6.3 Validation Testing

Validation testing is the final series of software tests and succeeds when software functions is executed in a manner that can be reasonably expected by the customer. When customer software is built for one customer, a series of acceptance tests are conducted to enable the customer to validate all requirements. If software is developed as a product to be used many customers. It is impractical to perform acceptance tests with each one. Most software product builder's use a process called alpha beta testing to uncover errors that only the end user seems able to find.

The ***Alpha*** test is conducted at the developer's site by a customer. The software is used in a natural setting with the developer of the user and recording errors and usage problems. Alpha tests are conducted in a controlled environment.

The ***beta*** test is conducted at one or more customer sites by the end user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a live application of the software in an environment that cannot be controlled by the developer. The costumer records all problems (real and imagined) that are encountered during beta testing and reports these to the developer at a regular interval.

6.4 System Testing

Software is incorporated with other system elements (hardware, people, and information) and a series of system integration and validation tests are conducted. In the following are the types of systems tests.

Recovery Testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed.

Security Testing attempts to verify that protection mechanisms built into a system will protect it from improper penetration.

Stress Testing executes a system in a manner that demands resources in abnormal quality, frequency or volume.

Performance Testing is designed to test the run-time performance of software within that the context of an integrated system.

7. Software Quality

Quality is a characteristic or attribute of something. As an attribute of an item, quality refers to measurable characteristics— things we are able to compare to known standards such as length, color, electrical properties. However, software, largely an intellectual entity, is more challenging to characterize than physical objects.

7.1 Quality Factors

There are useful categorizations of factors that affect software quality. These software quality factors, shown in Figure 7.1, focus on three important aspects of a software product: its operational characteristics, its ability to undergo change, and its adaptability to new environments. Figure 7.1 provides the following descriptions:

Correctness: The extent to which a program satisfies its specification and fulfills the customer's mission objectives.

Reliability: The extent to which a program can be expected to perform its intended function with required precision.

Efficiency: The amount of computing resources and code required by a program to perform its function.

Integrity: Extent to which access to software or data by unauthorized persons can be controlled.

Usability: Effort required to learn, operate, prepare input, and interpret output of a program.

Maintainability: Effort required to locate and fix an error in a program.

Flexibility: Effort required to modify an operational program.

Testability: Effort required to test a program to ensure that it performs its intended function.

Portability: Effort required to transfer the program from one hardware and/or software system environment to another.

Reusability: Extent to which a program [or parts of a program) can be reused in other Applications related to the packaging and scope of the functions that the program performs.

Interoperability: Effort required to couple one system to another.

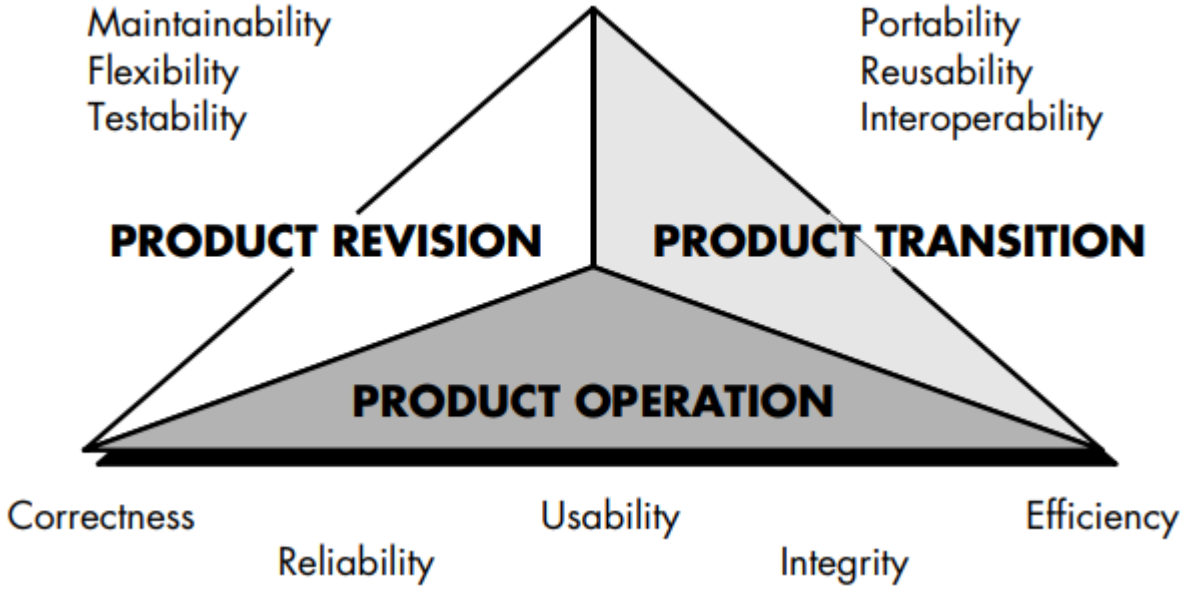

Figure 7.1 Software quality factors