CHAPTER 2
CLASSIFICATION OF SOFTWARE QUALITY ATTRIBUTES - A REVIEW

2.1 INTRODUCTION

As per ISO 8402 standard, quality of software is the totality of features and characteristics of a product/ service and its ability to satisfy stated/ implied needs. Therefore, quality is defined as a complex and multifaceted concept [GAR84]. In context of software engineering, quality of the software is the most important factor which is directly associated with wellness and conformance of the software. Quality attributes acquire the characteristics associated with the software and act as the measures to determine quality of the software. Also, these attributes guide the software engineering process for conformance of all the aspects of software development. Further, it has been noticed that software quality attributes contribute equally for successful software development, even though all sub-characteristics do not equally affect software product and process [KG05]. It is mandatory for any software development organization to produce quality software; particularly in this competitive era as quality attributes play a major role in determining the quality of software. Each quality attribute pertains to specific feature or property of software and has unique effect on that software. There exist many classifications of software quality attributes in literature. However, the scope of a desirable generalized classification remains especially in view of software specification requirements, business requirements and development environment in an integrated manner. And thus, provide guidance and ease to conform quality of software product, process and service [PI09a].

In addition to the substantial benefits of quality software provided to the end users, improvements are required to confirm wellness of software use to resolve difficulties in understanding instruction, manipulating parts or interpreting feedback [NIL92, PRE04]. Hence, high usability is desirable for all types of software. As discussed in previous Chapter, usability is broadly characterized as the degree of ease to use the system i.e. software should
be understandable, learnable and operable [NIL93]. Earlier, usability has been considered as a quality factor only. It has been observed that usability is often associated with the functionalities of the product in addition to being solely a characteristic of the user interface [NIL92]. In view of the fact that software quality assurance is a continuous process, usability has also to be assured from conception till deployment of the software simultaneously.

In this Chapter, we also propose generalized classification of software quality attributes along with classification of usability attributes. The generalized classification will serve the need of an integrated framework for quality inclusion in software development. In addition, the usability attributes’ classification is used in order to support the development of usable software. In the organization of this chapter, the review of relevant literature is presented in Section 2.2. Section 2.3 deals with proposed generalized classification of software quality attributes in detail. In Section 2.4, we cover the classification of quality attributes in view of usability of software. It is a noted fact that the software quality attributes are immensely important and playing roles with predefined behavior during software development [BBL76]. The change in view of quality endorsement modifies quality attributes’ hierarchical organization. In Section 2.5, we present comparison of generalized classification of software quality attributes and classification of usability attributes highlighting major usability attributes with intent of revealing the said hierarchical variation. Usefulness of these classifications and the dynamic role of the attributes are discussed in Section 2.6. Finally, we conclude with summary in Section 2.7.

2.2 BACKGROUND

Many researchers have presented certain issues related to the quality attributes and the respective classification. As per International Standards Organization (ISO), quality standards cover all the aspects of software quality. ISO 9126 is an international standard for the evaluation of software quality and has been covered with four major aspects namely; quality model, external metrics, internal metrics and quality in use metrics. Quality model has been described in: ISO/IEC 9126-1:2001 [ISO01]. ISO/IEC TR 9126-2:2003 has covered the external quality metrics. Internal metrics has been defined in ISO/IEC TR 9126-3:2003.
ISO/IEC TR 9126-4:2004 and used to define quality in use metrics. In addition, ISO/IEC 25000:2005 has also been used as guide for Software product Quality Requirements and Evaluation (SQuaRE) [ISO03a, ISO03b, ISO04, ISO05]. As per ISO 9126-1, the classification explores software quality attributes in terms of a structured set of characteristics as functionality, reliability, usability, efficiency, maintainability and portability, which have further been divided into quality sub-attributes.

There exists a variety of quality classification models such as McCall’s model, Boehm’s model, model of Software Assurance Technology Center (SATC), Software Quality Institute model, common subsets model etc. It is observed that there exist three important aspects of software product such as operational characteristics (also known as product operations), ability to undergo changes (termed as product revision) and adaptability to new environment (referred as product transition) [PRE04]. Operations aspect of the system covers correctness, reliability, usability, integrity and efficiency whereas maintainability, flexibility and testability have been considered as revision aspect elements of a system. Portability, reusability and interoperability attributes deal with adaptability of system. Another model represents a multilevel classification of quality attributes [FAI85, IEEE83]. General utility is at the root level in the representation which has further been expanded through quality factors maintainability, as-is utility and portability. Maintainability quality factor covers sub-factors modifiability, understandability and testability. Human engineering, efficiency and reliability have been encapsulated within as-is-utility whereas device independence and self-containedness have considered being the sub-attributes of portability. Further subdivision of these quality sub-attributes includes many crosscutting attributes. The model views the decomposition of quality attributes into source code characteristics.

The quality can be viewed in terms of goals such as requirement quality, product quality, testing effectivity and implementation effectivity [HR96]. Ambiguity, completeness, volatility, understandability, traceability, structure, maintainability, reusability, internal/external documentation, resource usage, completion rates and correctness are the attributes identified to achieve the stated goals. One improvement in ISO-9126 classification includes
reusability as the top level attribute with additional subordinate properties. Functionality, reliability, efficiency, usability, maintainability, portability and reusability are defined as the primary classes of attributes. Further subdivision of these classes includes the attributes to be used in other environments [DRO98]. The quality attributes in the form of subsets may be applicable for a specific phase of software development. A group of certain attributes such as Reliability, Availability, Serviceability, Usability, and Installability (RASUI) has been referred for effectiveness of the system, whereas a collection of attributes such as Functionality, Usability, Reliability, Performance, and Supportability (FURPS) has been proved important for software requirements conformance. Reliability, Availability, Scalability, and Recoverability (RASR) collectively represent another set of quality attributes recommended for assessing quality of the database. Some quality attributes such as Reliability, Availability, Maintainability and Safety (RAMS) are clustered together to deal with safety-critical systems [PRE04]. Above all, quality is conferred as a continuing commitment to produce quality software and provide a quality service needed at all levels and in all parts of the IT organization [LOU93]. Another detailed coverage has been presented that includes quality criteria, quality factors, document quality in view of users' need for operations and maintenance of system [LINK04]. Many more quality attributes are identified and most of them are based on ISO standards of quality attributes' classification [LINK05, LINK06, LINK07, LINK08, LINK09, JIM10, LINK10].

