Sl. no.	Attribute	Significance
1	Complexity	Software complexity describes a specific set of characteristics that focus on how our piece of code or software interacts with other pieces of code.
2	Efficiency	Software Efficiency determines the amount of code and testing resources required by a program to perform a particular function.
3	Interoperability	It is the ability of two or more systems to communicate or exchange data easily and to use the data that has been exchanged.
4	Learnability	Learnability is a quality of software products and interfaces that allows users to quickly become familiar with them and able to make good use of all their features and capabilities
5	Modularity	Modularity is the degree to which a system's components may be separated and recombined easily.
6	Portability	It is the ability of a software application to run on numerous platforms such as data portability, hosting, viewing, etc.
7	Reliability	It relates to the software's ability to continue operating, under given environmental conditions, for a particular amount of time.
8	Resilience	Software resilience refers to the ability to absorb the impact of disruption in one or more parts of a system, while still continuing to provide an acceptable level of service.
9	Reusability	Reusability is the likelihood that a segment of source code can be used again to add new functionalities with slight or no modification.
10	Robustness	Robustness is the ability of the system to handle with abnormalities like errors during execution, erroneous input, sudden changes in hardware or software environment, etc.
11	Scalability	It is the ability of a system to handle the demand for stress caused by increased usage without decreasing performance.
12	Security	It is the ability of a system to resist or block malicious or unauthorized attempts that destroy the system and at the same time provide access to legitimate users.

Stress testing

It is also known as endurance testing. This type of testing is done to test if the system is going to continue working it encounters large volume of data than expected. It may contain internal tables, input transactions, disk space, communication channels, etc. It tests the system under extreme conditions.

Configuration testing

This type of testing is done to test compatibility issue. This testing helps to determine minimal and optimal hardware and software configuration. It shows the effects of adding or modifying resources such as disk space, memory, CPU, network card, etc.

Compatibility testing

This type of testing is done to test whether the system is capable of running on different operating system, hardware, network environments, applications, etc. It is similar to multi-platform testing. It is more useful in web-based applications where we can test if the application can be accessed from every browser.



CHECK YOUR PROGRESS

Q3: What are the types of test plans?

Q4: Write the steps involved in the approach for product analysis.

Q6: What is security testing?



12.8 LET US SUM UP

- System testing is a level of testing that helps to validate a complete and fully-integrated software product. Its purpose is to evaluate the end-to-end system specifications.
- It is independent of the development team and helps in testing the quality of the system impartial.
- System testing is performed after the integration testing step and before the acceptance testing step.

- System testing process has the following steps: create a test plan, test case creation, select test data, test case execution, bug reporting and fixation and repeat the test cycle (if required).
- Test data may be artificially taken or taken from live data.
- A test plan is a detailed document that has a catalog containing the test strategy, objectives, schedule, estimations, deadlines and resources that are required for completing the particular project.
- The components of a test plan are: scope, schedule, resource allocation, environment, tools, defect management, risk management and exit parameters.
- There are three types of test plan: master test plan, phase test plan and specific test plan.
- The process of creation of a test plan involves the following steps: product analysis, develop test strategy, define test objectives, define test criteria, resource planning, plan test environment, schedule and estimation and establish test deliverables.
- Activity network for system testing consist of the following activities: prepare test plan, specify conditions for user acceptance testing, prepare test data for program testing, prepare test data for transaction path testing, plan user training, compile and assemble programs, prepare job performance aids and prepare operational documents
- The different types of system testing are functionality testing, recoverability testing, performance testing, scalability testing, reliability testing, documentation testing, security testing, usability testing, requirements testing, load testing, stress testing, configuration testing and compatibility testing



12.9 FURTHER READINGS

1. Glenford J. Myers, Corey Sandler, Tom Badgett, The Art of Software Testing, 3rd Edition
2. Ron Patton, Software Testing, 2nd Edition, 2005

Fitness for use means that not only the essential requirements but the requirements of the customers are also taken into consideration. For example, customers may desire a certain product with some additional requirements, like a hatchback car with added space, comfort, performance, etc. As different customer may have different requirements, so there may be different models of the same products.

13.3.1 Functional and Non-functional requirements

For a product to be successful it should be able to meet all its goals and requirements. In case of software, there are two types of requirements which are described as below:

- **Functional requirements**: These are the fundamental criteria that the software should essentially satisfy and are usually stated by the users / customers. Some of the examples of functional requirements are that the system should be able to accept and process valid input, produce the anticipated output, should be free of errors and bugs, etc. Here the user is mostly able to visualize them in the software.
- **Non-functional requirements**: These are the requirements that are mostly specified by the software developer or project leader based on the requirements and restrictions of the system as a whole. Although it consists of many attributes like Complexity, Efficiency, Interoperability, Learnability, Modularity, Portability, etc. (as shown in table 13.1), but not all of these attributes are obligatory to be fulfilled.

