CHAPTER 5

THE Q’FACTO 12 QUALITY MODEL

5.1 INTRODUCTION

The objective of this research is to build a comprehensive model that can be used by end users or consumers for the evaluation of COTS components. Our first proposal is based on the ISO 9126 quality model. The model has ten quality factors and hence it was named the Q’Facto 10 model. The proposed quality model was given for peer review and critical comments were received by anonymous reviewers of the published papers. The reviewers have made an excellent suggestion and based on the reviewers suggestion, we moved from Q’Facto 10 to Q’Facto 12 based on the ISO 25000 [SQuaRE] quality standard.

Since the ISO 25000 is the current quality standard followed and ISO 25000 has quality factors and quality measures that do more justice to end user quality evaluation requirements, it becomes necessary to upgrade the Q’Facto 10 quality model to the ISO 25000 quality standard. The upgraded model has twelve quality factors and is hence called the Q’Facto 12 quality model.

This chapter discusses the ISO 25000 [SQuaRE] quality standard, compares the Q’Facto 10 and Q’Facto 12 quality models and outlines the proposed Q’Facto12 Model. The chapter also discusses the procedure for quality factor calculation using inputs given from software professionals using the product.
5.2 ISO 25000 [SQuaRE]

ISO (The International Organization for Standardization) and IEC (the International Electro Technical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental in liaison with ISO and IEC also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. ISO/IEC 25000 was prepared by Joint Technical Committee ISO/IEC JTC 1 and Information Technology Subcommittee SC 7.

It has to be stressed that the SQuaRE series of International Standards is dedicated to software product quality only. SQuaRE ISO/IEC 25000n — Quality Management Division (ISO/IEC 2005) addresses software product quality requirements specification, measurement and evaluation and is separate and distinct from the "Quality Management" of processes which is defined in the ISO 9000 family of standards.

The major benefits of the SQuaRE series over its predecessor standards include:

1) the coordination of guidance on software product quality measurement and evaluation

2) guidance for the specification of software product quality requirements and
3) harmonization with ISO/IEC 15939 in the form of Software product Quality Measurement Reference

The SQuaRE specification consists of the following five divisions:

1) ISO/IEC 2500n - Quality Management Division,
2) ISO/IEC 2501n - Quality Model Division,
3) ISO/IEC 2502n - Quality Measurement Division,
4) ISO/IEC 2503n - Quality Requirements Division, and
5) ISO/IEC 2504n - Quality Evaluation Division,

We use the ISO/IEC 2501n technical report as a framework for our quality model (ISO/IEC 2005). SQuaRE replaces the current ISO/IEC 9126 series and the 14598 series. Figure 5.1 illustrates the SQuaRE architecture.

![Diagram of SQuaRE Architecture]

Figure 5.1 SQuaRE Architecture
5.3  Q'FACTO 12 AND Q'FACTO 10 COMPARISONS

The Q’Facto 12 model has the following advantages over the Q’Facto 10 quality model:

1) Firstly, the model is based on the ISO 2500 quality standard which is a higher version of the ISO 9126 quality standard which the Q’Facto 10 model is based upon. The ISO 25000 [SQuaRE] quality standard is a combination of the ISO9126 - 1 and the ISO9126 – 2 standards.

2) Secondly, the ISO 25000 quality standard defines quality factors in three categories: internal product measures, external product measures and quality in use measures like ISO 9126. However, the third category quality in use measures in ISO 25000 includes quality factors like Usability in Use and Safety in Use which are aptly suited for measuring COTS components quality from the end users point of view and therefore have been included in the model.

3) Thirdly, in order to manage the uncertainty associated with assessment, the Likert’s scale has been used for level assessment of certain quality measures. Uncertainty is inherent in assessment. Any assessment is conducted with limited resources and under various assumptions which may not hold true in real operations. To overcome this uncertainty it is better to use a subjective assessment. Therefore the Likert’s scale is used for level assessment in Q'Facto12 model.
4) Fourthly, changes have been made to the Maintainability, Portability, Security and Usability quality factors. The changes are elaborated in the next section.

5.4 ADDITIONAL AND MODIFIED SCHEMATICS

We have already discussed in detail in Chapter 4, the schemas for the first ten quality factors and the motivation behind including each quality factor. We shall now discuss the schemas for the two quality factors added to the Q’Facto 12 quality model namely Usability in Use and Safety in Use. We already have Usability as a quality factor in the model and therefore we were initially against including the above two factors since they are having the same ambiguous names. However, the quality criteria under these attributes are different in the ISO specification and therefore Usability, Usability in Use and Safety in Use have been included as separate quality factors. This keeps our model in harmony with ISO 2500.

5.4.1 Quality Factor Usability in Use

Usability in Use refers to the elegance and clarity with which the end user can interact with the component (ISO/IEC 2005).

![Figure 5.2 Schematic for Usability in Use](image-url)
The quality criteria **Satisfaction in Use** and **Effectiveness in Use** have been chosen as the quality criteria for this quality factor. This is because a user of a COTS component should be satisfied and should find the component effective in the integrated system. The quality measures are **Satisfaction Level** and **Effectiveness Level**. They are both level measures.

