Date __/__/____

Saathi

Q1 Explain Software Quality Asswrance techniques in detail. What are the advantages and disadvantages of Software Quality Asswrance?

Ans Software Quality Asswrance (SQA) is simply a set of activities which are used to ensure the quality of a Software. It ensures that processes, procedures as well as standards are suitable for the project and implemented correctly.

Software Quality Asswrance works parallel to development of software. It focuses on improving the process of development of software so that problems can be prevented before they become a major issue.

Generally, the quality of the software is verified by the third party organisation like 'International Standard Organisation.'

Date __/__/____

Saathi

- Software Quality Assurance (SQA) focuses on a software's ;

(i) Portability
(ii) Usability
(iii) Reusability
(iv) Correctness
(v) Maintainability
(vi) Error Control

- Software Quality Assurance has :-

(i) A quality management approach

(ii) Formal technical reviews

(iii) Multi-testing strategy

(iv) Effective software engineering technology

(v) Measurement and reporting mechanism

Date __/__/____

Saathi

- **Various Software Quality Assurance (SQA) techinques are :-**

1. **Reviewing :-**

In reviewing, a meeting is held by both internal and external stakeholders to review the whole project. The whole software is analysed and if an issue is found, it is distinguished whether it is a testing, development, requirement or design issue.
The main objective is to measure the quality of a software and ensure whether it meets the customer expectations or not.

2. **Auditing :-**

In auditing, the whole work product and all the data are inspected by the stakeholders to check whether

it follows the standard processes or not.

3. Functional Testing :-

In functional testing, the functionality of the whole software is tested whether it is functioning as expected or not. It checks, "what the system does" without knowing "how the system works". It is like the black-box testing of an application in which the user knows the expected output without knowing how it is produced.

4. Standardization :-

It assures that everything in the software is standardized, i.e. it follows all the standards in

documentation, development, quality control, etc. It reduces ambiguity and hence improves the quality of software.

5. Code Inspection :-

Code Inspection is one of the most formal kinds of review with the main objective of finding defects in the code and highlighting any issues in the code inspection is led by a trained moderator rather than the author of the code.
The meeting has a proper entry and exit criteria. Users must need complete preparation before the meeting in order to have complete knowledge of documents and all before raising their points.

Date __/__/____

Saathi

6. Walkthroughs :-

Software Walkthrough is a kind of informal process and, usually it is initiated by the author to read the document or code and the peer member write down their submission or errors in it and submit them. This process is not formally documented like Code Inspection and a moderator is not necessary. The main objective is to know the status of code completed to date and collecting suggestions from peers to improve the quality of software.

7. Stress Testing :-

Stress Testing is done to check how the system works under heavy load. This testing plays an important

role in software quality as in e-commerce applications, stress and load testing are done properly in order to test the capacity of the software. (How many maximum numbers of users can access an application at a time.)

8. Design Inspection

Design Inspection is done to check the various areas of software using the checklist like functional and interface design, conventions, general requirements and design, requirement traceability, logic, coupling and cohesion.

9. Simulation:-

A simulation is a tool that models a real-life situation in order to virtually examine the behaviour of the system under study.

Saathi

Date ___ / ___ / _____

- ## Advantages of Software Quality Assurance:-

1. ### Increases client's confidence :

Proper quality check at different levels of software, like review, inspection, auditing, etc. and with the involvement of both internal and external stakeholder increases the confidence of clients. The submission of the reports of defect and requirement metrics regularly also helps a lot in assuring the client that the work is being done on time.

2. ### Saves Money :

Defects found in the early stage either in requirement gathering, or code testing are easy and cost-effective to mitigate. Proper SQA

done at several levels helps to reduce those risks as maximum defects have been uncovered and resolved in early stages and hence saves money to fix the faulty software.

3. Promotes productivity and efficiency:-

Defects found and fixed in early stage of the development of a single module allows everyone to work efficiently and in a productive manner rather than be burdened by multiple bugs at once after the completion of the whole software.

4. Reduces end time client conflicts:-

There are many cases of disagreement between client and organisations later on regarding changes in

requirements, time and budget resulting in cancellation of the project and money loss. In SQA, everything is fixed at the starting of the project and documented properly without any ambiguity so that no conflicts would arise.

- **Disadvantages of Software Quality Assurance :-**

1. **Sometimes difficult to implement :-**

SQA defines all the activities and actions that should be taken at each step of software development in a very detailed manner. Therefore, sometimes it becomes very difficult to implement every single activity and process in development.

2. Time Consuming :-

Implementing each action in SQA is very time consuming and sometimes it wastes more time in documentation and meetings rather than working on the actual development and testing of the software.