With the technological revolution, computers are made available to broader group of people for variety of innovative tasks with increase in number of users and applications. Users are becoming less willing to use difficult and uncomfortable software and interfaces. Software usability is thus desirable to ensure ease of use [NIL92]. Therefore, the need has emerged to perceive the system/ software including usability concerns in the software development process and to measure the quality from the users' point of view i.e. comfort and convenience to use. As per the source of usability evolution, usability consists of attributes; learnability, efficiency, memorability, satisfaction and errors [NIL92]. Another model introduced flexibility as an attribute of usability along with effectiveness, learnability and attitude attributes [SHA90]. As per ISO/ IEC, usability has been considered as the group
of quality attributes namely; understandability, learnability and operability whereas usability is considered as a combination of operability, training and communicativeness in McCall’s model [PRE04, IEEE83, FAI85].

A comprehensive manifestation of quality is presented with intent of systematic characterization of usability in terms of behaviors. This model entails usability definition in terms of a set of characteristics namely; learnability, transparency, operability, responsiveness, customizability, foreign-language provision, command-context sensitivity, operational directness, hot-key and consistency. There exists subordinated behavior of some attributes amongst all aforesaid quality attributes. Usability has been given the importance at first level in hierarchy of some effective quality attributes [GEO98]. On the other hand, all other quality representations consider usability as one of the quality characteristics [FAI85, HR96, CON01, ONL02, FIR03, KG04]. Being usability as functional system of software, it has to be viewed as a whole in terms of all other quality characteristics and sub-characteristics for product and process improvisation. It will lead to understand usability issues and to improve usability of software thereby improving the whole software development process.

Software quality attributes are the multi-dimensional characteristics associated with multifaceted behavior of software. It has been clearly mentioned in literature that each quality attribute pertains to specific feature or property of software such as; control, performance, error tolerance, verifiability, accessibility etc. and has unique effect on software [IEEE99, BBL76, B+78. FSO04, ISO01]. In addition, classification of quality attributes with different perspectives is required for the successful software development, thereby providing guidance and ease to conform quality of product, process and service [BD02, G+07]. It has been noticed that software quality attributes contribute equally for successful software development even though all sub characteristics do not equally affect software product and process [BBL76, WBBF04, YIJ05]. Many classifications of quality attributes have been cited focusing on specific view. As per the view, the organization of quality attributes has modified. Nevertheless, it has raised the point whether different perspectives of software
quality attributes mould this behavior. And, it has been noted that with amendment in the quality organization, the role and behavior of the attribute also changes. In other words, modification in the organization of software quality in terms of quality attributes may transform the role and behavior of the attributes. Such dynamism may affect the quality of the software and thus its development. Hence, there exists a strong need to study the role, behavior and impact of software quality attributes in different perspectives.

2.3 GENERALIZED CLASSIFICATION OF SOFTWARE QUALITY ATTRIBUTES

We present a generalized classification of software quality attributes for systematic clustering in this Section. It is required to facilitate ease of quality inclusion in the software throughout the development process, and at the end to produce quality software as an outcome. At the same time, objective of quality service to the customers must also be achieved.

The literature survey reveals that each classification has cited with a specific view. Coverage of limited attributes and their usage have been the primarily observed limitations. Further, all sub characteristics do not equally affect software product and process [KG05]. Also, certain quality attributes such as functionality, reliability, availability and usability have been common in most of the classifications, but have been composed in different manner. In addition, many more quality attributes are identified which have not yet been classified relevantly [SQJ05, SWLM97]. However, there exists a lot of scope for generalized classification of quality attributes with systematic categorization of attributes in an integrated manner.

We have proposed generalized classification of software quality attributes with six major classes namely; Runtime attributes, Non-runtime attributes, Business Oriented attributes, Architecture Oriented attributes, Domain Specific attributes and Impact Oriented attributes in view of product, process and service as shown in Fig.-2.1. We describe each of these classes and further classification as follows:
Software Quality

Runtime Attributes

Non-runtime Attributes

Business Oriented Attributes

Architecture Oriented Attributes

Domain Specific Attributes

Impact Oriented Attributes

Fig.-2.1: Major Software Quality Attribute Classes
2.3.1 Runtime Attributes

This class of quality attributes is strictly concerned with the execution time. These attributes are required to be measured at the time of system execution. At run time, it is important to know about the working of the software and its behavior. It highlights on the fundamental process of transformation that software and hardware components of the system perform on inputs to produce outputs. For example, response of the system is considered to be of great importance and hence at the time of user-system interaction, the time and functional aspects of the software have to be thought about. Therefore, the quality attributes concerned with the software in this manner has taken up in runtime class. As shown in Fig.-2.2, we have identified sub-attributes of runtime class such as security, functionality, interoperability, performance, accessibility, usability, availability, and traceability that are used to assess the run time behavior of the system.

Security

Security is the first sub-class that deals with the ability of the system to resist unauthorized attempts of usage/behavior modification while providing service to users. It protects information from theft or corruption, and the preservation of availability.

Interoperability

It is the ability of system to cooperate with other systems while in execution and hence concerned with run time behavior of the system. The interaction with other systems includes interfacing with hardware (devices desired for development and operations of software), other software and related tools at various stages of the software development life cycle.

Accessibility

It is defined as the ability to access the functionality of the system, and to get possible benefit of the system, when the system is in operation. Traceability is associated with accessibility being ability to verify the history, location, or application of an item by means of documents at the time of software execution.
Fig.-2.2: Runtime Class of Software Quality Attributes
**Functionality**

Functionality refers to the ability of the system to accomplish intended work. It has been further classified into understandability, conciseness, consistency and clarity. Understandability refers to the capability of system being understood and accepted under the circumstances to accomplish any task. Conciseness is concerned with expressing more in few words. Consistency is the property of uniformity of successive results or events and is considered as part of functionality. Clarity refers to the ability of the system that clearly visualizes the concepts, as in thought, understanding and mind.