### 5.4.2 Quality Factor Safety in Use

Safety in Use refers to how safe the software is when used by end users (ISO/IEC 2005). We felt that considering the incidents discussed in chapter 1, it is important to measure the safety of the COTS component and therefore have included Safety in Use has a quality factor. We include **Risk of Software** as a quality criterion. Risk of software relates to any risk that might occur to the software to which the component will be integrated. The quality measure is Software Risk Level which is a level measure.

![Figure 5.3 Schematic for Usability in Use](image)

### 5.4.3 Modified Schematic for Maintainability and Portability

In the Q’Facto 10 quality model, Maintainability had three quality criteria, Stability, Ease of Migration and Ease of Replacement. We decided that Ease of Replacement was more suited to be a quality criterion under portability and so we have shifted the quality criteria Ease of Replacement and the quality measure Replacement Ease Level to the quality factor Portability. Also, since reviewers suggested that complexity is difficult to measure, we have removed the measures installability complexity and
deployability complexity. The modified schematics for Maintainability and Portability are shown in Figures 5.4 and 5.5 respectively.

![Diagram of Maintainability](image)

**Figure 5.4 Modified Schematic for Quality Factor Maintainability**

![Diagram of Portability](image)

**Figure 5.5 Modified Schematic for Quality Factor Portability**

#### 5.4.4 Modified Schematic for Usability

Changes have been made to the Usability quality factor in the ISO 25000 specification which has been incorporated in the Q'Facto 12 model.
Usability now has three quality criteria: learnability, helpfulness, and operability. Learnability has the quality measures **Training** and **Presence of Demonstration**. Helpfulness has the quality measures **Presence of Context Sensitive Help** and **Presence of Dialog Boxes**. Operability has the quality measure **Understandability**. Figure 5.6 shows the schematic for the Usability quality factor.

![Modified Schematic for Quality Factor Usability](image)

**Figure 5.6 Modified Schematic for Quality Factor Usability**

**5.4.5 Modified Schematic for Security**

The ISO 25000 specification has included a quality criteria **Cipherability** which evaluates the encryption and decryption capabilities of the software. We felt that a COTS component should also have such capabilities for added security. We have therefore added **Cipherability** as a quality criterion and **Data Encryption** as a quality measure under the Cipherability quality criteria as shown in Figure 5.7.
5.5 THE Q’FACTO 12 QUALITY MODEL

As already mentioned the Q’Facto 12 model is an upgrade of the Q’Facto 10 model and is based on the ISO 25000 quality standard. The model has twelve quality factors, twenty eight quality criteria and 40 quality measures. The first three assumptions discussed in section 4.2 in chapter 4 made for the Q’Facto 10 model hold good for the Q’Facto 12 model also. The quality factors, quality criteria and quality measures of the proposed model are outlined below:

5.5.1 Quality Factor 1 – Functionality

This characteristic expresses the ability of a component to provide the required services and functions that meet stated and implied needs when the component is used under specified conditions.

5.5.1.1 Quality criteria (1, 1) - self-contained

Szysperski (2007) mentions that one of the characteristic properties of a component is that it should be self-contained. It is an intrinsic property of the component and it means that the component is encapsulated with well-defined interfaces and can be executed independently with minimal outside support. Well-defined interfaces have contracts that are described by the
presence of preconditions and post conditions (Szyperski 2007). The quality measures are:

1. **Quality Measure (1, 1, 1) - Presence of preconditions and Post conditions**

   This metric indicates if each interface in the component has well defined preconditions and post conditions which determine what the component provides and requires.

   Presence of preconditions and Post conditions
   
   \[ = \{ 1 \text{ if present} \ 0 \text{ if absent} \} \quad (5.1) \]

2. **Quality Measure (1, 1, 2) - Modularity**

   Modularity measures the degree of independence of the component, which is the degree of functional cohesion

   \[ \text{Modularity} = \frac{\text{Functions provided by the component itself}}{\text{Total functions provided by the component}} \quad (5.2) \]

**5.5.1.2 Quality criteria (1, 2) - compliance:**

The attributes of the component that make the software adhere to application related standards or conventions or regulations in laws and similar prescriptions. The quality measures are:

1. **Quality Measure (1, 2, 1) - Presence of Standardization**

   This indicates if the component conforms to international standards.

   Presence of Standardization = \{ 1 \text{ if present} \ 0 \text{ if absent} \} \quad (5.3)
2. **Quality Measure (1, 2, 2) - Presence of Certification**

   This indicates if the component has been certified by an internal or external organization.
   
   Presence of Certification = \{1 \text{ if present} \quad 0 \text{ if absent}\} \quad (5.4)

5.5.1.3 **Quality criteria (1, 3) - accuracy**

   Accuracy is the ability of the component to execute efficiently and deliver the expected results. The quality measure is:

1. **Quality Measure (1, 3, 1) - Correctness**

   Correctness evaluates the number of correct results returned by the component operations according to the user specifications.

   \[
   \text{Computational accuracy} = \frac{\text{Accurate Results}}{\text{Total Results}} \quad (5.5)
   \]

5.5.2 **Quality factor 2 – security**

   The security of the component relates to its ability to prevent unauthorized access whether accidental or deliberate to programs or data.

5.5.2.1 **Quality criteria (2, 1) - access control**

   This attribute has also been slightly refined and added to the model. This attribute indicates how the component is able to control access to its provided interfaces. An example can be a component that provides interfaces with the functionality to identify or authenticate users. The quality measure is:
1. **Quality Measure (2, 1, 1) - Access Controllability:**

Access controllability measures whether access control mechanisms are implemented or not and if implemented followed by a description of the mechanism is available.

