Name: Adeyeye Yusuf Olatunji

Matric Number: HCSF/23/0027

Course code: Com324

Question 1:What is software engineering?

Answer:

Software Engineering is the application of engineering principles and techniques to the design, development, testing, maintenance, and management of software systems.

Question 2 : List the needs for Software engineering

Answer:

1. Programming languages and frameworks

2. Development tools and IDEs

3. Database management systems

4. Operating systems and platforms

5. Web technologies and protocols

6. Data structures and algorithms

7. Software development methodologies

8. Testing and debugging tools

Question 3: Explain the 5 Major steps in a software development life cycle

Step 1: Requirements Gathering and Analysis

- Collect and document customer requirements

- Identify project scope, goals, and deliverables

- Analyze and prioritize requirements

- Create a detailed requirements specification document

Step 2: Design

- Create a detailed design of the software architecture

- Develop a user interface (UI) and user experience (UX) design

- Define the system’s components, interfaces, and data models

- Create a design specification document

Step 3: Implementation (Coding)

- Write the code for the software application

- Develop and integrate individual components

- Conduct unit testing and integration testing

- Ensure coding standards and best practices are followed

Step 4: Testing and Quality Assurance

- Plan and execute various testing types (e.g., functional, performance, security)

- Identify and report defects or bugs

- Conduct quality assurance activities (e.g., code reviews, audits)

- Ensure software meets requirements and quality standards

Step 5: Deployment and Maintenance

Question 4: List 5 tools needed at each step in question 3.

Answer: Here are 5 tools needed at each step of the software development life cycle:

Step 1: Requirements Gathering and Analysis

1. Microsoft Office (Word, Excel, PowerPoint) for documentation

2. Jira or Trello for project management and tracking

3. MindMeister or Coggle for mind mapping and brainstorming

4. SurveyMonkey or Google Forms for stakeholder surveys

5. Lucidchart or Visio for diagramming and flowcharts

Step 2: Design

1. Sketch or Figma for user interface (UI) design

2. Adobe XD or InVision for user experience (UX) design

3. Microsoft Visio or Lucidchart for system architecture diagrams

4. Axure or Justinmind for prototyping and wireframing

5. Styleguidist or Bit for design system management

Step 3: Implementation (Coding)

1. Visual Studio Code or IntelliJ IDEA for coding

2. GitHub or GitLab for version control and collaboration

3. Junit or PyUnit for unit testing

4. Docker or Kubernetes for containerization and orchestration

5. Postman or Swagger for API testing and documentation

Step 4: Testing and Quality Assurance

1. Selenium or Appium for automation testing

2. JIRA or TestRail for test case management

3. Jenkins or Travis CI for continuous integration and deployment

4. SonarQube or CodeCoverage for code analysis and quality metrics

5. TestComplete or Ranorex for functional testing

Step 5: Deployment and Maintenance

1. Jenkins or CircleCI for continuous deployment

2. Docker or Kubernetes for containerization and orchestration

3. Puppet or Ansible for infrastructure automation

4. Nagios or Prometheus for monitoring and alerting

5. GitOps or ArgoCD for continuous delivery and deployment

Question 5: List 5 Software development methodologies

Answer:

1. Agile Methodology:

- Emphasizes flexibility, collaboration, and rapid delivery

- Iterative and incremental development

- Focus on customer satisfaction and continuous improvement

- Examples: Scrum, Kanban, Lean Software Development

2. Waterfall Methodology:

- Linear and sequential approach

- Emphasizes predictability and stability

- Phases: requirements, design, implementation, testing, deployment, maintenance

- Suitable for projects with well-defined requirements

3. DevOps Methodology:

- Combines development and operations teams

- Emphasizes continuous integration, delivery, and deployment

- Focus on automation, testing, and monitoring

- Examples: Continuous Integration/Continuous Deployment (CI/CD), Continuous Monitoring

4. Lean Software Development Methodology:

- Emphasizes eliminating waste and maximizing value

- Focus on continuous improvement and delivery

- Principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible

- Examples: Kanban, Scrum, Lean Startup

5. Extreme Programming (XP) Methodology:

- Emphasizes technical practices and teamwork

- Focus on delivering high-quality software quickly

- Practices: pair programming, test-driven development, continuous integration, refactoring

- Suitable for projects with rapidly changing requirements

Question 6: Draw and explain the Waterfall Model, V model and Spiral Model

Answer: The Waterfall Model is a linear, sequential software development methodology where each phase is completed before moving on to the next one.

The V Model Is a software development process that is an extension of the Waterfall Model. It is called the V Model because the process is shaped like a V, with the left side representing the development activities and the right side representing the testing activities.