**Performance**

Performance is related with the ability of the system to be timely used as desired. It is the measure of the desired outcome and applicability of the software fulfilling the said objectives. It has been further classified into attributes such as utility, response time, throughput behavior of the system, timeliness and structure. Utility is the capacity of the system to work as needed. Response time is the time elapsed from submitting an instruction till the first response of the system. Throughput behavior of the system is defined as the amount of work done in a given time. To improve performance of the system, maximum throughput is expected. Timeliness being the property of time plays a major role for better system performance. To get the desired performance of the system, sufficient storage of data and results has necessarily to be provided.

**Usability**

Usability is known as a measure of convenience and practicability of the product for intended users. Ease of use and ease of training to the end users pertaining to the system proved to be of great importance. Thus, usability has to be conformed at run time. It has further been subdivided into set of attributes working to have usable software namely; learnability, efficiency, access control, un-ambiguity, validity, resilience, customizability, practicability and operability. Learnability is the ability to know the details of the system without any external support. Efficiency being the property of a system to fulfill the requirement perfectly in short runs of time desired to be attribute of usability. Access control
deals with the protection and security of the system when in use. Repeatability is defined as the variability of the measurements obtained by one person while measuring the same item repeatedly. Another attribute in usability subset is un-ambiguity that is the ability of the system being interpreted in the same manner by all the users. It makes the users understand the intended purpose of the system. Validity refers to the logical, analytical or necessary trueness of the system at the time of use. Resilience is considered to be usability attribute being the property of system to energize itself when deformed elastically and then, recover. The software is used by many users for different purposes and satisfaction of the users is important. Customizability is the ability of the software to be changed by the user or programmer as per the need and to provide user satisfaction. Practicability is the ability that makes the software usable for a specified purpose. Operability is the major aspect of usability and defined as ability to keep a system in a functioning and operating condition.

Availability

Availability is described as the measure of time when the system is up and running correctly. It is the elapsed time between failures and the time needed to resume operation after a failure. Availability branches into set of attributes such as reliability, sustainability and anomaly management. Reliability is essential to confirm availability since it is the capability of software to maintain its level of performance under stated conditions for a stated period of time. Sustainability also plays an important role to measure availability as it is the property to uphold the system for the required function. Anomaly management deals with anomalies that may exist while system is running.

2.3.2 Non-runtime Attributes

Attributes of this class do not concerned with run time behavior of the system but play major role in determining quality in offline manner. Also, it has been noticed that the conformance of runtime quality depends on the conformance of non-runtime quality. It specifies criteria that can be used to judge the operation of a system and the way the system will do it. For example, software performance requirements, software external interface requirements, software design constraints are considered to be non-functional requirements
of the system and are embodied in the static structure of the software system. Non-functional requirements are difficult to test; therefore, they are usually evaluated subjectively with overall characteristics. Fig.-2.3 depicts such attributes that can evaluate non-runtime behavior of the system namely; documentation, manageability, completeness, portability, accuracy, integrity, reusability and testability.

Documentation

The documentation sub-class of non-runtime attributes refers to the process of preparing and providing evidences in the form of communicable material for system investigation. Successful system development has the essential requirement of well documentation. Completeness is concerned with implementing all the capability in terms of sufficient data items, functions, interface and code and hence it is considered as a class of non-run time attributes. The sub-class accuracy has been defined as the degree of conformity of a measured or calculated quantity to its actual value. Integrity is also a non-run time attributes and defined as the ability to separately develop components and make them work together correctly. Components are functioning in specific manner to achieve objectives, at the same time collectively working to achieve common goal of the system.

Testability

Testability attribute has given a great importance in software development as it provides error handling capability to the system. Testing of the system has to be accomplished prior to the execution of the system, thus testability plays a major role in survival of the software and considered to be a main non-runtime attribute.

Manageability

Manageability is the ability of the system to plan, organize resources, direct, administrate and control the overall process of software development. Manageability has further extended in a group of attributes, modifiability, maintainability and flexibility with equal share of each attribute for the intended purpose. Modifiability is the ease with which a software system can accommodate required changes. Maintainability facilitates updates to
Fig.-2.3: Non-runtime Class of Software Quality Attributes
satisfy new requirements. The software product that is maintainable is simple, well-documented, and should have spare capacity for processor and memory usage. Flexibility is defined as the ability to adapt to different circumstances.

**Portability**

Portability is the ability of a system to run under different computing environments. The software environment consists of a combination of hardware and software both. A set of attributes such as; machine independence, system independence, replaceability, installability, adaptability and data commonality has been covered under portability. Machine independence refers to the ability of the software to be used on any machine (i.e. hardware). System independence is the ability of the software to be installed, operated and modified on any system (i.e. hardware and operating system). Presence of these two attributes generates possibility of installability which is the capability of the software product to be installed in a specified environment. Installability is further extended to distributability that is the ability to share the common resources for applications and users. Adaptability leads to improve portability as it is the ability of the system to be modified by circumstances. Data commonality refers to the availability of common data for all the users and applications. Replaceability is the capability of the system to retain itself after replacement of data structure, function, module or program.

**Reusability**

Reusability is referred as the degree to which existing applications can be reused in new applications and hence identified as a sub-class of non-run time attributes. It has further been classified into a cluster of attributes recommended to measure reusability such as; representation independence, application independence, data encapsulation, function encapsulation and interfaceability. For reuse, system has to be independent of representation and application used. Data and function encapsulation are necessary for reusability as it refers to hiding the details about data and function. A major characteristic of reusability is interfaceability, which deals with exchanging information across the common boundary shared by the components including users.
2.3.3 Business Oriented Attributes

There exist many non-software quality attributes that influence other software or non-software quality attributes. These attributes attempt to confirm quality in view of business policies. Business objectives are specific statements that give projections about growth or development of companies. These are the stated, measurable targets to achieve business aims. The effective business objectives have to be specific, measurable, agreed, realistic and time specific. For example, a business objective could be to increase sales of the product by next year. We have presented this class of quality attributes as a combination of attributes such as cost and schedule, marketability, appropriateness for organization, and localization are observed to be quality attributes that affect business system as highlighted in Fig.-2.4.

Cost and Schedule

Cost and schedule is the ability to account cost of the system with respect to time and market, expected project lifetime, and utilization of systems. It plays significant decision making role for planning and executing business policies. Also, has a strong relationship with organizational profits and thus existence of the organization in today’s competitive era.