Access Controllability = \{1 \text{ if present}\ 0 \text{ if absent}\} \quad (5.6)

5.5.2.2 **Quality criteria (2, 2) - cipherability**

This attribute indicates whether the COTS component has encryption and decryption capabilities. The quality measure is:

1. **Quality Measure (2, 2, 1) – Data Encryption**

Data Encryption measures the ability of the component to deal with encryption/decryption in order to protect the data it handles.

Data Encryption = \{1 \text{ if present}\ 0 \text{ if absent}\} \quad (5.7)

5.5.3 **Quality factor 3 – interoperability**

The attributes of the component that reflect it’s ability to interact with other systems whether they are composed of components or not.

5.5.1 **Quality Criteria (3, 1) - Compatibility**

This is the attribute that tells us whether the data format or versions of the component is compatible with that of interacting components. The quality measures are:
1. **Quality Measure (3, 1, 1) - Version Compatibility**

   Version Compatibility indicates if the component is compatible with backward and forward versions of the same component.

   \[
   \text{Version Compatibility} = \begin{cases} 
   1 & \text{if present} \\
   0 & \text{if absent}
   \end{cases} \quad (5.8)
   \]

2. **Quality Measure (3, 1, 2) - Software Compatibility**

   Software Compatibility indicates if the component is compatible with backward and forward versions of the software needed to run the component.

   \[
   \text{Software Compatibility} = \begin{cases} 
   1 & \text{if present} \\
   0 & \text{if absent}
   \end{cases} \quad (5.9)
   \]

5.5.4 **Quality Factor 4– Reliability**

   Reliability is the capability of the component to maintain a specified level of performance when used in stated conditions in a stated period of time. Reliability can be assessed by measuring the frequency and severity of failure, the accuracy of output results, the mean time between failures and the ability to recover from failure and the predictability of the program (Rawesdah and Matelkah 2006).

5.5.4.1 **Quality criteria (4, 1) - recoverability**

   The attributes of the software that relate to the capability to reestablish its level of performance and recover the data directly affected in case of failure and the time and effort needed for it. The quality measures are:
1. **Quality Measure (4, 1, 1) - Persistence**

Persistence denotes the ability of the component to store its state in a persistent manner for later recovery.

\[ \text{Persistence} = \begin{cases} 1 & \text{if present} \\ 0 & \text{if absent} \end{cases} \quad (5.10) \]

2. **Quality Measure (4, 1, 2) – Presence of Exception Handling**

Presence of Exception Handling indicates whether exception handling mechanisms have been implemented in the COTS component.

\[ \text{Presence of Exception Handling} = \begin{cases} 1 & \text{if present} \\ 0 & \text{if absent} \end{cases} \quad (5.11) \]

5.5.4.2 **Quality criteria (4 , 2) – fault tolerant mechanism**

The attributes of the COTS component that relate to its ability to maintain a specified level of performance in case of software faults or infringements of its specified interfaces. The quality measure is:

1. **Quality Measure (4, 2, 1) - Presence of Fault Tolerant Mechanism**

Presence of Fault Tolerant Mechanism indicates whether fault tolerant mechanisms have been implemented in the component.

\[ \text{Presence of Fault Tolerant Mechanism} = \begin{cases} 1 & \text{if present} \\ 0 & \text{if absent} \end{cases} \quad (5.12) \]

5.5.5 **Quality Factor 5 – Efficiency**

It is the capability of a component to provide appropriate performance relative to the amount of resources used under stated conditions. Efficiency is usually measured in terms of time and resource usage. Hence,
efficiency measurement of a software component will be just the same as for any other software product.

5.5.5.1 Quality criteria (5, 1) - resource behavior

The attributes of the component that relate to the amount of resources used and the duration of each use in performing its functions. The quality measure is:

1. Quality Measure (5, 1, 1) - Disk Capacity

Disk Capacity measures the disk capacity used by the component. A level measure is used with five levels: 1 for very low usage (1.0), 2 for low usage to replace (0.8), 3 for neither high nor low usage (0.6), 4 for high usage (0.4) and 5 for very high usage (0.2). The figures in the brackets indicate the values assigned for each level when calculating the overall quality.

5.5.5.2 Quality criteria (5, 2) - time behavior

The attributes of the component that relate to response and processing times and throughput rates in performing its functions. The quality measure is:

1. Quality Measure (5, 2, 1) –Response Time

Response Time measure the time taken since a request is received until a response has been sent and is indicated using a Time metric. A level measure is used with five levels: 1 for very quick response time (1.0), 2 for quick response time (0.8), 3 for neither quick nor slow response time (0.6), 4 for slow response time (0.4) and 5 for very slow response time (0.2). The figures in the brackets indicate the values assigned for each level when calculating the overall quality.
5.5.6 **Quality Factor 6 -- Maintainability**

Maintainability is the effort required to replace a COTS component with the corrected version and to migrate an existing, software component from a current component based software system to a new version of the system (Gao et al 2002).

