system is to be tested should be statistically significant.

11.3 SOFTWARE QUALITY

Traditionally, the quality of a product is defined in terms of its fitness of purpose. That is, a good quality product does exactly what the users want it to do, since for almost every product, fitness of purpose is interpreted in terms of satisfaction of the requirements laid down in the SRS document. Although fitness of purpose is a satisfactory definition of quality for many products such as a car, a table fan, a grinding machine, etc.—fitness of purpose is not a wholly satisfactory definition of quality for software products. To give an example of why this is so, consider a software product that is functionally correct. That is, it correctly performs all the functions that have been specified in its SRS document. Even though it may be functionally correct, we cannot consider it to be a quality product, if it has an almost unusable user interface. Another example is that of a product which does everything that the users wanted, but has an almost incomprehensible and unmaintainable code. Therefore, the traditional concept of quality as fitness of purpose for software products is not wholly satisfactory.

The modern view of a quality associates with a software product several quality factors such as the following:

1. Portability: A software product is said to be portable, if it can be easily made to work in different hardware and operating system environments, and easily interface with external hardware devices and software products.

2. Usability: A software product has good usability, if different categories of users (i.e. both expert and novice users) can easily invoke the functions of the product.

3. Reusability: A software product has good reusability, if different modules of the product can easily be reused to develop new products.

4. Correctness: A software product is *correct*, if different requirements as specified in the SRS document have been correctly implemented.

5. Maintainability: A software product is *maintainable*, if errors can be easily corrected as and when they show up, new functions can be easily added to the product, and the functionalities of the product can be easily modified, etc.

B

11.4 SOFTWARE QUALITY MANAGEMENT SYSTEM

A quality management system (often referred to as quality system) is the principal methodology used by organizations to ensure that the products they develop have the desired quality. In the following, we briefly discuss some of the important issues associated with a quality system.

- A quality system is the responsibility of the organization as a whole. However, every organization has a separate quality department to perform several quality system activities. The quality system of an organization should have the full support of the top management. Without support for the quality system at a high level in a company, few members of staff will take the quality system seriously.
- The quality system activities encompass the following:
  - Auditing of projects
  - Review of the quality system
  - Development of standards, procedures, and guidelines, etc.
  - Production of reports for the top management summarizing the effectiveness of the quality system in the organization.

A good quality system must be well documented. Without a properly documented quality system, the application of quality controls and procedures become ad hoc, resulting in large variations in the quality of the products delivered. Also, an undocumented quality system sends clear messages to the staff about the attitude of the organization towards quality assurance. International standards such as ISO 9000 provide guidance on how to organize a quality system.