Here are the answers to the multiple-choice questions and explanations for the open-ended questions:

Multiple Choice Questions and Answers

1. **Which of the following is a non-functional requirement?**
    - A) The system shall display the user's profile after login
    - B) **The system shall process 10,000 transactions per second**
    - C) The system shall allow users to reset their password
    - D) The system shall support user authentication
2. **Which of the following is NOT an advantage of Agile methodology?**
    - A) Increased flexibility
    - B) Faster delivery of working software
    - C) **Extensive upfront documentation**
    - D) Better responsiveness to change
3. **In software testing, unit testing is concerned with:**
    - A) Testing the entire system
    - B) **Testing individual modules or components**
    - C) Testing the system's integration with external systems
    - D) Testing the system's performance under load
4. **Which software development model is also called the "linear sequential model"?**
    - A) Spiral Model
    - B) **Waterfall Model**
    - C) Agile Model
    - D) V-Model
5. **Which of the following is an example of white-box testing?**
    - A) Regression testing
    - B) Equivalence partitioning
    - C) **Code coverage analysis**
    - D) Usability testing

Open-Ended Questions and Explanations

6. **What is the difference between functional and non-functional requirements?**
    - Functional requirements define what a system is supposed to do, such as specific behaviors or functions. Non-functional requirements describe how a system performs a function (e.g., performance, usability, reliability).
7. **How does the waterfall model differ from the agile model?**
    - The waterfall model is a linear and sequential approach to software development, where each phase must be completed before the next begins. Agile is iterative and incremental, allowing for more flexibility and changes throughout the development process.
8. **What is the purpose of software testing, and what are its main levels?**
    - Software testing aims to identify defects and ensure that the software meets the required standards and functions correctly. The main levels include unit testing, integration testing, system testing, and acceptance testing.
9. **What are the characteristics of good software?**
    - Good software is typically reliable, efficient, maintainable, scalable, and user-friendly. It should meet the user's requirements and be delivered on time and within budget.
10. **How do version control systems help in software development?**
    - Version control systems track changes to code, allowing multiple developers to work on a project simultaneously. They help manage code versions, facilitate collaboration, and enable rollback to previous states if needed.
11. **What is the role of a software architect in a development team?**
    - A software architect designs the high-level structure of a software system, making key decisions about the architecture and ensuring that the system meets both technical and business requirements.
12. **What is the difference between verification and validation?**
    - Verification ensures the product is built correctly according to specifications, while validation ensures the right product is built and meets the user's needs and expectations.
13. **What are software design patterns, and why are they important?**
    - Design patterns are reusable solutions to common software design problems. They promote best practices, improve code readability, and facilitate communication among developers.
14. **What are the main challenges of software maintenance?**
    - Challenges include managing code complexity, ensuring compatibility with new systems or technologies, addressing bugs, and incorporating new features without affecting existing functionality.
15. **How does continuous integration (CI) improve software quality?**
    - CI involves frequently integrating code changes into a shared repository, with automated testing to detect issues early. This leads to faster feedback, reduced integration problems, and higher-quality software.
16. **What is technical debt, and how can it affect a project?**

- Technical debt refers to the implied cost of additional rework caused by choosing a quick or easy solution now instead of a better approach. It can lead to increased maintenance costs and reduced development speed over time.

17. **What are the advantages of using microservices over monolithic architecture?**
    - Microservices offer greater flexibility, scalability, and fault tolerance. They allow independent deployment and scaling of services, which can improve development speed and reliability.
18. **What is the purpose of requirement elicitation?**
    - Requirement elicitation involves gathering requirements from stakeholders to understand their needs and expectations. It ensures that the software will meet user requirements and business goals.
19. **How does user-centered design influence software development?**
    - User-centered design focuses on the needs and limitations of end-users at every stage of the development process,