5.5.6.1 **Quality criteria (6, 1) - stability**

Stability refers to the attributes of the COTS component that relate to the risk of unexpected modifications. Rawsdah and Matelkah (2006) mention that the greater the stability of the COTS components, the lesser maintenance required is required. The quality measure for stability is:

1. **Quality Measure (6, 1, 1) - Component Stability**

   We propose a COTS Component Stability (CCS) metric that provides an indication of the ability of a software product (based on changes that occur for each version of the software component) based on the Software Maturity Index metric proposed in IEEE std.982.1-1988.

   \[
   CCS = \frac{[M(t) - (F(a) + F(c) + F(d))]}{M(t)} \tag{5.13}
   \]

   - \( M(t) \) = the number of interfaces provided in the current version
   - \( F(c) \) = the number of interfaces in the current version that have been changed
   - \( F(a) \) = the number of interfaces in the current version that have been added
   - \( F(d) \) = the number of interfaces from the preceding release that were deleted in the current release.

   As the CS approaches 1.0, we can say that the software component has stabilized.
5.5.6.2 Quality criteria (6, 2) - ease of migration

The attributes of the COTS component that relate to the effort required to migrate an existing, software component from a current component based software system to a new version of the system. The quality measures for ease of migration are:

1. Quality Measure (6, 2, 1) - Migration Ease level

   A level measure is used with five levels: 1 for very easy (1.0), 2 for easy (0.8), 3 for slightly difficult (0.6), 4 for very difficult (0.4) and 5 for not at all possible (0.2). The figures in the brackets indicate the values assigned for each level when calculating the overall quality.

5.5.7 Quality Factor 7 –Portability

Portability is the ability of a component to be transferred from one environment to another. Portability is an intrinsic property to the nature of components, which are in principle designed and developed to be reused in different environments (Bertoa and Valecillo 2002).

5.5.7.1 Quality criteria (7, 1)- installability

The attributes of the component that relate to the effort needed to install the component in a specified environment. The quality measure is:

1. Quality Measure (7, 1, 1) - Installability Documentation

   Installability Documentation indicates if the component developer has provided the component user proper documentation outlining the installation steps.

   \[ \text{Installability Documentation} = \begin{cases} 1 & \text{if present} \\ 0 & \text{if absent} \end{cases} \quad (5.14) \]
5.5.7.2 Quality criteria (7, 2) - deployability

The attributes of the component that relate to the effort needed to deploy the component in a specified environment. The quality measure is:

1. Quality Measure (7, 2, 1) - Deployability Documentation

   Deployability Documentation indicates if the component developer has provided the component user proper documentation outlining the deployment steps.

   \[ \text{Deployability Documentation} = \{1 \text{ if present} | 0 \text{ if absent}\} \quad (5.15) \]

5.5.7.3 Quality criteria (7, 3) - adaptability

   The attributes of the software that relate to on the opportunity for it’s adaptation to different specified environments without applying other actions or means than those provided for the component to be considered. The quality measure is:

1. Quality Measure (7, 3, 1) - Mobility

   Mobility indicates whether the component can be deployed in any other container other than the one it was transferred in.

   \[ \text{Mobility} = \{1 \text{ if present} | 0 \text{ if absent}\} \quad (5.16) \]

5.5.7.4 Quality criteria (7, 4) - replaceability

   The attributes of the COTS component that relate to the effort required to replace a COTS component with the next version. The quality measure for ease of replaceability:
1. **Quality Measure (7, 4, 1) - Replacement Ease level**

   Replacement Ease Level measures the ease with which the COTS component can be replaced. A level measure is used with five levels: 1 for very easy to replace (1.0), 2 for easy to replace (0.8), 3 for neither easy nor difficult (0.6), 4 for very difficult to replace (0.4) and 5 for very very difficult to replace (0.2). The figures in the brackets indicate the values assigned for each level when calculating the overall quality.

5.5.8 **Quality Factor 8 – Testability**

   The testability of software components is one of the most important factors determining the quality of components. According to Gao and Wu (2003), building components with good testability simplifies test operations, reduces test costs and increases software quality. This is more important in the case of COTS components because the components are developed and tested in a totally different site from the place of use of the component. The component buyers usually get only the executable file and a few documents with which they have to carry out the integration testing.

5.5.8.1 **Quality criteria (8, 1) - test documentation**

   The attributes of the COTS component that demonstrate whether proper documentation is available for the testing process. This includes the presence of a well-documented test suite, user’s manual, well-detailed test cases, description of the testing environment and also the milestones to be encountered in the testing process. The quality measures are:

1. **Quality Measure (8, 1, 1) - Test Suit documentation**

   Test Suit documentation measures if whether proper documentation has been provided for plans, tests cases, testing milestones, testing environment etc.

   Test Suit Documentation = \{ 1 if present \ 0 if absent \} \quad (5.17)
2. Quality Measure (8, 1, 2) - Proofs of Previous Tests

Proofs of Previous Tests measures if the history of previous tests has been provided which demonstrate how successful or unsuccessful the tests were.

Proofs of previous tests = \{1 \text{ if present}, 0 \text{ if absent} \} \quad (5.18)

5.5.8.2 Quality criteria (8, 2) - component controllability

The attributes of the COTS component that indicate how easy is it to control the program or component on its inputs /outputs, operations and behavior. Component users look at the “controllability “of a COTS component in three aspects: a) behavior control b) feature customization and c) installation and deployment. Controllability is important because it directly affects testability. Gao and Wu (2003) say that testers and customers expect components to provide a set of control functions in software so that they can use them to check and monitor diverse component behaviors according to their needs. The quality measures are:

1. Quality Measure (8, 2, 1) - Component Execution Control

This metric measures whether the component allows the user to control its entire execution from beginning to end. It means that it is possible for the user to start, stop and rerun the execution of the component according to their wish.

