Carvallo, J.P. [et al.]. A framework for selecting workflow tools in the context of composite information systems. A: International Workshop on Database and Expert Systems Applications. "Database and Expert Systems Applications: 15th International Conference, DEXA 2004: Zaragoza, Spain, August 30-September 3, 2004: proceedings". Berlín: Springer, 2004, p. 109-119.
The final authenticated version is available online at https://doi.org/10.1007/978-3-540-30075-5_11

A Framework for Selecting Workflow Tools in the Context of Composite Information Systems

Juan P. Carvallo, Xavier Franch, Carme Quer, Nuria Rodríguez

Universitat Politècnica de Catalunya (UPC)
c/Jordi Girona 1-3, Campus Nord, mòdul C6, 08034 Barcelona
{carvallo | franch | cquer}@lsi.upc.es, nuria.rodriguez-camara@upc.es

Abstract. When an organization faces the need of integrating some workflow-related activities in its information system, it becomes necessary to have at hand some well-defined informational model to be used as a framework for determining the selection criteria onto which the requirements of the organization can be mapped. Some proposals exist that provide such a framework, remarkably the WfMC reference model, but they are designed to be applicable when workflow tools are selected independently from other software, and departing from a set of well-known requirements. Often this is not the case: workflow facilities are needed as a part of the procurement of a larger, composite information system and therefore the general goals of the system have to be analyzed, assigned to its individual components and further detailed. We propose in this paper the MULTSEC method in charge of analyzing the initial goals of the system, determining the types of components that form the system architecture, building quality models for each type and then mapping the goals into detailed requirements which can be measured using quality criteria. We develop in some detail the quality model (compliant with the ISO/IEC 9126-1 quality standard) for the workflow type of tools; we show how the quality model can be used to refine and clarify the requirements in order to guarantee a highly reliable selection result; and we use it to evaluate two particular workflow solutions available in the market (kept anonymous in the paper). We develop our proposal using a particular selection experience we have recently been involved in, namely the procurement of a document management subsystem to be integrated in an academic data management information system for our university.

1. Introduction

The goal of workflow management systems is to provide automated support to the enactment and coordination of business process in an organizational context [1]. This type of systems is being used more and more not only in staff-intensive environments but also in all those contexts in which clearly defined circuits for types of data exist.

This increasing use of workflow technologies has mainly two consequences. On the one hand, it is necessary to improve the state of the art of the workflow solutions available in the market with new functionalities or improving their quality of service (e.g., [2, 3]). On the other hand, we need to bridge the gap among workflow producers and consumers to facilitate the adoption of the adequate solutions with respect the requirements of organizations. This paper focuses on this second aspect.

There are several proposals that identify relevant criteria for driving the selection of a workflow tool, issued both by academic parties [1, 4, 5] and professional firms (e.g., [6, 7]). These proposals are enough for those cases in which the requirements

over the tool are clear and the tool is selected without considering its connections with other subsystems. However, the situation is often different. The usual case is that organizations establish some departing requirements that are very diverse in nature, including not only workflow-related services but also other type of functionalities such as customer relationship management, document management, and others. Therefore, it is necessary to identify the types of components that form the composite information system that is being procured, and refine the departing requirements into more detailed ones, assigning them to the components.

In this paper we present MULTSEC (**MULT**iple **SE**le**C**tion), a method that can be applied in the selection of multiple components, and we explain its application in a concrete real case, the procurement of a subsystem for managing documents in an academic setting. MULTSEC consists of four activities (see fig. 1): determination of the goals of the system-to-be; identification of the types of components of the system and assignment of the goals to these types; development of an informational model for describing the selection criteria (i.e., the quality factors) for the types of components; and stepwise refinement of the departing goals into detailed requirements expressed in terms of the selection criteria. The last two activities are carried out tightly intertwined.

