1. List 5 of Software qualities with Examples

Software Qualities are listed below:

* Functionality :

Functionality is the essential purpose of any product or service.Functionality is expressed as a totality of essential functions that the software product provides. The more functions a product has then the more complicated it becomes to define its functionality. Example: Atm Machine:

To withdraw required amount.

To check the current balance.

* Reliability :

Once a software system is functioning, as specified, and delivered the reliability characteristic defines the capability of the system to maintain its service provision under defined conditions for defined periods of time. One aspect of this characteristic is fault tolerance that is the ability of a system to withstand component. They determine the maximum allowed failure rate of the software system, and can refer to the entire system or to one or more of its separate functions.

For example if the network goes down for 20 seconds then comes back the system should be able to recover and continue functioning.

* Usability:

Usability only exists with regard to functionality and refers to the ease of use for a given function. Usability deals with the staff resources needed to train a new employee and to operate the software system.The ability to learn how to use a system is also a major sub characteristic of usability.

For example a function of an ATM machine is used to dispense cash as requested. Placing common amounts on the screen for selection, i.e.1000, 2000,5000 etc, does not impact the function of the ATM but addresses the Usability of the function.

* Efficiency:

It deals with the hardware resources needed to perform the different functions of the software system. It includes processing capabilities (given in MHz), its storage capacity (given in MB or GB) and the data communication capability (given in MBPS or GBPS).

For example: if the machine or system use less resources to complete the same task and give response quickly then the system is efficient.

* Maintainability:

It refers to the ease with which modifications can be made in a software system to extend its functionality, improve its performance, or correct errors.It considers the efforts that will be needed by users and maintenance personnel to identify the reasons for software failures, to correct the failures, and to verify the success of the corrections.Maintainability is impacted by code readability or complexity as well as modularization.

Anything that helps with identifying the cause of a fault and then fixing the fault is the concern of maintainability.

For example, if the system can adapt to any changes and can fix any bug if arises then system can be maintainable.

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

Quality Assurance:

QA is set of activities for ensuring quality in the process by which products are developed. QA aims to prevent defects with a focus on the process used to make the product. It is a proactive quality process. The goal of QA is to improve development and test processes so that defects do not arise when the product is being developed.Eg: Verification

Quality Control:

QC is set of activities for ensuring quality in product. QC aims to identify (and correct) defects in the finished product. Quality control, therefore, is a reactive process. The goal of QC is to identify defects after a product is developed and before it's released.Eg:Validation

Difference between

|  |  |
| --- | --- |
| QA | QC |
| The set of activities which takes into confidence that the quality of the process by which the product is developed is up to the mark is known as Quality Assurance. | The set of procedures which guarantees the fitness of the product is known as Quality Control. |
| Quality Assurance is proactive which is just opposite of the Quality Control because quality assurance takes place at the time of development of the product | Quality control takes place after the final product is produced. |
| QA is more process-oriented | QC is product-oriented. |
| Examples of QA include process definition and implementation, training, audits and selection of tools. | Examples of QC include technical reviews, software testing and code inspections. |

1. Principle of Software testing

There are seven principles in software testing:

1. Testing shows 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

1. Compare Agile with Waterfall and V- model

|  |  |  |  |
| --- | --- | --- | --- |
|  | Agile | Waterfall | V-model |
| Quality | Quality focus on all aspects of SDLC at any given time | Quality focus changes from Analysis to design to code to test | Quality focus on  verification and validation phase |
| Quality Control | Early detection and fixing in each sprint followed by stabilization | Detection and fixing during system regression testing at last phase | Every single phase in the development cycle, there is a directly associated testing phase. |
| Testing | Only at the end | Testing at every iteration | Start with first stage itself |
| Quality of Development | Better Quality | Average Quality | Better Quality than waterfall |
| Risk | Low project schedule risk | High project schedule risk | Low project schedule risk |
| Planning Scale | Long-term | Short-term | Short-term |
| Customer Availability | Available throughout project | Requires customer involvement only at requirement analysis phase | Does Not interact much,involvement only at requirement analysis phase |