Component Execution Control = \{1 \text{ if present}, 0 \text{ if absent} \} \quad (5.19)

2. Quality Measure (8, 2, 2) - Component Environment Control

These metric measures whether the component allows the user to control its interaction with the environment in which it is executing.

Component Environment Control = \{1 \text{ if present}, 0 \text{ if absent} \} \quad (5.20)
3. **Quality Measure (8, 2, 3)-Component Function Feature Control**

   This metric measures whether the component allows the user to switch control between the different functionalities offered by the component while the component is executing.

   \[
   \text{Component Function Feature Control} = \begin{cases} 
   1 & \text{if present} \\
   0 & \text{if absent}
   \end{cases} \quad (5.21)
   \]

5.5.8.3 **Quality criteria (8, 3) - traceability**

   The attributes of the COTS component that refers to the extent of its built in capacity of tracking the status of component attributes and component behavior. In the real world, engineers have trouble carrying out operation and state traces in COTS components due to the inaccessibility of the source code. A few research efforts that allow COTS components to be tracked by adding a program tracking mechanism (Gao et al 2002). However, they are not so successful, due to the lack of standardized component trace formats and trace mechanisms (Gao and Wu 2003). Performance trace and error trace can be made possible with the help of log files. The quality measures are:

1. **Quality Measure (8, 3, 1) - Performance Trace**

   Performance Trace describes whether the component allows the user to perform a performance trace of the component execution.

   \[
   \text{Performance Trace} = \begin{cases} 
   1 & \text{if present} \\
   0 & \text{if absent}
   \end{cases} \quad (5.22)
   \]

2. **Quality Measure (8, 3, 2) - Error Trace**

   Error Trace describes whether the component allows the user to perform an error trace of the component execution.

   \[
   \text{Error Trace} = \begin{cases} 
   1 & \text{if present} \\
   0 & \text{if absent}
   \end{cases} \quad (5.23)
   \]
5.5.9 Quality Factor 9–Reusability

According to Hopkins John (2000), reusability is important in the development of a component-based system. Hopkins John (2000) says that in the context of component based software engineering; reusability can refer to the ability to reuse existing components to create a more complex system. There is a great demand for reusable components in the component market. Reusability is also a major driving force of the component market (Hopkins Jon 2000). There are expectations that a high quality component has to be reusable. Hence, it becomes necessary to include reusability as a quality factor in our quality model.

5.5.9.1 Quality criteria (9, 1) - generality

A reusable component has to be generic in nature. Generality is the set of attributes that determine how reusable the component and it’s capability of extending specific interaction knowledge to new situations. The quality measures are:

1. Quality Measure (9, 1, 1) - Presence of Domain Abstraction

Presence of Domain Abstraction measures whether the component can be reused across several domains related to the specific functionality that the component offers.

\[
\text{Presence of Domain Abstraction} = \begin{cases} 1 & \text{if present} \\ 0 & \text{if absent} \end{cases} \quad (5.24)
\]

2. Quality Measure (9, 1, 2) - History of Reuse

History of Reuse indicates if previous uses of the reusable component have been recorded and documented for future reference.

\[
\text{History of Reuse} = \begin{cases} 1 & \text{if present} \\ 0 & \text{if absent} \end{cases} \quad (5.25)
\]
5.5.9.2 Quality criteria (9, 2) - hardware/ software independence

It would be highly favorable for a reusable component not to depend on any particular hardware or software architecture.

1. Quality Measure (9, 2, 1) - Presence of Hardware Independence

Presence of Hardware Independence measures if the component is dependent or not dependent on any particular hardware.

Presence of Hardware Independence = \{1 \text{ if present} \mid 0 \text{ if absent}\} \quad (5.26)

2. Quality Measure (9, 2, 2) - Presence of Software Independence

Presence of Software Independence measures if the component is dependent or not dependent on any particular software.

Presence of Software Independence = \{1 \text{ if present} \mid 0 \text{ if absent}\} \quad (5.27)

5.5.9.3 Quality criteria (9, 3) - locatability

The attributes of the component that determine the ease with which the appropriate reusable component can be obtained from a component repository. The quality measures are:

1. Quality Measures (9, 3, 1)- Accessibility

Accessibility measures the ease with which the reusable component can be easily located from a component repository. A level measure is used with five levels: 1 for very easy to locate (1.0), 2 for easy to locate (0.8), 3 for slightly difficult to locate (0.6), 4 for very difficult to locate (0.4) and 5 for able to locate only with external support (0.2). The figures in the brackets
indicate the values assigned for each level when calculating the overall quality.

5.5.10  Quality Factor 10 – Usability

Usability is the capability of the component to be understood, learnt and used under specified conditions. Usability is the effort required to learn, operate, prepare input and interpret the output of the program. In the case of COTS and reusable COTS components only the component users can measure this characteristic because the component developers are generally third party vendors who have no connection whatsoever to the component users.