Marketability

Another influential class of business oriented attributes has termed as marketability. It is the capability to use the system with respect to market competition and to use feedback to increase profitability of the business. One must understand the market needs and have to take appropriate actions accordingly.

Appropriateness of Organization

This sub-class of business oriented attributes has been explored for availability of the human input, allocation of expertise, and alignment of team and software structure. Also, it facilitates business process re-engineering to provide adaptable system. Generality and commonality are the factors that contribute to measure appropriateness of organization. Generality refers to the availability of the system for the majority of people while
Fig. 2.4: Business Oriented Class of Software Quality Attributes

Fig. 2.5: Architecture Oriented Class of Software Quality Attributes
commonality is the ability of the software to be used with common features for distinct users.

Localization

Localization is stated to be the major sub-class of business oriented attributes as any business system has to be localized to realize benefits. It helps to grasp the need of the local customers and to wrap/cover the closer territories of business.

2.3.4 Architecture Oriented Attributes

The quality attributes used to measure structural aspect of the system are known as architecture oriented attributes. There are many common ways of designing software modules and communications between these modules. For example, blackboard, client-server, database-centric architecture, distributed computing, event driven architecture, implicit invocation, monolithic application, peer-to-peer, pipes and filters, plug-in, representational state transfer, structured (module-based), service-oriented, search-oriented, space based, shared nothing, three-tier model are some of the available architectures that can be used while system development. Architecture oriented attributes are also non-software attributes. Fig.-2.5 illustrates attributes that are considered for assurance of the quality of structural design of the system namely; conceptual integrity, correctness, structuredness, scalability, extensibility, supportability and self-containedness.

Conceptual Integrity

Conceptual integrity has been defined as the integrity of the overall structure composed from variety of small architectural components. Therefore, it plays a vital role for building quality software. Accountability to satisfactorily fulfill all requirements of the system is expressed in order to achieve conceptual integrity.

Correctness

Realization of requirements has dependency on design; hence correctness has been covered under architecture oriented attributes. It has been identified as the characteristic of
various processes and work products of both; a development process and management process of the software under development.

**Structuredness**

This aspect of the system is the capacity to organize activities in well manner and is of great interest while analysis and design of the software.

**Scalability**

Scalability deals with handling growing amount of work in a graceful manner, or to be readily enlarged. Therefore, it perfectly occupies a place in architecture oriented class.

**Extensibility**

Extensibility is a system design phenomenon where the implementation takes future growth into consideration. It is a systemic measure of the ability to extend a system and the level of effort required to implement the extension.

**Supportability**

Supportability has been defined as affectivity of system structure to provide service to the intended users. It has been proved that the basis of a desired system is a good architecture and hence essential to provide support of use.

**Self-containedness**

Self-containedness is the ability of the system to contain its projected purpose in its definition and hence considered as architecture oriented attribute. In other words, it may be called as self descriptive and self sufficient in context of specificity of software.

**2.3.5 Domain Specific Attributes**

This class of quality attributes deals with the specific business and application domain. The objective of defining the domain is developing mass-customized products that reduce the costs, delays, and inflexibility characteristic of software and systems. Also, it
increases the ability to align business activities to the needs and produce solutions to cater the specific needs. It has further been categorized in three sub-classes that can contribute for the stated purpose such as sensitivity, calibrability and stability and represented in Fig.-2.6.

**Sensitivity**

Sensitivity is described as the degree to which a system component can pick up something being measured. It has dependency on the area of application and development.

**Calibrability**

Calibrability is defined as the ability of the system to recalibrate itself to some specific working range. It has been associated with unique standards and measurements against the standards for evaluating product and process.

**Stability**

Stability is the sub-class of domain specific attributes and is the degree to which software can be run over periods of time without crashing or otherwise malfunctioning. It has termed as a characteristic of application area and the platform used to build the system.

**2.3.6 Impact Oriented Attributes**

This class of quality attributes is mainly concerned with determining the overall effect of the system from users view point. User satisfaction is the primary goal of any software and has to be cultivated at conception stage of the system. Therefore, software has to be available in such a manner that influences the user without a need to redress. The software should have influence on the users in many ways. Hence, simplicity, communicativeness, maturity, self-descriptiveness, fault tolerance, affect and helpfulness are the major factors contributing to evaluate this aspect of the system as illustrated in Fig.-2.7.
Fig.-2.6: Domain Specific Class of Software Quality Attributes

- Stability
- Calibrability
- Sensitivity

Fig.-2.7: Impact Oriented Class of Software Quality Attributes

- Simplicity
- Communicativeness
- Maturity
- Helpfulness
  - Visibility
  - Survivability
- Fault tolerance
- Self-descriptiveness
- Affect
Simplicity

One class of impact oriented quality attributes is simplicity defined as the property of being simple or uncombined i.e. the availability of the system without complexity. Also, a simple system is easy to understand and use, and hence proved effective.

Maturity

Maturity attribute represents the state of the system being sensible and fully grown. Hence, maturity has considered being directly proportional to effectivity.

Fault Tolerance

It is the ability of a system to continue performing operations properly, even in case of failure of some of its components. Since it has been desired for proper functioning of the system, it is considered to be the attribute of impact oriented class.

Affect

It is also classified as an impact oriented attribute as it is the capability of the system of being loaded and used effectively. Affect is the measurement of influence of the system over organization wide operations. Any decision has effect over the organizational process. Therefore, affect has been considered to have significant share in view of totality of the system.

Communicativeness

Communication is another sub-class of impact oriented attributes as it is the ability to convey desired and expected information to the intended users, organizational process or within the system.

Self Descriptiveness

We have identified another attribute of impact oriented attributes as self descriptiveness. It is interpreted as the capability of the system to sufficiently describe itself and hence shows suitability with impact oriented class.
Helpfulness

Helpfulness is the degree with which system is ready to help users and hence justify itself to be the attribute of impact oriented class of attributes. Helpfulness is further classified into a pair of attributes such as; visibility and survivability. Visibility is the transparency of system in context of interface, data and communication. Survivability refers to the ability of the system to persist in the same mode till use.

The classes in generalized classification and attributes in each class are shown in Table-2.1.

