**2318083 – SW Metrics Exam - short questions**

1. Downtime: Downtime refers to the period when a system is not operational, typically due to system failures, maintenance, or upgrades. This metric is crucial for assessing the reliability and availability of software systems, impacting user satisfaction and operational continuity.
2. Reliability: Reliability measures the probability that a software system will perform without failure under specified conditions for a designated duration. It is critical for establishing user trust and is quantified using metrics like Mean Time Between Failures (MTBF) and failure rate.
3. Product Measures: Product measures focus on the attributes of the software product itself, including code complexity, size, performance efficiency, and feature set. These metrics are used to assess product quality, guide development efforts, and optimize product design.
4. Stages of Design: The stages of design in software development typically include requirement gathering, architectural design, detailed design, and implementation. Each stage progressively refines the system specifications and prepares the software for the subsequent integration and testing phases.
5. Design Quality: Design quality metrics assess how well the software design aligns with the specified requirements and best practices. High design quality ensures easier maintenance, better performance, and increased user satisfaction through robust and efficient architecture.
6. Benefit of Code Coverage Measurement: Measuring code coverage helps ensure that sufficient test cases have been written, covering various parts of the program. It is vital for identifying untested parts of a code base, reducing the likelihood of bugs and improving software reliability and maintainability.
7. Change Request Metrics: Change Request Metrics track and analyze the nature, volume, and impact of requested changes in a software project. These metrics help in managing change control processes, assessing the implications of modifications, and ensuring project stability and quality.
8. Maintenance Quality Metrics: These metrics assess the effectiveness of software maintenance activities by tracking parameters like frequency of failures, mean time to repair, and response time for maintenance issues. Effective maintenance quality metrics lead to improved system reliability, reduced downtime, and lower long-term costs.
9. Identify the Benefits of Applying the Metrics: Applying metrics in software development helps quantify various aspects of system performance, quality, and efficiency. They provide critical data that supports strategic decision-making, enhances process improvements, and benchmarks against industry standards.
10. Failure Characteristics: Understanding failure characteristics involves analyzing how failures occur, the conditions that lead to failures, and their consequences. This analysis is crucial for improving software design, enhancing fault tolerance, and developing strategies to mitigate adverse effects on users.