5.5.10.1  Quality criteria (10, 1) - learnability

The attributes of a software component that relate to the user’s efforts for learning its application (for example, operation control, input and output). The quality measures are:

1)  Quality Measure (10, 1, 1) - Training

   Training indicates whether training courses/materials are available for the component and if available whether information about the training is provided.

   \[
   \text{Training} = \begin{cases} 
   1 & \text{if present} \\
   0 & \text{if absent}
   \end{cases} \quad (5.28)
   
\]

2)  Quality Measure (10, 1, 2) – Presence of Demonstration

   Presence of Demonstration measures the percentage of the component services that are shown in the demo compared to the total number of provided services and interfaces.
Demonstration coverage = \{ 1 \text{ if present} | 0 \text{ if absent} \} \quad (5.29)

### 5.5.10.2 Quality Criteria 2 – Helpfulness

Helpfulness refers to the attributes of the software that relate to the user’s effort for learning the application. The quality measures are:

1) **Quality Measure (10, 2, 1) – Presence of Context Sensitive Help**

   This measures the quality of the Help system provided with the component for discovering and understanding its services in terms of its completeness, clarity and usefulness. A level measure is used with five levels: 1 for excellent (1.0), 2 for very good (0.8), 3 for good (0.6), 4 for average (0.4) and 5 for a poor Help system (0.2). The figures in the brackets indicate the values assigned for each level when calculating the overall quality.

2) **Quality Measure (10, 2, 2) – Presence of Dialog Boxes**

   This indicates whether appropriate dialog boxes with diagnostic messages pop up when errors take place.

   \[
   \text{Presence of Dialog Boxes} = \{ 1 \text{ if present} | 0 \text{ if absent} \} \quad (5.30)
   \]

### 5.5.10.3 Quality criteria 3 – operability

The attributes of the software component that relate to the user’s efforts for learning it’s application. The quality measures are:

1) **Quality Measure (10, 3, 1) – Understandability**

   Understandability measures whether appropriate documentation is present that will help the operator how to understand how to operate the component and also contain trouble shooting instructions.
Understandability = \{ 1 \text{ if documents are present} \\
0 \text{ if documents are absent} \} \quad (5.31)

5.5.11 Quality Factor 11: Usability in Use

Usability in use refers to the elegance and clarity with which the end user can interact with the component.

5.5.11.1 Quality criteria (11, 1) - Satisfaction in Use

The attributes of a software component that relate to the user’s efforts for learning its application (For example, operation control, input and output). The quality measures are:

1. Quality Measure (11, 1, 1) - Satisfaction Level

Satisfaction Level measures how satisfied the customer is while interacting with the component. The satisfaction level measure is used with five levels: 1 for very highly satisfied (1.0), 2 for highly satisfied (0.8), 3 for satisfied (0.6), 4 for low satisfaction (0.4) and 5 for very low satisfaction (0.2). The figures in the brackets indicate the values assigned for each level when calculating overall quality.

5.5.11.2 Quality criteria (11, 2)- Effectiveness in Use

The attributes of a software component that indicate how effective the component is. The quality measure is:
1. **Quality Measure (11, 2, 1) - Effectiveness Level**

   Effectiveness Level measures how effectively the component carry’s out its functionality. The Effectiveness level measure is used with five levels: 1 for very high effectiveness (1.0), 2 for high effectiveness (0.8), 3 for neither high nor low effectiveness (0.6), 4 for low effectiveness (0.4) and 5 for very low effectiveness (0.2). The figures in the brackets indicate the values assigned for each level when calculating overall quality.

5.5.12 **Quality Factor 12 : Safety in Use**

   **Safety in Use** refers to how safe the component is when used by end users.

5.5.12.1 **Quality criteria (12, 1) - risk of software**

   The attributes of a software component that relate to any risk that might occur to the software to which the component will be integrated. The quality measure is:

1. **Quality Measure (12, 1, 1) - Software Risk Level**

   **Software Risk Level** measure is used with five levels: 1 for very high risk level (0.2), 2 for high risk level (0.4), 3 for neither high nor low risk (0.6), 4 for low risk (0.8) and 5 for very low risk (1.0). The figures in the brackets indicate the values assigned for each level when calculating overall quality.

   The outline of the model is given in Figure 5.8
Figure 5.8 The Q’Facto 12 COTS Quality Model
5.6 ILLUSTRATIVE STUDY

An illustrative study showing how the quality factor has been calculated using the Q’facto 12 model is shown in the Tables 5.1 and 5.2 with two examples using inputs given by software professionals through our questionnaires. A uniform weight of 1 has been considered for all the quality factors.