2.4 CLASSIFICATION OF USABILITY ATTRIBUTES

Coming across the avant-garde trends and innovative concepts used in software development in which usability has been emerged as the most important concern for a quality product, it seems essential to reorganize quality attributes in terms of usability. In view of generalized classification and usability benefits, classification of quality attributes can be reviewed in usability perspective. Now days, user satisfaction, comfort and convenience are appearing important while using software with increase in number of computer users, applications, variety of tasks performed and the manner of use. Therefore, it has become necessary to confirm wellness of use of the software to resolve difficulties while understanding instruction, executing tasks, interpreting data/ feedback irrespective of the level of users. Since usability is associated with the user satisfaction directly, it has encouraged reviewing software quality attributes classification in usability perspective.

We have proposed another classification of usability attributes also in this Chapter. Instead of being solely a characteristic of the user interface, this classification has considered usability as a functional system of software where all other quality attributes participate in usability of the software [PI10]. Here, the proposed classification mainly follow the definition and objectives of usability as; the usability of product is the degree to which specific users can achieve specific goals within a particular environment, effectively, efficiently, comfortably, and in an acceptable manner [ISO98]. The attributes classified in this manner are termed as usability attributes. The proposed classification of usability
### Table-2.1: Generalized Classification of Software Quality Attributes

<table>
<thead>
<tr>
<th>1.0 Runtime</th>
<th>2.0 Non-runtime</th>
<th>3.0 Business Oriented</th>
<th>4.0 Architecture Oriented</th>
<th>5.0 Domain Specific</th>
<th>6.0 Impact Oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Functionality</td>
<td>2.1 Manageability</td>
<td>3.1 Cost &amp; Schedule</td>
<td>4.1 Conceptual integrity</td>
<td>5.1 Sensitivity</td>
<td>6.1 Simplicity</td>
</tr>
<tr>
<td>1.1.1 Understandability</td>
<td>2.1.1 Manageability</td>
<td>3.2 Economy</td>
<td>4.2 Correctness</td>
<td>5.2 Calibrability</td>
<td>6.2 Maturity</td>
</tr>
<tr>
<td>1.1.2 Concision</td>
<td>2.1.2 Manageability</td>
<td>3.3 Marketability</td>
<td>4.3 Structuredness</td>
<td>5.3 Stability</td>
<td>6.3 Installability</td>
</tr>
<tr>
<td>1.1.3 Consistency</td>
<td>2.2 Portability</td>
<td>3.4 Appropriateness of Organization</td>
<td>4.4 Scalability</td>
<td></td>
<td>6.4 Fault tolerance</td>
</tr>
<tr>
<td>1.1.4 Clarity</td>
<td>2.2.1 Machine</td>
<td>3.5 Localization</td>
<td>4.5 Extensibility</td>
<td></td>
<td>6.5 Helpfulness</td>
</tr>
<tr>
<td>1.2 Performance</td>
<td>2.2.2 Independence</td>
<td></td>
<td>4.6 Supportability</td>
<td></td>
<td>6.6 Affect</td>
</tr>
<tr>
<td>1.2.1 Utility</td>
<td>2.2.3 System Independence</td>
<td></td>
<td>4.7 Self-containment</td>
<td></td>
<td>6.7 Communicativeness</td>
</tr>
<tr>
<td>1.2.2 Response timing</td>
<td>2.2.4 Replaceability</td>
<td></td>
<td></td>
<td></td>
<td>6.8 Self-descriptiveness</td>
</tr>
<tr>
<td>1.2.3 Throughput behavior</td>
<td>2.2.5 Installability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2.4 Timeliness</td>
<td>2.2.6 Adaptability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.3 Security</td>
<td>2.2.7 Data commonality</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.4 Availability</td>
<td>2.3 Reusability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.4.1 Reliability</td>
<td>2.3.1 Representation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5 Usability</td>
<td>2.3.2 Independence</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.1 Learnability</td>
<td>2.3.3 Application</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.2 Efficiency</td>
<td>2.3.4 Data encapsulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.3 Access Control</td>
<td>2.3.5 Function encapsulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.3.1 Repeatability</td>
<td>2.3.6 Interfaceability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.4 Usability</td>
<td>2.4 Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.5 Validity</td>
<td>2.4.1 Manageability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.6 Resilience</td>
<td>2.4.2 Manageability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.7 Customizability</td>
<td>2.4.3 Manageability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.8 Practicality</td>
<td>2.4.4 Manageability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5.9 Operability</td>
<td>2.4.5 Manageability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.6 Interoperability</td>
<td>2.5 Testability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.7 Accessibility</td>
<td>2.5.1 Manageability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.8 Traceability</td>
<td>2.5.2 Manageability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 6.0 Impact Oriented
  - 6.1 Simplicity
  - 6.2 Maturity
  - 6.3 Installability
  - 6.4 Fault tolerance
  - 6.5 Helpfulness
  - 6.6 Affect
  - 6.7 Communicativeness
  - 6.8 Self-descriptiveness
attributes has five major classes namely; Conceptual, Design, Business, Conformance and Post implementation class of usability attributes in view of product, process and service as shown in Fig.-2.8. For coverage of quality attributes, this classification uses generalized classification of software quality attributes [PI09d]. We describe each of these classes and its further classification as follows:

2.4.1 Conceptual Attributes

These are the attributes that are related to the fundamental block building of the system. Usability is not the concern of consideration at implementation stage only. We can understand with an example of building a house where its usability has to be considered from the aspects of its area, its plan and facilities which accomplished through drawing maps with all the measurements assuming that it would be real. Similarly, it has to be adopted for the software construction. Hence, at conceptual level of software development we have to provide those measures with which usability can be guaranteed. This class of attributes contains two primary attributes namely; practicability and learnability as shown in Fig.-2.9. We classify conceptual attributes of usability as follows:

Practicability

It is defined as the measure of usability of the software for a specified purpose. It aims at providing easy to use system. Usability objectives must be defined at the basis level of development prior to start the development process and hence practicability of the system is mandatory. Accessibility and calibrabilty are sub-attributes contributing in assessing practicability of the software.

