**List five of software qualities with examples.**

Some of the software qualities are:

1. Functionality: The software must perform the functions that it is meant to do. Not only this but also it must do it accurately. The different functions must be properly integrated and should be interoperable. Also, it should be secure. For example, a school management system should be able to perform the following functions:
   1. Provide detailed information about students such as attendance, assignments, grades, prizes, etc.
   2. Provide detailed information about teachers such as timetable, attendance, and so on.
   3. Provide access to parents about their children's details.
   4. Submission of assignments online, and so on.

Functionality is one of the most important qualities of software because if it cannot perform the functions it is supposed to do, nothing else matters in the software.

1. Reliability: In testing, reliability refers to the degree of consistency of a measure. If the same procedure is performed multiple times, the similarity in the results determines the reliability. Similarly, if multiple users perform the same actions, they should get similar results. In software testing, reliability is a measure which includes the maturity, fault tolerance and recoverability. Maturity is related to the frequency of failure. Similarly, fault tolerance means the capacity of the software to withstand and recover from a failure. Recoverability is the ability to return to the fully operational state after a failure. For example, many people try to fill the form of driving license through the online registration system. For some people, it becomes successful, however, for others it fails. Sometimes, the whole website fails because it cannot handle all the requests. After the failure, it is not operational for 1 hour. This system cannot be considered reliable because it doesn’t have a proper mechanism to inform users that it cannot handle further request without going down. Another thing is that it takes a lot of time to recover.
2. Efficiency: Efficiency refers to the consumption of time and resources the software takes. For example, if a user wants to fill a long form, how long does it take to completely handle the request and successfully add a record. Doing so, how much resources such as CPU memory, disk, network, etc. it uses. That is to say, it determines how well the software performs.
3. Usability: Usability means the ease of use of the software in different environments. For instance, if the employees aren’t having a hard time using the software, then it’s usable. Similarly, if the software can be used in different platforms (i.e. Is interoperable), then it’s usable. Moreover, the time and effort it takes to learn to use the software also determine the usability of the software. For example, Bhatbhateni uses software provided by IMS. So, the company must teach its employees how to use the software. If they can learn it easily, and use it without any problems, then it’s usable.
4. Maintainability: There is no such thing called bug-free in software. Similarly, there needs to be incorporated some changes during the course of its use due to the technological changes, requirements and policy changes, etc. Maintainability refers to the ability to identify the bugs in the software. Also, it is the ability to perform some changes without breaking the system. Similarly, the software and its components should be test-ready. For example, browsers add some features in their major releases. Those features need to be tested, without breaking the browser. Similarly, the features should be easily added.

**Compare and differentiate between QA and QC (activities, roles in STLC). Explain in your own words with examples.**

Quality assurance is the process of maintaining quality during the process of the software development lifecycle. On the other hand, quality control is the process of actually validating the product developed. In the QA process, the team needs to be active from the starting of the software engineering process (i.e. planning). However, it is not the same for the QC. In STLC, the role of QC comes at last. The differences are shown below in the table.

|  |  |
| --- | --- |
| **Quality Assurance** | **Quality Control** |
| It is related to ensuring quality during the process of developing software. | It is related to testing the actual software developed. |
| It is a proactive process. | It is a reactive process. |
| Everyone in the team is responsible for this. | Generally, the testers perform this process. |
| Its goal is to prevent defects in the final product. | Its goal is to remove defects existed in the final product. |
| It is a managerial tool. | It is a corrective tool. |
| It is process-oriented. | It is product-oriented. |

For example, the verification comes under QA, whereas the validation comes under QC.

**Pre-reads: Principles of software testing.**

1. Testing shows the presence of defects.
2. Exhaustive testing is not possible.
3. Early testing.
4. Defect clustering.
5. Pesticide paradox.
6. Testing is context-dependent.
7. Absence of errors fallacy.

**Compare Agile with Waterfall and V models.**

|  |  |  |
| --- | --- | --- |
| **Agile** | **Waterfall** | **V-shaped** |
| The SDLC process is repeated in different iterations. | The SDLC process is performed in a single routine. | The development cycle is similar to the waterfall model. |
| Testing is performed in each stage of each iteration. | Testing is performed in a separate stage of the development lifecycle. | Testing is performed in the verification phase and validation phase of the development lifecycle. |
| Errors can be easily identified in the early stages. | Errors are found in testing stage only. | Errors are found early if tested properly. |
| The cost of fixing the bugs is less. | The cost of fixing the bugs is high. | The cost of fixing the bugs is less if the test is performed properly in the early stages. |
| The continuously changing requirements are addressed by this model. | The predefined and fixed requirements provide an advantage in this model. | The requirements are similar to the waterfall model. |