13.2 INTRODUCTION

In the previous unit, we learned about the significance of System testing, its different types, cases and the process of performing testing. In this unit, we will understand the concept of software quality, the factor and level of quality assurance, and finally how quality is associated with software requirement specification and design specification. In the next unit, we will discuss about the implementation of a system.

13.3 SOFTWARE QUALITY

When we say quality of a product, different people may visualize it differently. It depends on the individual's need, interest, opinion and point of view. There can even be multiple levels in an individual's interpretation of quality. For example, a person with low income may regard a simpleMaruti-Suzuki car as having great quality, whereas a person with very high income may find the lowest model of Mercedes is of low quality. Another example is that in some manufactured products like say box-cover, if there are some products that miss the approved standards by a very slight margin, still they may be considered good because there is a high probability that customers might never notice it or even ignore such small defects. But same cannot be said in case of other products like eye glasses, medicines, etc. where even a small defect may not be acceptable. In some cases, people associate expensive and high price with good quality, and lesser price with lower quality. For example, an expensive item like bag or watch of international brands may be considered very good quality and the same item of lesser price may be considered having lesser quality.

To quantify quality in industry, two definitions of quality exist: "conformation to requirement" and "fitness for use". Conformation to requirement means that the requirements and standards for a product are clearly predefined and the product has to satisfy all these set requirements in order to be considered of good quality. For example, a hatchback car has to have certain requirements like a rear flip-up hatch door, front-wheel drive-train, foldable rear seats, etc. A car has to satisfy all the essential necessities of hatchback car to be considered a quality hatchback car.

12.10 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q No 1: System testing follows white box testing – False.

Ans. to Q No 2: System testing is performed before integration testing and after acceptance testing.

Ans. to Q No 3: Test data may be artificially taken or taken from live data. It means test data can be created solely for test purpose or may be taken from user's actual files. Artificial data should be properly created. It should provide all combination of values or formats. It should be made possible to test all logic and transaction path subroutines. Live data are biased towards typical values. Artificial data provides extreme testing conditions. It provides extreme values for testing the boundaries of the candidate system.

Ans. to Q No 4: There are three types of test plan—

- Master test plan
- Phase test plan
- Specific test plan

Ans. to Q No 5: The steps involved in the approach for product analysis: Perform product walkthrough, interview client, designer and developer and review product and project documentation in loops

Ans. to Q No 6: Security testing is a type of testing is done to ensure that the programs can be accessed by authorized personnel and that the authorized personnel can access the functions available to their security levels. The system should not allow unauthorized access to data and resources. The goal of security testing is to test how well a system protects against unauthorized internal or external access or damage. Security testing can be done in areas like authentication, authorization, data validation, transport security, data protection, session management, etc.



12.11 MODEL QUESTIONS

Q1. What is system testing?

Q2. What is the need of system testing?

Q3. Explain the components of a test plan.

Q4. How a test strategy is developed in the creation of a test plan?

Q5. What are test criteria? Explain its types.

Q6. Explain the test deliverables required before, during and after testing.

Q7. Write notes on activity network for system testing.

Q8. What is the difference between configuration testing and compatibility testing?

Q9. What is test plan?

Q10. What are the approaches for a test scheme?

*** ***** ***

UNIT 13: QUALITY ASSURANCE

UNIT STRUCTURE

13.1 Learning Objectives
13.2 Introduction
13.3 Software Quality
13.3.1 Functional and Non-functional Requirements
13.4 Levels of Quality Assurance
13.5 Quality Factor Specification
13.5.1 McCall's Factor Model
13.6 Software Requirement Specification
13.6.1 Structure of Software Requirement Specification
13.6.2 Characteristics of a Good Software Requirement Specification
13.7 Software Design Specification
13.7.1 Characteristics of a Good Software Design Specification
13.7.2 Software Design Principles
13.8 Let Us Sum Up
13.9 Further Readings
13.10 Answers To Check Your Progress
13.11 Model Questions

13.1 LEARNING OBJECTIVES

After going through this unit, you will be able to –

- understand the essence of quality and the concept of software quality as a whole
- learn about the issues and factors that affect the quality of software
- learn about the significance of Software Requirement Specification and how to formulate it
- understand principles of Software Design Specification