Learnability

It is the measure of comfort for users to accomplish basic tasks while first time they encounter the system. Ease of use varies from system to system. Hence, this issue has to be carefully incorporated prior to design of the system. Usability issues must be learnt to implement usability. An application may have distinct users. It is desirable to know user category and ease of use from respective view point. Therefore, learnability has been sub-
Fig.-2.8: Software Usability Attributes Classes
Fig.-2.9: Conceptual Attributes of Usability

Fig.-2.10: Design Attributes of Usability
divided into attributes such as understandability, completeness and scalability. At the stage of conception understanding with the system has necessarily to be developed in view of usability. All the usability references must be complete and there should not be any size constraints on the usability of the system. Usability has to be treated as a generalized feature.

2.4.2 Design Attributes

Usability has also referred to apply methods/techniques for improving ease-of-use during the software design process. It has been noted that failure or success of the system/software mostly depends upon design of the system. Since erroneous design reduces the possibility of successful software development and implementation, it must be carefully handled while usable software development also. Process of usability consideration has to be initiated before design, to be continued during the design and has to be carried after the design. Hence, quality design is essential to acquire usable software. User centered design is recommended for usable software development [CL99]. As shown in Fig.-2.10, we discuss the attributes contributing in this class as follows:

Operability

It has defined as the ability to keep a system in a functioning and operating condition. For attaining usability, user-centered design is the core considering assurance of systems' operations in a desired manner. Therefore, operability has been included unavoidably a design concern. Focusing on the importance of design for usability implementation, there exist some attributes that seem to appear for operability of the software such as; functionality, performance, interoperability, fault tolerance and simplicity. System's performance (desirable), its interaction with the other systems (internal or external), continuing operations properly in the event of the failure are identified as the operational characteristics from the perspective of usability. A simple system is always easy to understand and use. Therefore, simplicity has to be the design characteristics related to operability.
Efficiency

It is termed as the relationship between the level of performance of the software and the amount of resources used, under stated conditions. Usable software has to be efficient enough to provide desired level of satisfaction. Once again, this issue has to be addressed during the design of the system. Therefore, efficiency has occupied a place in design class of usability attributes. The attributes identified being associated with efficiency are namely; utility, response time, throughput behavior, sensibility and stability. System should be utilized fully and it must be stable while operations are being performed. It must handle the operations carefully so as to satisfy variety of users using the system. The system response has to be fast so that throughput can be increased. All of these concerns must be put on stress while designing the system.

Access Control

Without any disagreement, it is accepted that permissions to the system’s access are to be defined at the design stage. It helps user to understand the type and the manner of access allowed to them. Thus, access control is posed in the design class of usability attributes. There exist sub-attributes such as security, integrity and repeatability contributing in assessment of access control. Properly defined access controls improve security and integrity of the system. Also, it helps to maintain integrity of the system with repeatable use.

Un-ambiguity

It is the ability to understand the system by all its users in the stated manner. Hence, un-ambiguity must be the main issue in usability design. Un-ambiguous design increases the probability of successful software building. The attributes namely; consistency, clarity and interpretability are identified as the sub-attributes of un-ambiguity due to inclined behavior of these attributes towards invariable understanding of users. The interactions with the system have to be clear and consistent so as to make the use of system easy for variety of users.
2.4.3 Business Attributes

Similar to meta-data, there exist meta-attributes of usability. These are the attributes bearing dependency of other attributes and even influence the existence of the usable system/software itself. It is the fact that the ultimate goal of any kind of development is to gain profit at organizational level and in business as well. And, it is true for usable software development also. In this case, an important aspect of user satisfaction has been included within the stated goal of a business organization. Affect and customizability are identified as business class usability attributes and have been sub-divided further as shown in Fig.-2.11.

Affect

It is the measure of organization wide operations. In view of the software usability, influence of the software towards the business is the major concern. It has classified in sub-attributes namely; conciseness, cost and schedule, economy, marketability and appropriateness of organization. Though business needs every aspect to be covered, unnecessary tasks are to be avoided so as to make system concise. Reachable schedule, cost and economy are the essential factors of business success. Marketability is the important issue for any business to gain profit. Also, the appropriateness of the organization to handle the system and thus the business affects business.

Customizability

The ability for software to be changed by the user or programmer is called customizability. Main objective of the software always rests at fulfilling users’ requirements, and hence the software is to be custom built for usability implementation. To achieve this objective, customizability is sub-divided into sub-attributes such as reusability, documentation and extensibility leading to perform as per the expectations of the user with ease and comfort.

2.4.4 Conformance Attributes

Conformance is the measure of wellness of the software to meet the defined specifications. It is the verification of criteria and definitions for the desired outcome. The
Fig.-2.11: Business Attributes of Usability

Fig.-2.12: Conformance Attributes of Usability
major concern of this class of usability attributes is to attain the usability with conformance of concept and design to achieve business objectives and goal. It has been further sub-divided into sub-attributes namely; helpfulness and validity as shown in Fig.-2.12.

**Helpfulness**

It is a measure of ability to provide help through all stages while system is in use. It supports for conformance of usability. It has been implemented with sub-attributes such that portability, localization and supportability.

**Validity**

It refers to the measures of logical, analytical or necessary trueness of the system. For conformance of analysis and design of the system, valid system is to be provided. Its sub-attributes are namely; availability (includes reliability), accuracy and correctness. Software tends to confirm design in terms of reliability, accuracy and correctness.

**2.4.5 Post Implementation Attributes**

It has been discussed earlier that usability must be implemented as functional feature of the system. Though usability consideration is thought about from conception, full attention must be given to verify the operational system in context of user satisfaction in all the aspects of its working. It has to be confirmed after the field installation of the system at the same time. It is therefore essential to collect the feedback information which in turn will be used for desired improvements. The usability attributes possessing the properties associated with post development activities of the software are termed as post development attributes of usability. Usability attributes of this class along with sub-attributes as presented by Fig.-2.13 are explained as follows:

**Resilience**

It is defined as ability to maintain the system for any unfortunate happening. For any usable system this attribute is central to the objective. This has to be performed while the system is in use and hence considered to be post implementation usability attribute.
Fig.-2.13: Post Implementation Attributes of Usability
Manageability is its major sub-attribute which has further been sub-divided into maintainability, flexibility and testability sub-attributes. To possess resilience, software must be manageable i.e. it should be flexible, testable and hence maintainable.

**Adaptability**

It is the ability of the software to adapt external and environmental changes. A usable system has to be adaptive. It is apparently be the post implementation class usability attribute as it is concerned with the adaptive maintenance of the software.

