1. **Explain what software engineering is and discuss its importance in the technology industry.**

Software engineering is a systematic application of engineering principles, methods, and tools to the development and maintenance of high-quality software systems. It involves the design, development, testing, deployment, and maintenance of software products

**Importance of software engineering**

* we can create innovative solutions
* Software engineering provides a way to manage the resources, schedule, and budget of the project, and to ensure that the team is working together effectively.
* it provides a structured approach and helps to manage the costs, risks, and schedule of the project
* Overall, software engineering is essential for creating high-quality software that meets the needs of the users and is easy to maintain

1. **Identify and describe at least three key milestones in the evolution of software engineering**.

* Introduction of Object-Oriented Programming (OOP) (1980s)

Object-oriented programming (OOP) introduced the concept of encapsulating data and behavior into objects, which represent real-world entities. Pioneered by languages like Smalltalk and popularized by C++ and Java, OOP focuses on concepts like inheritance, polymorphism, and encapsulation.

* The advent of structured programming in the 1970s

Structured programming emerged as a response to the complexity and unmanageable nature of early software development practices. Key figures like Edsger Dijkstra advocated for a more disciplined approach to writing code, emphasizing the use of control structures like loops and conditionals, while avoiding unstructured jumps (e.g., GOTO statements).

* The rise of agile methodologies in the 2000s.

The Agile Manifesto, introduced in 2001, emphasized iterative development, collaboration, and flexibility over rigid, long-term planning. Agile methodologies like Scrum and Kanban transformed how teams approached software projects. Simultaneously, DevOps emerged, integrating development and operations to streamline the deployment and maintenance of software

1. **List and briefly explain the phases of the Software Development Life Cycle.**
2. Requirements: - gather and document user and system requirements
3. Design- creating detailed designs of the system architecture and user interface
4. Implementation- actual coding and building of the software
5. Testing- test whether the developed software has met the quality and functional requirements
6. Deployment-release the software to the end user
7. Maintenance- provide updates of the software and offer support after deployment
8. **Compare and contrast the Waterfall and Agile methodologies. Provide examples of scenarios where each would be appropriate.**

**Comparison**:

1. Both methods aim to ensure the quality and functionality of the software, and to satisfy the customer's requirements and expectations
2. Both methods use various types of testing techniques and tools, such as unit testing, integration testing, system testing, regression testing, performance testing, automation testing

**Contrast:**

|  |  |  |
| --- | --- | --- |
| **S/N** | **Waterfall** | **Agile** |
| 01 | Sequential and linear | Iterative and incremental |
| 02 | Low flexibility; changes are difficult and costly once the project is underway | High flexibility; welcomes changes even in late development stages |
| 03 | customer involvement is Limited after the requirements phase | customer involvement is Continuous and high throughout the project |

1. **Describe the roles and responsibilities of a Software Developer, a Quality Assurance Engineer, and a Project Manager in a software engineering team.**
2. Software developer- Do write codes and implements software solutions
3. Quality assurance engineer- Oversee the quality of the software to meet its intended goal by designing and executing test plans
4. Project manager-Oversees planning, execution and delivery of software projects
5. **Discuss the importance of Integrated Development Environments (IDEs) and Version Control Systems (VCS) in the software development process. Give examples of each.**
6. IDEs- Software suites that provide comprehensive tools for writing, debugging, and testing code (e.g., Visual Studio, Eclipse, IntelliJ IDEA)
7. VCS- Software tools for tracking changes to source code and coordinating work among team members (e.g., Git, Subversion)
8. **What are some common challenges faced by software engineers? Provide strategies to overcome these challenges.**
9. Changing Requirements:Requirements may change during the development cycle, leading to scope creep and project delays.
10. Tight Deadlines:Pressure to deliver software products on schedule can result in rushed development and compromised quality.
11. Technical Debt:Accrued from shortcuts or suboptimal solutions, technical debt can impede future development efforts and increase maintenance costs.

**Strategies for Overcoming Challenges**

Strategies for overcoming challenges include effective communication, agile methodologies, prioritization of tasks, and regular reassessment of project goals and timelines.

1. **Explain the different types of testing (unit, integration, system, and acceptance) and their importance in software quality assurance.**
2. Unit- Testing individual components or modules of software. It's easier to implement since you can use unit testing frameworks like JUnit and Jest hence can be automated.

1. Integration-testing interactions between different components or subsystems. It tests how multiple modules function as a group.

It’s importance: it can be automated using testing frameworks

1. System- testing the overall functionality and performance of a complete and fully integrated software solution. It tests whether the system meets the specified requirements and its suitable to be used by end user.
2. Acceptance- testing the software against user requirements to ensure it meets user needs.

#Part 2: Introduction to AI and Prompt Engineering

1. **Define prompt engineering and discuss its importance in interacting with AI models**.

Prompt engineering

Is a process of generating Artificial Intelligence solutions to generate desired Artificial intelligence outputs.

1. **Provide an example of a vague prompt and then improve it by making it clear, specific, and concise. Explain why the improved prompt is more effective.**

Example of a prompt: “tell me about Tanzania”

Improved prompt: “Tell me the population and location of Tanzania”

This improved prompt is more clear, specific and concise because it asks exactly what is need about Tanzania. All we want to know about Tanzania is its location and its population.