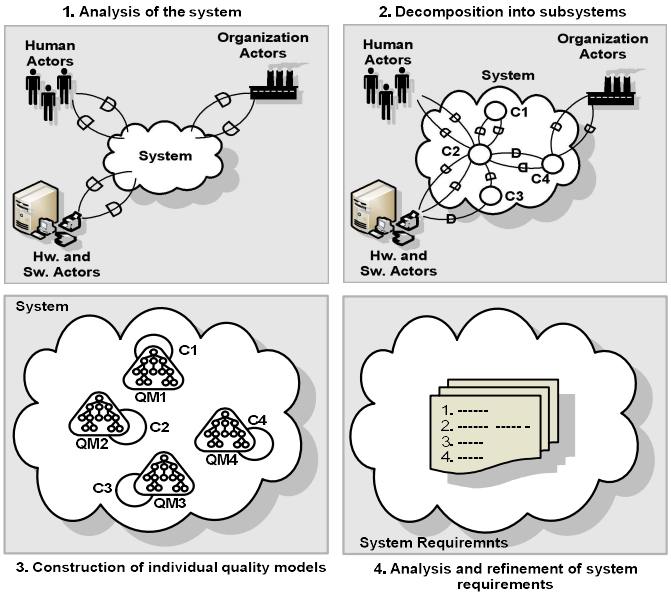

Figure 1: Representation of the MULTSEC method.

One of the key points of our MULTSEC is the convenience to have a uniform informational model for representing the selection criteria, in order to allow the simultaneous analysis of the components. We use the ISO/IEC 9126-1 quality standard [8] as such a framework. It proposes to structure the selection criteria as a quality model, defined as a hierarchy of quality factors, and fixes the higher levels of

this hierarchy, whose factors must be subsequently refined. The quality models of all the types of components of the systems share these higher levels.

The rest of the paper is structured as follows. In section 2 we present the case study that drives the paper and determine a set of initial goals of the system, used in section 3 to identify the types of components of the system. Section 4 analyses the construction of quality models for the types of components, and in section 5 we develop in some depth the construction of an ISO/IEC 9126-1-compliant quality model for workflow tools. In section 6 we show how the quality factors can be used to refine and analyze the requirements. In section 7 we evaluate two real workflow tools using the quality model. Finally, section 8 provides the conclusions of the work.

2. PRISMA: a System for Academic Data Management

PRISMA is an ongoing software project carried out in the Universitat Politècnica de Catalunya (UPC) at Barcelona, Spain. The goal of the project is to provide the UPC with an integrated management of academic data such as students, courses, departments, etc., instead of the departing situation in which every school owns its particular management system. The PRISMA system has already been successfully used in several designated pilot schools and it is supposed to be fully delivered during 2004.

At the heart of such a system, we find the need of storing, manipulating, rendering and eventually removing (when obsolete or useless) a great deal of documents of several types, such as student application forms, school curricula, academic diplomas, official advertisements and so on. The authors of this paper were been in charge of procuring a subsystem to be integrated into the PRISMA system in charge of these services. We applied the MULTSEC method with this aim.

As first step of MULTSEC, the PRISMA team determined a set of initial high-level goals addressing this part of the system functionality. Some relevant goals appear at table 1. We can distinguish among goals that are functional in nature, e.g. goals 1 and 2, and goals that are non-functional, e.g. goals 4 and 8.

3. An Architecture for the System

The second step in MULTSEC aims at decomposing the system into types of components and assigning the system goals to these types. Examples of type of components are document management and workflow management. Please note that a particular component may cover more than one type of component, e.g. there are a lot of document management tools that offer workflow capabilities. In our approach, it is crucial to keep these two concepts (type of component and component) separated because the types are the ones that determine the selection criteria.

The decomposition in MULTSEC is driven mainly by two different carriers:

– Since we use goals as the starting point of system decomposition, we follow a goal-oriented framework to guide this decomposition process and represent the resulting architecture. Among different available options, we have opted by the *i** goal-oriented specification language [9] and in particular its SD models. In SD

Table 1: Some initial goals of the PRISMA system with respect to academic data management.

Number	Goal
1	The system shall capture both paper documents and electronic documents in different formats
2	The system shall define and control automatically the flow of documents
3	The system shall visualize documents in the web when required
4	The system shall be easy to configure
5	The system shall present multi-language interfaces
6	The system shall define different types of users with their own access rights
7	The system shall provide version management
8	The system shall keep data privacy
9	The system shall react quickly to external stimuli

models, intentional actors depend on others to attain their goals. Actors and dependencies, which form together an intentional network, are determined from the initial goals of the system. Actors represent types of components and dependencies reflect how a type of component needs another for attaining goals, satisfying quality of service levels, performing tasks or consuming resources.

