

## SW Exam

1) Code coverage can help in enhancing your development process by offering feedback, valuable insights, and direction. Utilizing code coverage to monitor software quality allows you to gauge project advancement, pinpoint potential problems, and effectively tackle challenges as they arise.

2) Change request metrics are measurements used to track and evaluate various aspects of change requests within a project or software development lifecycle. These metrics help in understanding the volume, nature, impact, and efficiency of change requests.

3) Maintenance quality metrics are measurements used to assess the effectiveness, efficiency, and overall quality of maintenance activities within a software system or project. These metrics focus on evaluating the performance of maintenance tasks, bug fixes, enhancements, and updates after the initial development phase.

4) Performance Evaluation: Metrics provide quantitative data to assess performance, helping teams gauge their efficiency, productivity, and effectiveness in handling tasks and meeting goals.

Quality Improvement: Metrics highlight areas of improvement by identifying patterns, bottlenecks, or recurring issues, allowing teams to focus efforts on enhancing software quality.

Decision Making: Data-driven insights from metrics aid in informed decision-making, guiding resource allocation, prioritization of tasks, and identifying critical areas for investment or improvement.

Risk Management: Metrics help in identifying and mitigating risks by flagging potential issues early, enabling proactive measures to minimize their impact on project timelines and quality.

Process Optimization: Metrics facilitate process optimization by identifying inefficiencies or areas where workflows can be streamlined, leading to improved productivity and reduced waste.

Communication and Transparency: Metrics provide a common language for stakeholders, enabling clearer communication about project status, progress, and challenges, fostering transparency and trust.

5) Failure characteristics in software metrics refer to the patterns or attributes associated with software failures or defects. These characteristics help in understanding the nature, causes, and impacts of failures within a software system.

6) These metrics serve as measures to quantify how often and severely downtime events affect systems. They include indicators like availability, mean time between failures (MTBF), mean time to repair (MTTR), recovery point objective (RPO), and recovery time objective (RTO).

7) Reliability in software metrics refers to the consistency and dependability of software in delivering intended functionalities and performance under specific conditions over time. It signifies the software's ability to operate without failure and maintain its functionality within expected parameters.

8) Product measures in software metrics refer to the quantitative assessments used to evaluate specific attributes or characteristics of the software product itself. These measures focus on aspects related to the product's structure, complexity, size, functionality, and quality.

9) In the context of software development, design stages encompass various phases that focus on planning, structuring, and defining the architecture and functionalities of a software system. Each stage benefits from specific metrics that aid in evaluating the progress, quality, and potential risks associated with the design process. These metrics help teams make informed decisions, identify potential issues early, and ensure that the software is being developed according to predefined standards and requirements.

10) Design quality in software metrics refers to the characteristics and attributes of the software design that contribute to its effectiveness, efficiency, maintainability, and overall excellence. It encompasses various aspects of the design that impact the software's reliability, scalability, performance, and ease of modification.