In this classification, it should be noted that some usability attributes occur in all the classes such as; structuredness, timeliness and documentation. These attributes are required at each stage of software development to produce usable software. Table-2.2 summarizes the classification of software quality attributes in usability perspective (i.e. classification of usability attributes).

**2.5 COMPARISON**

Other than ISO standards of classification of software quality attributes, we have proposed classifications of software quality attributes namely; generalized classification and classification of usability attributes and are illustrated in Fig.-2.14. Quality being essential feature of the software, it is really important to observe the participation of quality attributes with associated impact. Also, changes may be attempted for improvements to provide most appreciable and usable software. With this intent, we illustrate the vigorous role of quality attributes in above mentioned classifications [P109e, P109h]. This Section deals with comparison of generalized classification and classification of usability attributes by referring the primary level usability attributes as underlined and highlighted in italic in Table-2.2 and Table-2.1 to exhibit the varying behavior. These attributes are namely; access control, adaptability, affect, customizability, efficiency, helpfulness, learnability, operability, practicability, resilience, un-ambiguity and validity.
Table-2.2: Classification of Usability Attributes

<table>
<thead>
<tr>
<th>1.0 Conceptual</th>
<th>2.0 Design</th>
<th>3.0 Business</th>
<th>4.0 Conformance</th>
<th>5.0 Post Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.1 Practicability</strong></td>
<td><strong>2.1 Operability</strong></td>
<td><strong>3.1 Affect</strong></td>
<td><strong>4.1 Helpfulness</strong></td>
<td><strong>5.1 Resilience</strong></td>
</tr>
<tr>
<td>1.1.1 Accessibility</td>
<td>2.1.1 Functionality</td>
<td>3.1.1 Consciseness</td>
<td>4.1.1 Portability</td>
<td>5.1.1 Manageability</td>
</tr>
<tr>
<td>1.1.2 Calibrability</td>
<td>2.1.2 Performance</td>
<td>3.1.2 Cost &amp; schedule</td>
<td>4.1.2 Localization</td>
<td>5.1.1.1 Maintainability</td>
</tr>
<tr>
<td><strong>1.2 Learnability</strong></td>
<td>2.1.3 Interoperability</td>
<td>3.1.3 Economy</td>
<td>4.1.3 Supportability</td>
<td>5.1.1.2 Flexibility</td>
</tr>
<tr>
<td>1.2.1 Understandability</td>
<td>2.1.4 Fault tolerance</td>
<td>3.1.4 Marketability</td>
<td>4.2 Validity</td>
<td>5.1.1.3 Testability</td>
</tr>
<tr>
<td>1.2.2 Completeness</td>
<td>2.1.5 Simplicity</td>
<td>3.1.5 Appropriateness of organization</td>
<td>4.2.1 Availability</td>
<td><strong>5.2 Adaptability</strong></td>
</tr>
<tr>
<td>1.2.3 Scalability</td>
<td><strong>2.2 Efficiency</strong></td>
<td><strong>3.2 Customizability</strong></td>
<td>4.2.1.1 Reliability</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.1 Utility</td>
<td>3.2.1 Reusability</td>
<td>4.2.2 Accuracy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.2 Response Time</td>
<td>3.2.2 Extensibility</td>
<td>4.2.3 Correctness</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.3 Throughput behaviour</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.4 Sensibility</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.2.5 Stability</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2.3 Access Control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3.1 Security</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3.2 Integrity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.3.3 Repeatability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>2.4 Un-ambiguity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4.1 Consistency</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4.2 Clarity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4.3 Interoperability</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fig.-2.14: Multidimensional Dynamism of a Quality Attribute
Dynamic Behavior of Attributes

We present the dynamic role/behavior of quality attributes in different perspective-based classifications in this Section. Although, the definition of some quality attributes remain same (irrespective of the class in which these attributes lie) but due to the change in environment, behavior of these attributes change dynamically. As an effect, the impact of attributes plays an important role in the classification. To realize the change of role and the effect, we consider the above said twelve usability attributes in order and refer the behavior in both of the classifications. First, the role of access control in generalized classification is defined as subordinate attribute to usability attribute in runtime class whereas in usability attributes' classification, it is defined as a major attribute of design class with suitable contribution in usability. Another one is adaptability, which has been defined as a non-runtime attribute subordinate to manageability in generalized classification and is considered as post implementation classes attribute in classification of usability attributes and act as adapter for external and environmental changes.

Affect is the next attribute that belongs to the impact class in generalized classification while the same attribute being an attribute of business class measures organization wide operations in classification of usability attributes. Next attribute is customizability that reflects its behavior as a runtime and subordinate attribute in generalized classification, but is considered as business classes attribute in usability attributes' classification due to perpetually related to the customer requirements and satisfaction. Efficiency being essential for usability is a runtime and subordinate attribute generalized classification instead a major design class attribute in classification of usability attributes. Consequently, in generalized classification, helpfulness has been defined as an impact class attribute while it is defined as conformance class usability attribute responsible for providing support to conform system usage as per classification of usability attributes. The learnability attribute is present in generalized classification as runtime and subordinate attribute and records its role as one of the main conceptual class attribute in classification of usability attributes.
Operability is the attribute defined as sub-attribute of usability in runtime class of attributes in generalized classification whereas plays an important role being design class attribute as defined in usability attributes’ classification. Practicability is concerned with usability as subordinate attribute in runtime class as mentioned in generalized classification.

On the other hand, it is a conceptual class attribute and to be considered from the conception phase of the development in classification of usability attributes., Resilience is also a subordinate attribute of usability in runtime class of attributes in generalized classification. It is the attribute considered as attribute of post implementation class in classification of usability attributes with managing misshaping of the system. Last but one is the un-ambiguity attribute defined as again subordinate of usability attribute in generalized classification while in usability attributes’ classification it is defined as a key design attribute. Lastly, validity is the attribute performing role of subordinate to usability attribute of runtime class in generalized classification but is assigned role of being a conformance class attribute in classification of usability attributes.

The role of above mentioned twelve software quality attributes with respect to the views corresponding to generalized classification and classification in usability perspective is shown in Table-2.3 in a summarized manner.