Table 5.1 Quality Factor Calculation [Sample 1]

Name of the Component : Events
Name of Contributor : Mr. M.Ananth Raja ,
Designation & Company : Project Manager, Rushmore Consultancy Services

<table>
<thead>
<tr>
<th>S.No</th>
<th>Quality Measures</th>
<th>User Value</th>
<th>User Weight</th>
<th>Value * Weight</th>
<th>Quality Factor Measure QFi</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Presence of Preconditions and Post conditions ((I, I, I))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>A.2</td>
<td>Modularity ((I, I, 2))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A.3</td>
<td>Presence of Standardization ((I, 2, I))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A.4</td>
<td>Presence of Certification ((I, 2,2))</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>A.5</td>
<td>Correctness ((I, 3, I))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A.</td>
<td><strong>Quality Factor 1 (QF1-Functionality)</strong></td>
<td></td>
<td></td>
<td></td>
<td>0.8</td>
</tr>
<tr>
<td>B.1</td>
<td>Access Controllability ((2, I, I))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B.2</td>
<td>Data Encryption ((2, 2, 1))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>B.</td>
<td><strong>Quality Factor 2 (QF2 – Security)</strong></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>C.1</td>
<td>Version Compatibility ((3, 1,1))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C.2</td>
<td>Software Compatibility ((3, 1, 2))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C.</td>
<td><strong>Quality Factor 3 (QF3 – Interoperability)</strong></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>D.1</td>
<td>Presence of Exception Handling ((4,1,1))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>D.2</td>
<td>Persistence ((4, 1, 2))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>D.3</td>
<td>Presence of Fault Tolerant Mechanism ((4 2, 1))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>D.</td>
<td><strong>Quality Factor 4 (QF4-Reliability)</strong></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>E.1</td>
<td>Disk Capacity ((5 1, 1))</td>
<td>1</td>
<td>0.8</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>E.</td>
<td><strong>Quality Factor 5 (QF5-Efficiency)</strong></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>F.1</td>
<td>Component Stability ((6, 1, 1))</td>
<td>0.3</td>
<td>1</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>F.2</td>
<td>Migration Ease Level ((6, 2, 1))</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>F.</td>
<td><strong>Quality Factor 6 (QF6 -Maintainability)</strong></td>
<td></td>
<td></td>
<td></td>
<td>0.65</td>
</tr>
</tbody>
</table>
### Table 5.1 (Continued)

<table>
<thead>
<tr>
<th>S.No</th>
<th>Quality Measures</th>
<th>User Value</th>
<th>User Weight</th>
<th>Value Weight</th>
<th>Quality Factor Measure QF_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.1</td>
<td>Installability Documentation (7, 1, 1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>G.2</td>
<td>Deployability Documentation (7, 2, 1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>G.3</td>
<td>Mobility (7, 3, 1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>G.4</td>
<td>Replacement Ease Level (7,4, 1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>G.</td>
<td><strong>Quality Factor 7 (QF7- Portability)</strong></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H.1</td>
<td>Test Suit Documentation (8,1,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H.2</td>
<td>Proofs of Previous Tests (8,1,2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H.3</td>
<td>Component Execution Control (8,2,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H.4</td>
<td>Component Environment Control (8,2,2)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>H.5</td>
<td>Component Function Feature Control (8,2,3)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H.6</td>
<td>Performance Trace (8,3,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H.7</td>
<td>Error Trace (8,3,2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>H.</td>
<td><strong>Quality Factor 8 (QF8-Testability)</strong></td>
<td>0.857</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I.1</td>
<td>Presence of Domain Abstraction (9,1,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>I.2</td>
<td>History of reuse (9,1,2)</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>I.3</td>
<td>Presence of Hardware Independence (9,2,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>I.4</td>
<td>Presence of Software Independence (9,2,2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>I.5</td>
<td>Accessibility (9,3,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>I.</td>
<td><strong>Quality Factor 9 (QF9-Reusability)</strong></td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J.1</td>
<td>Training (10,1,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>J.2</td>
<td>Presence of Demonstration(10,1,2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>J.3</td>
<td>Presence of Context Sensitive Help(10,2,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>J.4</td>
<td>Effort To Configure (10,3,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>J.5</td>
<td>Understandability (10,3,2)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>J.</td>
<td><strong>Quality Factor 10 (QF10 –Usability)</strong></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K.1</td>
<td>Customer Satisfaction Level (11,1,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>K.2</td>
<td>Effectiveness Level(11,2,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>K.</td>
<td><strong>Quality Factor 11 (QF10-Usability in Use)</strong></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L.1</td>
<td>Software Risk Level (12,1,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>L.</td>
<td><strong>Quality Factor 12 (QF12-Safety in Use)</strong></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total Quality Factor = (\sum_{i=1}^{12} QF_i/12 =)</td>
<td>0.9255</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.2 Quality Factor Calculation [Sample 2]

Name of the Component : QTP Testing Tool
Name of Contributor : Mr. G.Pradeep
Designation & Company : Team Lead, Mphasis Technologies Ltd