3. High Cost :-

Though implementing SQA reduces the cost of fixing bugs, but for small projects, with a low budget, it is very difficult to implement SQA. For small projects, hiring the whole team of people and implementing SQA causes a drastic increase in the cost of a project.

Saathi

Date ___ / ___ / _____

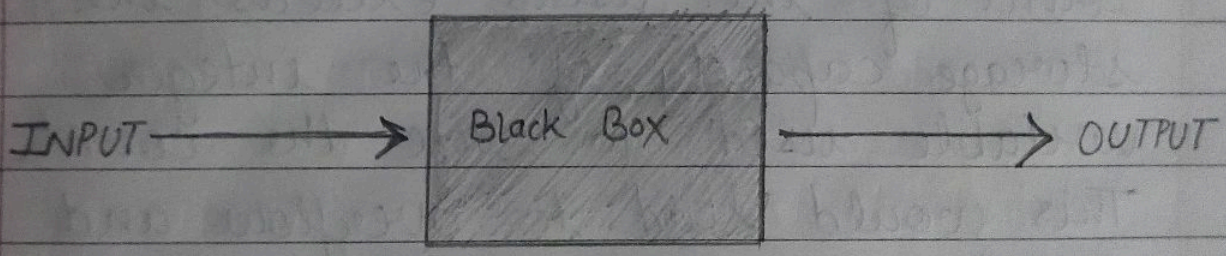
Q2: Give atleast three examples in which black-box testing might give the impression that "everything's OK", while white-box tests might uncover an error. Do the same for white-box testing.

Ans:- 'White-box testing' techniques analyse the ~~st~~ internal structure : used data structures, internal design, code structure, and the working of the software rather than just the functionality of the software. It is also called as Transparent testing or open-box testing.
The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

Date __/__/____

Saathi

'Black-box testing' is a technique of software testing which examines the functionality of software without peering into its internal structure or code. In this method, the tester selects a function and gives input value to examine its functionality and checks whether the function is giving expected output or not.



- Examples in which black-box testing might give the impression that everything is okay while white box tests might uncover an error :—

Saathi

Date ___/___/_____

1. **Login Counter :**

Suppose the software that is being tested stores the number of users that have logged in. To achieve this, it used a 16-bit integer to store the count of the users. Functionally, there is nothing wrong with the software and Black-box testing might give the impression that everything is okay until, the count of the users exceeds the storage capacity of the integer variable used to store the count. This would lead to overflow and faulty behaviour of the software. If the integer is unsigned, the count would revert to 0 and for a signed integer, the count would show negative value which is impossible. This error can easily be detected through looping in white-box testing.

Date ___ / ___ / ___

Saathi

2. Miscalculations :-

Take for instance the following calculation:-

sqrt(16) = 4 ; sqrt(4) = 2 ; sqrt(2) = 1.412 .....
..... and so on.

Now if you reverse the order of the execution :-

square(1.412) = 2, square(2) = 4 ;
square(4) = 16.

Functionally, this is right and you finally arrive at the same value (16) with which you started. But if you subtract 16 (integer) from the result, you will get a precision error. This happens because floating point arithm-etics is complicated to perform on computers and result was not exactly 16.00 but something like 16.000342.

Date ___ / ___ / _____

It may not seem like a big problem but think what a disaster it could be for a banking application, or a scientific research that requires high precision. This would have been easily detected in white-box testing.

3. Never-ending loops :-

When there is a never ending loop triggered by a condition, for example, suppose when a vehicle reaches a place, a condition is satisfied and a loop is triggered which keeps instantiating objects of a class. Functionally, there is nothing wrong with this but this could potentially be a disaster in terms of memory leaks.

- Examples in which white-box testing might give the impression that "everything is okay" while black-box testing might uncover an errors-

1. Interface Functionality :-

Black-box testing is really useful when you are testing the user interface of your software. Suppose you are testing the Graphical user Interface (GUI) of the software. White-box testing may tell you that every component is working correctly and producing accurate result, but it will not tell you if there is some issue with the user interface of the software. Only black-box testing will uncover the issues related to the user interface such as unaligned GUI elements and components.

Date __/__/____

2. Testing a range of inputs :-

In Black-Box testing, the software is tested on a range of inputs. Different classes of inputs are provided to the software and the behaviour of the system is observed. Any issues or errors that may be caused due to invalid input to the software, can only be uncovered by Black-box testing. For example; Division by zero error can be easily uncovered by Black-Box testing whereas white-Box testing may fail to uncover it.

3. Testing Performance and behaviours :-

White-box testing doesnot concern itself with performance issues and behaviours of the software. Black-

Date ___/___/_____

Saathi

Box testing is used to benchmark the performance of the software on very large sizes of inputs. It also uncovers various performance issues that the system might have.
Black-box testing is also used to detect issues and errors in the software behaviour.