2.6 BENEFITS

In the proposed generalized classification, we have made efforts to reconcile the limitations of the existing classification approaches with the stated aim. Certain attributes such as functionality, performance, usability, portability, reusability have been reorganized in perspective of suitability for improvisation of quality in respective class. Also, the proposed quality classification exclusively focuses on business and economy view of the system. In addition, impact oriented attributes have been incorporated to extend the overall quality of the system. Thus, the classification proves to be highly beneficial to produce quality software, as it is a unified approach to make use of the attributes for the product, process, policy, plan and service. It will provide ease of quality inclusion and assurance in software development. It may be concluded that generalized classification of software quality
<table>
<thead>
<tr>
<th>S. No.</th>
<th>Quality/Usability Attributes</th>
<th>Generalized Classification</th>
<th>Classification in Usability Perspective</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Access Control</td>
<td>A runtime attribute and subordinate attribute to usability attribute. Measured at execution time.</td>
<td>A design attribute and helps users to understand type and manner access given to them. Security, integrity and repeatability are contributing in assessment of access control.</td>
</tr>
<tr>
<td>2</td>
<td>Adaptability</td>
<td>A non-runtime attribute. Measured off line.</td>
<td>A post implementation attribute and act as adapter for external and environmental changes.</td>
</tr>
<tr>
<td>3</td>
<td>Affect</td>
<td>An impact oriented attribute. Measures influence of the software on users and used to improve the impact in all aspects of software functionality.</td>
<td>A business attribute, measures organization wide operations.</td>
</tr>
<tr>
<td>4</td>
<td>Customizability</td>
<td>A runtime attribute and subordinate attribute to usability. Measured at execution time.</td>
<td>A business attribute, ability for software to be changed by the user or programmer.</td>
</tr>
<tr>
<td>5</td>
<td>Efficiency</td>
<td>A runtime attribute and subordinate attribute to usability attribute. Measured at execution time.</td>
<td>A design attribute that shows the relationship between the level of performance of the software and the amount of resources used.</td>
</tr>
<tr>
<td>6</td>
<td>Helpfulness</td>
<td>An impact oriented attribute. Measures influence of the software on users and used to improve the impact in all aspects of software functionality.</td>
<td>A conformance attribute to provide support, portability and localized application to users to conform the usage of the software.</td>
</tr>
<tr>
<td>7</td>
<td>Learnability</td>
<td>A runtime attribute and subordinate attribute to usability. Measured at execution time.</td>
<td>A conceptual attribute. It has to be considered from conception of the system to get usable software.</td>
</tr>
<tr>
<td>8</td>
<td>Operability</td>
<td>A runtime attribute and subordinate attribute to usability attribute. Measured at execution time.</td>
<td>A design attribute and deals with the ability to keep a system in a functioning and operating condition.</td>
</tr>
<tr>
<td>9</td>
<td>Practicability</td>
<td>A runtime attribute and subordinate attribute to usability attribute. Measured at execution time.</td>
<td>A conceptual attribute aims at providing easy to use system.</td>
</tr>
<tr>
<td>10</td>
<td>Resilience</td>
<td>A runtime attribute and subordinate attribute to usability attribute. Measured at execution time.</td>
<td>A Post implementation attribute and ability to maintain the system for any unfortunate happening.</td>
</tr>
<tr>
<td>11</td>
<td>Un-ambiguity</td>
<td>A runtime attribute and subordinate attribute to usability attribute. Measured at execution time.</td>
<td>A design attribute and provides ability to understand the system by all its users in the stated manner.</td>
</tr>
<tr>
<td>12</td>
<td>Validity</td>
<td>A runtime attribute and subordinate attribute to usability attribute. Measured at execution time.</td>
<td>A Post implementation issue and measures of logical, analytical or necessary trueness of the system.</td>
</tr>
</tbody>
</table>
attributes provides a structured and integrated classification of quality attributes covering aspects related to software, software process, organizational policies and service.

Further, we have presented a classification of software quality attributes in which usability plays a vital role rather than simply a quality characteristic. Here, usability has been represented in terms of hierarchy of other quality attributes termed as usability attributes. The classification provides five major classes namely; Conceptual class, Design class, Business class, Conformance class and Post implementation class. These classes are formed based on the usability considerations for software development process. It has offered a dimension of quality in view of user satisfaction. It will lead to provide the most usable software along with the process of usable software development. This classification will be helpful in usability research domain for finding impact of usability attributes in usable software and its development. Also, it will provide guideline to understand usability issues and will support to find measures of usability. Further, usability metrics can be determined for the product and process both. In proposed classification, we have explicitly considered phase wise software development process. And hence, it does not focus on scalability aspect of the system. It provides systematic organization of usability attributes for software usability assurance and also will be highly useful for developing/ computing measures associated with software usability.

Dynamic role and behavior of software quality attributes put emphasis on the development of software. The perspective may change the effect of attributes during the development. It will be extremely useful for developer in at least three folds. First, it will be helpful to identify the major attributes to be involved in the software development in different perspectives. Second, it will be useful to understand the changes at attribute level as per the view. And third, it will be highly useful to develop perspective specific software measures according to the goal and objectives of the development with a specific view.
2.7 SUMMARY

In this Chapter, we have attempted to provide classifications of software quality attributes namely; generalized classification and classification in perspective of usability. End-users perceive quality in software products and services which must meet the requirements and continue to do so till the use of software. Techniques, tools, and procedures can improve software quality, if deployed in an appropriate environment and are especially important for the developers. In this view, we have presented the current state of art of software quality attributes and usability attributes with related literature survey. Our generalized classification provides the structured organization of quality attributes in six major classes and covers the quality measurements in terms of process, product and service related to software. On the other hand, usability being directly associated with the user satisfaction, our classification in usability perspective with five major classes has provided sufficient exposure to the usability attributes in the stages during the software development process. The umbrella properly covers the attributes from the aspect of usability in the software product by considering the process and the services to both the organization and the users simultaneously. Thus, we can conclude that usability has impact during the software development environment. Also, the dynamic behavior of quality attributes in different perspectives has been discussed in this Chapter. It has been observed that the perspective affects the quality attributes in terms of their role and impact in software development process. We have concluded the chapter mentioning benefits of the classifications and dynamism of software quality attributes.