The Spiral Model Is a software development process that combines elements of both iterative and waterfall models. It is called the Spiral Model because the process is shaped like a spiral, with each loop representing a phase of development.

Question 7: Explain Functional and Non-Functional requirements of a software, give examples

Answer:

Functional Requirements (FRs) define what the software should do, describing its functionality, features, and behaviors. They specify the software’s interactions with users, inputs, and outputs.

Examples:

1. User authentication: The system shall allow users to log in using their username and password.

2. Data storage: The system shall store customer information, including name, address, and phone number.

3. Payment processing: The system shall process credit card payments securely.

4. Search functionality: The system shall allow users to search for products by keyword.

5. Reporting: The system shall generate monthly sales reports.

Question 8:Explain 3 types of Software testing

Answer: Unit Testing

Unit testing involves testing individual software components or units, such as functions, methods, or classes, to ensure they work correctly in isolation.

Purpose:

- Verify that each unit performs its intended function

- Catch bugs and errors early in the development cycle

- Ensure units are stable and reliable before integrating them into the larger system

Example:

* Testing a login function to ensure it authenticates users correctly

ii. Integration Testing:

Integration testing involves testing how different software components or units work together to ensure they integrate seamlessly.

Purpose:

- Verify that units work together as expected

- Identify issues with data flow, interfaces, and dependencies

- Ensure the system behaves as expected when components interact

Example:

* Testing how the login function integrates with the user database and authentication system

iii. System Testing:

System testing involves testing the entire software system, from end to end, to ensure it meets requirements and works as expected in real-world scenarios.

Purpose:

- Verify that the system meets functional and non-functional requirements

- Identify issues with system performance, security, and usability

- Ensure the system is stable, reliable, and meets user expectations

Example:

Testing an e-commerce system to ensure it handles user registration, login, product browsing, and checkout processes correctly.

Question 9:What is a software CASE tool?

Answer: A software CASE (Computer-Aided Software Engineering) tool is a program that supports various activities during the software development process, such as:

1. Requirements gathering and management

2. Design and modeling

3. Code generation and debugging

4. Testing and quality assurance

5. Project management and configuration control

CASE tools provide automated support for software development, improving productivity, quality, and consistency. They can be used for:

1. Diagramming and modeling (e.g., UML, data flow diagrams)

2. Code generation and reverse engineering

3. Requirements management and traceability

4. Project planning and scheduling

5. Version control and configuration management

6. Testing and defect tracking

Examples of CASE tools include:

1. IBM Rational Rose (UML modeling)

2. Microsoft Visio (diagramming and modeling)

3. Sparx Systems Enterprise Architect (UML modeling)

4. Atlassian Jira (project management and issue tracking)

5. GitHub (version control and collaboration)

CASE tools can be categorized into three main types:

1. Upper CASE (UCASE): Focuses on front-end activities like requirements gathering and design.

2. Lower CASE (LCASE): Focuses on back-end activities like code generation and testing.

3. Integrated CASE (ICASE): Combines UCASE and LCASE tools to support the entire software development life cycle.

Question 10: Mention 5 types of software CASE tools

Answer: Here are 5 types of software CASE (Computer-Aided Software Engineering) tools:

1. \*Requirements Management Tools\*:

Examples:

- IBM Rational DOORS

- Microsoft Team Foundation Server (TFS)

- Jama Connect

Purpose:

- Manage and track software requirements

- Ensure requirements traceability

- Collaborate with stakeholders

1. \*Design and Modeling Tools\*:

Examples:

- Sparx Systems Enterprise Architect

- IBM Rational Rose

- Microsoft Visio

Purpose:

- Create UML, data flow, and other diagrams

- Model software architecture and design

- Generate code from models

1. \*Code Generation and Debugging Tools\*:

Examples:

- Eclipse

- Visual Studio

- IntelliJ IDEA

Purpose:

- Generate code from models or specifications

- Debug and test software

- Refactor and optimize code

1. \*Project Management and Version Control Tools\*:

Examples:

- Atlassian Jira

- GitHub

- GitLab

Purpose:

- Manage projects and track progress

- Version control and collaboration

- Issue tracking and defect management

1. \*Testing and Quality Assurance Tools\*:

Examples:

- Selenium

- Junit

- TestRail

Purpose:

- Automate testing and validation

- Ensure software quality and reliability

- Track defects and issues

Other types of CASE tools include:

- Configuration Management Tools (e.g., Ansible)

- Documentation Tools (e.g., Doxygen)

- Reverse Engineering Tools (e.g., IBM Rational Rose)

- Integration Tools (e.g., Apache Kafka)