– To facilitate reuse and make the decomposition process more agile, types of components can be arranged in a taxonomy [10]. The taxonomy can be build from categorizations proposed by professional consultant companies, third-party organizations and knowledge accumulated from previous experiences. In [11] we have presented the rationale for building a taxonomy for business applications which includes some of the types of components that appear in our case study, remarkably the one of workflow tools. Fig. 2 shows an excerpt of this taxonomy. In order to make the taxonomy usable in our MULTSEC, the goals for each type of component should be clearly stated. For instance, the goal of the *Workflow* type may be formulated as "Facilitate or even automate business processes".

Fig. 3 shows an *i** SD model embracing the most representative type of components of the system and some of the relationships among them. The precise meaning of the 3 types of dependencies shown in the figure is out of the scope of the paper, see [9] for details. Table 2 reallocate the system goals into the types of components. We can check that functional-like goals are covered by a few, often just one, system components; on the contrary, non-functional-like goals are cross-cutting and therefore affect the majority, often all, of the system components.

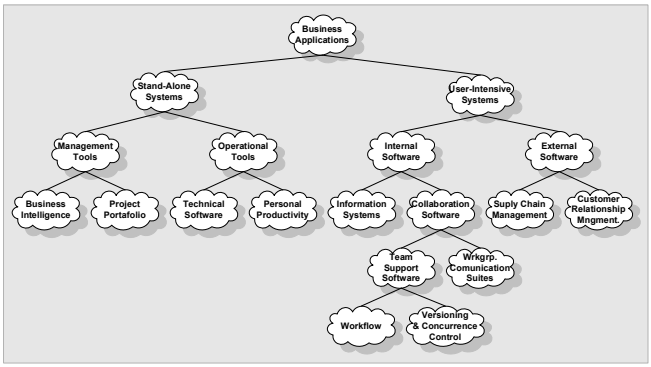

Figure 2: An excerpt of a taxonomy for business applications.

Table 2: Relationships of the system goals and the system architectural components.

Number	Components			
	DM	W	WCM	DI
1	x			x
2		x		
3			x	
4	x	x	x	x
5	x	x	x	x
6	x	x	x	
7	x			
8	x	x	x	x
9	x	x	x	x

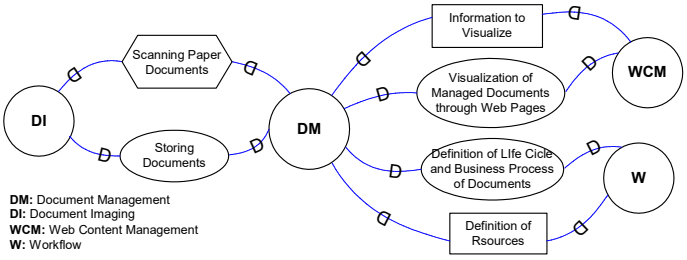

Figure 3: Some types of components of the PRISMA document management subsystem and their relationships represented with an *i** SD model.

4. Building Individual Quality Models for the System Components

In the third and fourth steps of MULTSEC the types of components are analysed in order to determine which are the relevant quality features that need to be assessed during system procurement. The starting point of the analysis of a type of component are the system goals that have been assigned to this component. These goals are to be decomposed using a quality framework for obtaining both more detailed system requirements, and measurement criteria for them in the form of quality attributes. In our method, we have chosen the ISO/IEC 9126-1 quality standard [8] as quality framework. It is one of the most, if not the most, widespread software quality standards available in the software engineering community. The fact of having the same quality framework for all the types of components of the system makes the resulting requirements more structured and easy to analyse, as shown in section 6.

An ISO/IEC-9126-1-compliant quality model is defined by means of general characteristics of software, which are further refined into subcharacteristics, which in turn are decomposed into attributes. Attributes collect the properties that software components belonging to the domain under analysis exhibit. Intermediate hierarchies of subcharacteristics and attributes may appear making thus the model highly structured.

The ISO/IEC 9126-1 standard fixes 6 top level characteristics: functionality, reliability, usability, efficiency, maintainability and portability. It also fixes their further refinement into 27 subcharacteristics but does not elaborate the quality model below this level, making thus the model flexible. The appendix summarizes this quality model. The model is to be completed based in the exploration of the software domain under analysis; because of this, we may say that the standard is very versatile and may be tailored to domains of different nature, such as the one of workflow tools.

