

Q.1: ① What are reviews and audits, why it is necessary

**Review:** A review is a process of examining software artifacts such as requirements, design documents, code and test plans to identify defects, potential issues, or areas for improvement. Reviews are usually conducted by peers, managers or other stakeholders and the goal is to improve the quality and maintainability of the software.

**Audit:** An audit is a more formal and structured evaluation of software processes, procedures, and products. An audit is typically conducted by an independent third party and involves a comprehensive examination of software artifacts and processes to assess compliance with established standards, regulations, or best practices.

It is necessary in software engineering to ensure that software products are of high quality and meet the requirements of stakeholders.



Ques 2 How is quality measure in software engineering? List the criteria for good software.

Ans: To measure quality we have many attribute of software such as;

- |                  |                   |                    |
|------------------|-------------------|--------------------|
| 1) Functionality | 2) Reliability    | 3) Maintainability |
| 4) Efficiency    | 5) Usability      | 6) Portability     |
| 7) Consistent    | 8) Cost-effective |                    |
| 9) Scalable      | 10) Comprehensive |                    |
| 11) Reliable     | 12) Compatible    |                    |
| 13) Sensitive    | 14) Traceable     |                    |
| 15) Flexible     | 16) Secure        |                    |

and many more

Ques 3 What are characteristics of software quality? What is the structure of SQA plan.

⇒ Characteristics of software quality can vary depending the perspective and requirements of the software, some of them are,

- |                  |                    |                |
|------------------|--------------------|----------------|
| 1) Functionality | 2) Reliability     | 3) Usability   |
| 4) Efficiency    | 5) Maintainability | 6) Portability |
| 7) Security      |                    |                |

⇒ Structure of SQA plan can vary depending on the specific requirements of the software project, but generally ~~also~~ includes the following sections



- 1) Introduction - overview of SQA plan and the project.
- 2) Quality objective: This section outlines the specific quality objectives for the software project, e.g. - reliability, usability and security.
- 3) Roles and Responsibilities: section defines the roles and responsibility of the SQA team and other stakeholders.
- 4) Documentation: including coding standards, design documents and test plans.
- 5) Testing: includes test plans, test cases and test results.
- 6) Configuration Management: including version control, release management and change management.
- 7) Tools and Techniques: automated testing tools, code review tools and bug tracking tools.
- 8) Training and Education: outlines training and education requirements for the SQA team and other stakeholders.
- 9) Risk Management
- 10) SQA Metrics.



Ques: (1) Can we use reliability models for quality management and tracking? Why?

⇒ Yes, reliability model can be used for quality management and tracking. Reliability models are used to analyze the reliability of a product or system, which is an important aspect of quality management.

Reliability models can help in tracking the performance of a product or system over time, identifying potential areas of failure or weakness, and making improvements to enhance its reliability and overall quality. By using reliability models, organizations can proactively identify and address potential issues before they become major problems that could impact customer satisfaction.



Ques: ⑤ Prepare a survey on "Software Quality" for your own project. It should include 10 different questions that can define the good quality products.

⇒ Since, I am working with my team in e-commerce project. On the basis of that I prepared 10 different questions.

- ⇒
- ① How often do our experience crashes or errors when using the software?
  - ② How easy is it to navigate through the software's user interface?
  - ③ How quickly does the software respond to user input?
  - ④ How frequently does the software receive updates or bug fixes?
  - ⑤ How secure is the software in terms of protecting user data?
  - ⑥ How customizable is the software to fit specific user needs?
  - ⑦ How well does the software fit specific user needs?
  - ⑧ How clear and concise is the documentation or help resources provided for the software?
  - ⑨ How well does the software handle high traffic or usage levels?
  - ⑩ How reliable is the customer support for the software in case of issues or questions?