accurate then software cannot be developed.
- Testable: It should be Testable i.e it should be capable of being tested in any way.
- Traceable: It should be traceable, that is each requirement in it should be identified differently. Each requirement should have its own identity.
- Verifiable: It should be verifiable. The requirements are verified by experts and testers.

(7) Explain six functions of requirement engineering process. (6m)

⇒ Requirement Engineering involves understanding what the customer needs, figuring out it's possible, negotation a solution, writing down the requirements clearly, cheking if they make sense, and managing them as they're turned into a system.

This process include seven main steps: starting with understanding the initial idea, gathering requirements, detailing them, discussing and agreeing on them, writing them down clearly, checking if they re correct, and keeping track of them as project progresses.

Inception (starting)

Inception is the initial phase where we ask why project is needed and analyze factors such as business needs, potential markets, and services to be provided.

This helps establish a basic understanding of problem, the stakeholders involved and desired solution, setting scope of project.

Elicitation (gathering requirements)

It involves gathering information by asking customers, users and other stakeholders about the objectives of system or product, what needs to be accomplished, how it aligns with business needs, and how it will be used day-by-day.

Elaboration

The information obtained from the customer during inception and Elicitation and refined during elaboration

This task focuses on developing a refind requirement model that identifies requirements for three domains, functional and behavioral domain. It

- Describe how the end user will interact with the system.
- Business domain entities that is visible to end user.
- The attributes of each class are defined and services that are required by each class are identified.
- The relationship and collabration between classes are identified, and variety of supplementary diagrams are produced.

Requirement management

It comprises activity to identify, control and track requirements and changes throughout the project lifecycle ensuring alignment with project goals and stake holders needs.

Validation

Validation ensures the accuracy and completeness of requirements by conducting technical reviews to identify errors, in consistencies or missing information.

Page No.:
Date: / /

Negotiation

Negotiation involves determining if the requirements are truely necessary by prioritizing them through Stakeholder input and addressing conflicts to ensure satisfication for all parties involved.

Specification

It entails documenting requirement using various formats such as written documents, graphical models or prototypes, tallored to be complexity of software being developed.