

1. Software engineering is the systematic application of engineering concepts, practices, and tools to the development and ongoing maintenance of superior software systems. It includes designing, creating, testing, implementing, and maintaining software products. Software engineering is crucial to the technology industry because it enables the development of software systems and applications that power a wide range of contemporary activities, including communication, commerce, entertainment, and healthcare.
2. A few notable developments are the creation of programming languages (such as C and Fortran), the discipline of software engineering's founding in the 1960s, the introduction of structured programming in the 1970s, and the emergence of agile approaches in the 2000s.
3. **Software Development Life Cycle (SDLC):**

There are various phases in the Software Development Life Cycle (SDLC), these phases include:

- **Requirements gathering:** Compiling and recording system and user requirements. In this stage, insights into business requirements for applicable users and systems are gathered to develop tailored solutions to the requirements.
- **Design:** Producing comprehensive and high-level designs of the user interface and software architecture tailored to users or systems requirements.
- **Implementation:** Coding and software construction following the design guidelines.
- **Testing:** Performing a range of tests to guarantee that the program satisfies functional and quality criteria.
- **Deployment:** Giving consumers or users access to the software.
- **Maintenance:** After the program is deployed, continuing support, upgrades, and improvements are provided.

4. **Comparing Waterfall and Agile Software Development Methodologies:**

Waterfall is a linear and sequential approach to software development. It consists of distinct phases such as requirements gathering, design, implementation, testing, deployment, and maintenance. Each phase must be completed before the next one begins, and there is little room for changes once a phase is finished.

Agile is an iterative and incremental approach that emphasizes flexibility, customer collaboration, and rapid delivery of small, functional pieces of software. Development is divided into small cycles called sprints, typically lasting 1-4 weeks.

Aspect	Waterfall	Agile
Approach	Linear and sequential	Iterative and incremental
Flexibility	Low	High
Customer Involvement	Limited to initial and final stages	Continuous throughout the project
Documentation	Extensive	Minimal
Testing	At the end	Continuous
Change Management	Difficult and costly	Easy and expected

5. **Roles and responsibilities:**

- **Software Developer:** In charge of creating code and putting software solutions into action.
- **Quality Assurance Engineer:** Creates and implements test programs to guarantee software quality.
- **Project Manager:** Responsible for supervising software project planning, execution, and delivery.

6. **IDEs and VCS**

- **Integrated Development Environments (IDEs):** toolkits for writing, debugging, and testing software (e.g., Visual Studio, Eclipse, IntelliJ IDEA).
- **Version Control Systems (VCS):** Software tools (e.g., Git, Subversion) for monitoring source code changes and managing team efforts.

7. **Software Development Challenges:**

- **Changing Requirements:** During the development cycle, requirements may change, which could cause scope creep and project delays.
- **Tight Deadlines:** Rushing software development might lead to a sacrifice in quality when software products are delivered on time.
- **Technical Debt:** Technical debt, which results from shortcuts or less-than-ideal solutions, can raise maintenance expenses and hinder further development.

Techniques for Overcoming Obstacles: Techniques for overcoming obstacles include prioritizing work, agile approaches, effective communication, and frequent reevaluation of project objectives and schedules.

8. **Testing and Importance of testing:**

Early fault detection and correction during the development process results in higher-quality software products. Quality control techniques include code reviews, automated testing, and continuous integration.

- **Unit testing** is the process of testing distinct software units or components.
- **Integration testing** is Investigating how various parts or subsystems interact.
- **System testing** is the process of testing the entire software system.
- **Acceptance testing** is the process of comparing the program with user requirements to make sure it satisfies their demands.

9. **Prompt engineering** is the practice of designing and refining prompts—questions or instructions—to elicit specific and accurate responses from AI models. It involves crafting inputs that guide AI systems, particularly large language models (LLMs), to produce desired outputs by understanding the context, intent, and nuances of the query

10. **Importance of Prompt Engineering**

- **Enhanced Accuracy:** Well-crafted prompts help AI models generate more accurate and relevant responses, reducing the need for extensive post-processing.
- **Efficiency:** By minimizing irrelevant or biased outputs, prompt engineering streamlines the interaction between humans and AI, saving time and resources.
- **User Satisfaction:** Effective prompts ensure that AI systems meet user expectations, improving overall user experience and satisfaction.
- **Adaptability:** It allows AI models to handle a wide range of queries, from simple to highly technical, by fine-tuning the prompts to match the task's requirements.

11. **Example of a Vague Prompt and Its Improvement**

- **Vague Prompt:** “Tell me about the weather.”

- **Improved Prompt:** “Provide a detailed weather forecast for Lagos, Nigeria, for the next three days, including temperature, precipitation, and wind speed.”

Explanation: The improved prompt is more effective because it:

- **Specifies the Location:** Indicates that the weather forecast should be for Lagos, Nigeria.
- **Defines the Time Frame:** Requests information for the next three days, making the response time-bound.
- **Details of the Required Information:** Asks for specific weather elements such as temperature, precipitation, and wind speed, ensuring the response is comprehensive and relevant.

By being clear, specific, and concise, the improved prompt reduces ambiguity and guides the AI to provide a more accurate and useful response.