**Section A**

1. Steps involved in Requirement engineering for CRM

Requirements engineering (RE) – Is the process of finding out, analysing, documenting and checking services and constraints of the system.

* **Requirements elicitation and analysis** – This is the process of deriving the system requirements through observation of existing systems, discussions with potential users and procurers.
* **Requirements specification** – This is the activity of translating the information gathered during requirements analysis into a document that defines a set of requirements.
* **Requirements validation** – This activity checks the requirements for realism, consistency, and completeness.

1. Discuss the techniques for requirement eliciting analysing and validation.

**Requirements elicitation** involves meeting with stakeholders of different kinds to discover information about the proposed system. You need to spend time understanding how people work, what they produce, how they use other systems, and how they may need to change to accommodate a new system.

**Requirement eliciting analysis techniques**

* **Interviewing** – Conduct a one-on-one discussion with stakeholders to gather detailed information.
* **Observation or ethnography** – where you watch people doing their job to see what artifacts they use, how they use them, and so on.

**Requirements validation techniques**

* **Requirements reviews** – The requirements are analysed systematically by a team of reviewers who check for errors and inconsistencies.
* **Prototyping** – This involves developing an executable model of a system and use it to see if it meets customer needs and expectations.
* **Test-case generation** – Requirements should be testable.

1. Issues affecting diverse types of software

* **Heterogeneity** – Systems are required to operate as distributed systems across networks, include different types of computer and mobile devices, as well as running on general-purpose computers.
* **Business and Social Change** – New software needs to developed as emerging economies develop and new technologies become available.
* **Security and Trust** – We have to make sure that malicious users cannot attack our software and that information security is maintained especially for remote software systems accessed through a web page or web service interface.
* **Scalability** – We need new tools and innovative ways to develop software across a very wide range of scales, from very small embedded systems to Internet-scale, cloud-based systems that serve a global community.

**Section B**

1. Strength and weakness of waterfall model, Incremental model, Agile model, and Spiral model.

**Waterfall model** – is a software development model that cascade from one phase to another. It presents the software development process as a number of stages.

**Strength of waterfall model**

* Simple and easy to understand.
* Well-suited for projects with clear, unchanging requirements.

**Weakness of waterfall model**

* Testing is delayed until the end of the project.
* Not suitable where software requirements change quickly.

**Incremental model** – It is based on the idea of developing an initial implementation, getting feedback from users, and evolving the software through several versions until the required system has been developed.

**Strength of Incremental model**

* Inexpensive to implement requirements.
* Easy to identify and fix errors.

**Weakness of Incremental model**

* The process is not visible.
* System structure tends to degrade as new increments are added.

**Agile model** – are incremental development methods in which the increments are small and new releases of the system are created and made available to customers every two or three weeks.

**Strength of Agile model**

* Emphasizes collaboration and communication.
* Allows for rapid iteration and feedback.

**Weakness of Agile model**

* May lack long-term planning and vision.
* May not be suitable for small projects due to higher development costs.

**Spiral** **model** –

**Strength of Spiral** **model**

* Suitable for projects with evolving requirements.
* Provides maximum control over costs, resources, and quality.

**Weakness of Spiral** **model**

* The model is much more complex.
* Expensive and not suitable for small projects.

1. Model you would use for the development and why based on requirement, timeline and risk-factors.
2. Choosen model impact project management, communication and risk mitigation

**Section c**

1. Why Agile methodology

Agile methods are incremental development methods in which the increments are small, and, typically, new releases of the system are created and made available to customers every two or three weeks. They involve customers in the development process to get rapid feedback on changing requirements. They minimize documentation by using informal communications rather than formal meetings with written documents.

1. Scrum framework: how it can be applied. Roles of scrum.

Scrum is an agile method that provides a framework for organizing and managing agile projects. It focuses on organizing the workflow, providing visibility, and ensuring continuous improvement.

**How Scrum Can Be Applied.**

* Review Work to Be Done.
* Select Items (from Product Backlog).
* Plan Sprint.
* Sprint.
* Review Sprint.
* Potentially Shippable Software.
* Cycle Repeats

**Roles of scrum.**

* **Product Owner**
* **Scrum Master**
* **Development team** – A self-organized group of software developers who are responsible for developing the software and other essential project documents.

1. How you can plan a sprint and the iteration. How you can measure progress of the sprint.

**Sprint Planning**

**Sprint** – A development iteration. Sprints are usually 2 to 4 weeks long.

**How you can plan a sprint.**

Define the sprint goal, select items from product backlog, break down items into separate tasks, create the sprint backlog, timebox the sprint

**Iteration planning**

**Iteration** – it focuses on planning the next increment of a system. This usually represents 2 to 4 weeks of work for the team.

**How you can plan the iteration.**

Daily Scrum, work on task, update task board.

**How you can measure progress of the sprint.**

* Burndown chart – which visually represent the remaining work against time.
* Velocity – Measures the amount of work completed in each sprint.
* Sprint Goal Success Rate – Measures how often the team successfully achieves their sprint goals.

1. How Agile helps addressing feedback.

by promoting a culture of continuous improvement, iteration, and collaboration, enabling teams to quickly incorporate feedback into development cycles.

* **Incremental Development** – Software is built and released in small, manageable versions.
* **Customer Involvement** – Stakeholders provide feedback at every stage.
* **Less Documentation, More Action** – Agile reduces paperwork and focuses on coding.
* **Frequent Releases** – New software updates every 2-3 weeks.

Helps businesses respond quickly to market changes. Allows continuous improvement based on real-world use. Ensures software is always aligned with business needs.

1. Risk management in Agile: potential risks and how to mitigate them. How regular reviews contribute on the identification of risk early.

**Potential risks with agile methodology.**

* **Contractual Challenges**: Agile informality conflicts with traditional legal contracts that require fixed, detailed requirements.
* **Maintenance Issues**: Agile is better suited for new development than software maintenance, which is a major cost for large companies.
* **Distributed Teams**: Agile focus on small, co-located teams struggles with the reality of global, distributed teams.
* **Customer Involvement**: Maintaining customer engagement during software maintenance can be difficult, as changes are less frequent.
* **Team Continuity**: Agile teams work best when they remain intact, but developers may leave for new projects, affecting maintenance efforts.