<table>
<thead>
<tr>
<th>S.No</th>
<th>Quality Measures</th>
<th>User Value</th>
<th>User Weight</th>
<th>Value * Weight</th>
<th>Quality Factor Measure QFi</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Presence of Preconditions and Post conditions ((1, 1,1))</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>A.2</td>
<td>Modularity ((1, 1, 2))</td>
<td>0.6</td>
<td>0.6</td>
<td>0.36</td>
<td></td>
</tr>
<tr>
<td>A.3</td>
<td>Presence of Standardization((1, 2,1))</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>A.4</td>
<td>Presence of Certification ((1, 2,2))</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>A.5</td>
<td>Correctness ((1, 3, 1))</td>
<td>0.9</td>
<td>0.5</td>
<td>0.45</td>
<td></td>
</tr>
<tr>
<td>A.</td>
<td><strong>Quality Factor 1 (QF1-Functionality)</strong></td>
<td></td>
<td></td>
<td>0.9236</td>
<td></td>
</tr>
<tr>
<td>B.1</td>
<td>Access Controllability ((2, 1, 1))</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>B.2</td>
<td>Data Encryption ((2, 2, 1))</td>
<td>1</td>
<td>0.6</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>B.</td>
<td><strong>Quality Factor 2 (QF2 – Security)</strong></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C.1</td>
<td>Version Compatibility ((3, 1,1))</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>C.2</td>
<td>Software Compatibility((3, 1, 2))</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>C.</td>
<td><strong>Quality Factor 3 (QF3 – Interoperability)</strong></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>D.1</td>
<td>Presence of Exception Handling((4,1,1))</td>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>D.2</td>
<td>Persistence ((4, 1, 2))</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>D.3</td>
<td>Presence of Fault Tolerant Mechanism ((4 2, 1))</td>
<td>1</td>
<td>0.4</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>D.</td>
<td><strong>Quality Factor 4 (QF4-Reliability)</strong></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>E.1</td>
<td>Disk Capacity ((5, 1, 1))</td>
<td>0.4</td>
<td>0.3</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>E.</td>
<td><strong>Quality Factor 5 (QF5-Efficiency)</strong></td>
<td></td>
<td></td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>F.1</td>
<td>Component Stability ((6, 1, 1))</td>
<td>0.5</td>
<td>0.5</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>F.2</td>
<td>Migration Ease Level ((6, 2, 1))</td>
<td>0.8</td>
<td>0.8</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>F.</td>
<td><strong>Quality Factor 6 (QF6 -Maintainability)</strong></td>
<td></td>
<td></td>
<td>0.68</td>
<td></td>
</tr>
<tr>
<td>G.1</td>
<td>Installability Documentation ((7, 1, 1))</td>
<td>1</td>
<td>0.6</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>G.2</td>
<td>Deployability Documentation ((7, 2, 1))</td>
<td>0</td>
<td>0.6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>G.3</td>
<td>Mobility ((7, 3, 1))</td>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>G.4</td>
<td>Replacement Ease Level ((7,4, 1))</td>
<td>0.8</td>
<td>0.8</td>
<td>0.64</td>
<td></td>
</tr>
</tbody>
</table>
Table 5.2 (Continued)

<table>
<thead>
<tr>
<th>S.No</th>
<th>Quality Measures</th>
<th>User Value</th>
<th>User Weight</th>
<th>Value * Weight</th>
<th>Quality Factor Measure QFi</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.</td>
<td>Quality Factor 7 (QF7- Portability)</td>
<td></td>
<td></td>
<td></td>
<td>0.72</td>
</tr>
<tr>
<td>H.1</td>
<td>Test Suit Documentation (8,1,1)</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>H.2</td>
<td>Proofs of Previous Tests (8,1,2)</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>H.3</td>
<td>Component Execution Control (8,2,1)</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>H.4</td>
<td>Component Environment Control (8,2,2)</td>
<td>1</td>
<td>0.6</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>H.5</td>
<td>Component Function Feature Control (8,2,3)</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>H.6</td>
<td>Performance Trace (8,3,1)</td>
<td>1</td>
<td>0.8</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>H.7</td>
<td>Error Trace (8,3,2)</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>H.</td>
<td>Quality Factor 8 (QF8-Testability)</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>I.1</td>
<td>Presence of Domain Abstraction (9,1,1)</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>I.2</td>
<td>History of reuse (9,1,2)</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>I.3</td>
<td>Presence of Hardware Independence (9,2,1)</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>I.4</td>
<td>Presence of Software Independence (9,2,2)</td>
<td>1</td>
<td>0.3</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>I.5</td>
<td>Accessibility (9,3,1)</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>I.</td>
<td>Quality Factor 9 (QF9-Reusability)</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>J.1</td>
<td>Training (10,1,1)</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>J.2</td>
<td>Presence of Demonstration(10,1,2)</td>
<td>1</td>
<td>0.9</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>J.3</td>
<td>Presence of Context Sensitive Help(10,2,1)</td>
<td>0.8</td>
<td>0.8</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>J.4</td>
<td>Presence of Dialog Boxes (10,3,1)</td>
<td>1</td>
<td>0.7</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>J.5</td>
<td>Understandability (10,3,2)</td>
<td>1</td>
<td>0.6</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>J.</td>
<td>Quality Factor 10 (QF10 –Usability)</td>
<td></td>
<td></td>
<td></td>
<td>0.9487</td>
</tr>
<tr>
<td>K.1</td>
<td>Customer Satisfaction Level (11,1,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>K.2</td>
<td>Effectiveness Level(11,2,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>K.</td>
<td>Quality Factor 11 (QF10-Usability in Use)</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>L.1</td>
<td>Software Risk Level (12,1,1)</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>L.</td>
<td>Quality Factor 12 (QF12-Safety in Use)</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Total Quality Factor = \sum_{i=1}^{12} QFi /12 =</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.9726</td>
</tr>
</tbody>
</table>

5.7 CONCLUSION

A Q'facto 12 model has been designed using the ISO 25000 quality standard as a framework that can be used for evaluating COTS components. The model is an upgrade of the Q’Facto 10 model. The objective is to create a model that can be used for the evaluation of the quality of COTS components.
by end users before integration into existing software systems. Therefore it becomes necessary to get the model validated by professionals from the industry who use COTS components. The model validation becomes the topic of the next chapter.