5. A Quality Model for Workflow Management Systems

In this section we give a more detailed presentation of the quality model that we built for the workflow domain in the context of the PRISMA project. We followed a method defined elsewhere [12] which applies several steps:

1. Determining first-level quality subcharacteristics. In this step we chose subcharacteristics applicable to the workflow domain among the 27 that the ISO/IEC 9126-1 standard provides.
2. Defining a hierarchy of subcharacteristics. Subcharacteristics are further decomposed into new ones yielding to a hierarchy. As a result we obtained subcharacteristics like *Suitability* which were decomposed into subcharacteristics specific to the workflow domain and others like *Security* whose decomposition is fairly common in different domains. Some subcharacteristics appear in more than one decomposition, like *Interface parameterization*. Table 3 shows some subcharacteristics of *Functionality*, *Usability* and *Efficiency* characteristics.
3. Decomposing subcharacteristics into attributes. We decomposed the abstract subcharacteristics into quality attributes. Some attributes were directly measurable but others required further decomposition. Table 3 shows some quality attributes for some of the subcharacteristics therein.
4. Assigning metrics for the attributes. Metrics either quantitative or qualitative are defined. Typical issues like scale of the metric, measurement protocol and others [13] must be taken into account.

6. Analysis of System Requirements using the Quality Models

The resulting quality model can be used not only to describe quality requirements in a structured manner, but also as a framework to support requirements elicitation. On the one hand quality model construction can be driven towards the identification of the quality attributes which make operational the abstract goals of the system, in such way that quality requirements can be expressed as constraints over them. On the other hand attributes in the quality model can be used to discover new system requirements which were not originally considered. When making abstract goals of table 1 operational in terms of quality attributes several situations arose:

– Some goals such as goal 5 were directly mapped into a single attribute in the model, the *Interface languages supported*. This attribute belongs to the *Interface Language Issues* subcharacteristic. Once located, we realized that this subcharacteristic has other attributes that could also contribute in a positive way to

Table 3: Quality model for the *Workflow* type of component: an excerpt

Characteristic		Subcharacteristics		Attributes
		Level 1	Level 2	
Characteristic	Functionality	Suitability	Process Definition	
			Task Management	
			Organization and Notifications	Notification rules administration
				User notification types
				User notification mechanisms
				E-mail notification format
				Organizational model administration
				Substitution rules
				Support rules
				Directories importation
			Monitoring Processes and Activities	
			Reports	
		Security	Data Protection	Security provider
				Security transfer protocols
				Certification services
				Encryption algorithms
				Electronic signatures
			ACL Management	ACL support
				Access rights
				Group restrictions
				User restrictions
	Usability	Understand-ability	Interface Parameterization	(see decomposition in *Operability* below)
			Interface Languages Issues	Interface languages supported
				Modification of text labels of the application
				Vendor support to provide the interface in a particular language
		Learnability	Documentation and Support	
		Operability	Interface Parameterization	Interface standards
				Interface parameterization offered
			Accounts Administration	Individual users and groups
				Private and public accounts
				Individual and shared directories
				LDAP
				Single sign-on
				Connection with other directories
	Efficiency	Time Behavior	Response Time	Estimated work time for each activity
				Average time to complete the tasks from a workflow process instance

the attainment of the goal, namely *Modification of text labels of the application* and *Vendor support to provide the interface in a particular language*. Thus we proposed additional requirements of the system based on these attributes.

- In the case of goals 2 and 4 the situation was more complex, they implied a mixture of functionalities, supported by the selection of several attributes of the model. In the case of goal 4, it was made operational in terms of attributes of the *Operability* subcharacteristic, e.g. *Accounts Administration* and *Interface*

Parameterization among others. In the case of goal 2 attributes related to some *Suitability* subcharacteristics such as *Process Definition, Task Management* an *Organization and Notifications,* were required. The attributes of these subcharacteristics define the core functionality of the workflow component, thus they are hardly reusable in other domains. However, they became the main source of requirements for the workflow component. Needless to say that this is due to the fact that individual attributes of the quality model help to identify and/or refine requirements related to them.

