1. Software Engineering a systematic application of engineering principles, methods and tools to the development and maintenance of high quality software systems.
2. It differs from traditional programming in its involvement from design, development ,testing, deployment and maintenance of software products
3. Software development life cycle is a process used to design, develop, and test high quality software systems. The phases include;

* Requirements: Gathering and documenting user needs and system requirements. Design: Creating high level designs of the software architecture and user interface.
* Implementation: Writing code and building the software according to the design specifications.
* Testing: Conducting various tests to ensure the software meets quality standards and functional requirements.
* Deployment: Releasing the software to users or customers
* Maintenance: Providing ongoing support, updates and enhancements to the software after deployment.

1. Agile vs Waterfall

**Agile:** This is an iterative and incremental approach focused on flexibility, collaboration and responding to change.

**Waterfall**: A sequential approach with distinct phases

1. Requirement engineering can be defined as the process of defining, documenting, and maintaining the requirements in the software development process.

The process includes:

* Elicitation: Gathering requirements from stakeholders through interviews, surveys, and observation.
* Analysis: Understanding and refining the requirements to ensure they are feasible and necessary.
* Specification: Documenting the requirements in a clear and concise manner, often in a requirements specification document.
* Validation: Ensuring the requirements accurately reflect the needs and are achievable.
* Management: Handling changes to requirements as the project evolves.

The importance of requirement engineering in the SDLC is that a clear requirements provide a roadmap for the entire project and ensures the final product meets user needs.

1. Modularity in software design is a design principle that divides a software system into smaller, manageable, and independent modules or components. It is easier to understand, debug, and modify individual modules.
2. The different levels of software testing involves;

* Unit Testing: Testing individual components or modules of software
* Integration Testing: Testing interactions between different components or subsystems
* System Testing: Testing the entire software system as a whole.
* Acceptance Testing: Testing the software against user requirements to ensure it meets the user’s needs.

Testing is important in software development because it ensures the software functions correctly and meets requirements and also help to identify and resolve defects before deployment.

1. Version control systems are software tools for tracking changes to source code & coordinating work among team members, (e.g Git). It is important in terms of facilitating team collaboration.
2. A software project manager oversees the planning, execution and delivery of software projects. They face challenges such as tight deadlines, technical debt.
3. Software maintenance is the modifying and updating of software products after its initial deployment to correct faults, improve performance, or adapt it to a changed environment.
4. Ethical issues that software engineers might face issues like ensuring user data is protected and not misused, developing secure software to protect against threats. Software engineers can ensure they adhere to ethical standards in their work by being open about the development process and potential issues.