- The case of goal 6 was similar to goals 2 and 4, it was also made operational in terms of several attributes of the model, namely those belonging to the *Accounts Administration* and *ACL Management* subcharacteristics. However when considered the system as a whole we found two different situations. On the one hand we considered highly desirable that all the components of the system use the same *Directory Service*, so we identified a new actor in the system providing this functionality with its own quality model; as a consequence, quality attributes concerning users were defined once instead of several times. This case aligns with the iterative nature of the MULTSEC method in which additional components of the system (in this case a directory service) can be identified at any stage of the process. On the contrary, ACLs are not common to every application, e.g. document management tools will include permissions to add, edit, delete or publicise documents, whilst workflow systems may include the abort process, reassume, reassign or expedite permissions, among others.
- Several security-related quality attributes, e.g. *secure transfer protocols*, *certification services*, *encryption algorithms* and *electronic signatures*, were used to operationalize goal 8. As in the previous case it was considered desirable that all the components in the system share the same security mechanisms.

In general it can be said that cross-cutting goals let to the identification of system level requirements. This is one of the reasons why quality models of the different components of the system are build upon the same standard and with the same method. This easies the matching of quality attributes of the different models and the statement of system-level quality requirements common to all of them.

As a further example of the utility of this characteristic consider a requirement to measure the time required by the system to incorporate a document in the system, several throughput and response times have to be considered, including: the capturing rate of document imaging tool, the response time of the document management tool and its document processing rate, the time required by the content management tool to publish documents, and the response time overhead of the workflow process managing the different tasks. Although all of them belong to different quality models, they are categorised under the same subcharacteristics. Thus by navigating the same branch of the hierarchy in all the models at the same time they can be reconciled.

A final matter which is worth to remark is that some attributes identified in individual quality models may also lead to the identification of system-level requirements. This is the case of the *Single sign-on* quality attribute which was first identified when constructing the workflow quality model. Other components also require logging into the system (usually by supplying a username and a password), but it can be desirable to avoid this redundancy in order for the system to be perceived as a single unit.

7. Evaluation of Workflow Tools

Once the quality model for workflow and the rest of system types of components was completed, we evaluated several tools. As mentioned in section 3, some tools covered more than one type of component, in which case we evaluated altogether the criteria of the involved types. In the case of workflow tools, after a first screening process usual in COTS selection processes [14] we evaluated in detail two particular tools. Both are mainly workflow tools, aiming at simplifying the management of complex processes by providing an intuitive visual design environment and the ability to track workflow process events during process enactment. We observed that (see table 4):

- Some attributes had the same value in both tools. For instance, *User notification mechanisms*: both offer the possibility of receiving notifications via email or in the lists of things to do in the activities page, as well as a simple advice in the screen.
- In addition to those attributes that are evaluating the same in both tools, we sometimes found that one tool has more values in a quality attribute than the other but this difference was not important (e.g., *Interface languages offered*). In this cases the attributes were not impacting in the selection.
- In other quality attributes we detected that the can take more values than the ones that appear considering just the requirements. This is the case of *Security transfer protocols*: initially we just required SHTTP, but then we observed that both were also offering SSL and then we added this as security transfer protocol related requirement.
- The real relevant attributes were those related to the selection goals with significant differences. This is the case of *Support rules*: although both tools support the organization rules *by role* and *by user,* Tool 1 adds the *by department* and *by group*.

8. Conclusions

Lots of work have been done in the context of improving workflow technologies with new facilities. Also, some approaches exist for the assessment of the quality of service of workflow tools considered isolated [1, 4, 5, 6, 7]. However, to the best of our knowledge, there is a lack of studies for assessing the quality of service of workflow tools when considering the procurement of a complex system in which workflow facilities are just a part of the entire functionalities, which turns out to be a very usual situation. The goal of this paper has been to provide a solution to this problem, presenting the MULTSEC method to build quality models compatible for all the types of components of the architecture.

Table 4: Matching requirements and tools information with the quality atributes

Attribute	Requirements	Tool 1	Tool 2
User notification mechanisms	As much as possible	e-mail, task page, message (screen)	e-mail, task page, message (screen)
Interface languages offered	Catalan, Spanish and English	English, Catalan, Spanish and other 13	English, Catalan, Spanish and other 27
Security transfer protocols	SHTTP	SHTTP, SSL	SHTTP, SSL
Support rules	As much as possible	By role, by group, by department, by user	By role, by user

We think that the most salient features of our approach are:

- We have considered workflow tools selection in the most usual context, i.e. as a part of a more comprehensive selection process. We have supported this by using the same informational quality model for all the types of components. Although it may be argued that for some types of components well-defined reference frameworks exist (e.g., the WfMC framework for workflow), we do believe that uniformity is more important (in some cases). Also, it is worth to remark that a way to overcome this problem is to provide a mapping from the domain-specific reference frameworks to the common informational quality model.
- This informational quality model has been chosen to be one of the most, if not the most, widespread quality models in the software engineering field, the ISO/IEC 9126-1 standard. Using a widespread standard facilitates knowledge reuse, transfer of knowledge among different types of components (i.e., quality subcharacteristics and attributes that behave the same), and dissemination of results.
- The informational quality model is tightly bound to the requirements elicitation, analysis and operationalization activities.
- We use in the approach some artefacts such as *i** diagrams and taxonomies that provide a degree of formalism that makes our process more reliable. Also these elements support reusability (types of components appear in different projects; the quality models are inherited downwards the taxonomy) and understandability (relationships are made explicit).

References

1. Worflow Management Coalition. "The Workflow Reference Model". Document Number TC00-1003, 1995.
2. H. Schuschel, M. Weske. "Integrated Workflow Planning and Coordination". In *Proceedings of 14th International Conference on Database and Expert Systems Application* (DEXA'03), LNCS 2736, 2003.
3. M. Shen, D.-R. Liu. "Coordinating Interorganizational Workflows Based on Process-Views". In *Proceedings of 13rd International Conference on Database and Expert Systems Application* (DEXA'01), LNCS 2113, 2001.
4. C. Patel, K. Supekar, Y. Lee. "A QoS Oriented Framework for Adaptative Management of Web Service Based Workflows". In *Proceedings of 14th International Conference on Database and Expert Systems Application* (DEXA'03), LNCS 2736, 2003.
5. J. Cardoso, A. Sheth, J. Miller. "Workflow Quality of Service". In *Proceedings IFIP TC5/WG5.12 International Conference on Enterprise Integration and Modeling Techniques* (ICEIMT'02), 2002.
6. http://www.document-management-software-system.net/rfp_template_db-gen.html. Last accessed March 2004.
7. http://www.wngs.com/wgs_e/wfsp4.htm. Last accessed March 2004.
8. *ISO/IEC Standard 9126-1 Software Engineering – Product Quality – Part 1: Quality Model, 2001*.
9. E. Yu. "Towards Modeling and Reasoning Support for Early-Phase Requirements Engineering". In *Proceedings 3rd IEEE International Symposium on Requirements Engineering* (ISRE), 1997.
10. R. Glass and I. Vessey, "Contemporary Application-Domain Taxonomies". *IEEE Software*, 12 (4), July 1995.

11. J.P. Carvallo, Xavier Franch, Carme Quer, Marco Torchiano. “Characterization of a Taxonomy for Business Applications and the Relationships among them”. In *Proceedings of the 3rd International Conference on COTS-Based Software System* (ICCBSS), LNCS, 2004.
12. X. Franch, J.P. Carvallo. “Using Quality Models in Software Package Selection”. *IEEE Software*, 20(1), January/February 2003, pp. 34-41.
13. N.E. Fenton, S.L. Pfleeger. *Software Metrics: A Rigorous and Practical Approach*. PWS, 1998.
14. N. Maiden, C. Ncube. "Acquiring Requirements for COTS Selection". *IEEE Software*, 15(2), 1998.

Appendix. The Quality Model presented in the ISO/IEC 9126-1 Quality Standard

Note: this appendix is included just for refereeing purposes; it should not be considered part of the paper

Functionality	
suitability	presence and appropriateness of a set of functions for specified tasks
accuracy	provision of right or agreed results or effects
interoperability	capability of the software product to interact with specified systems
security	prevention to (accidental or deliberate) unauthorized access to data
functionality compliance	adherence to application of functionality related standards or conventions
Reliability	
maturity	capacity to avoid failure as a result of faults in the software
fault tolerance	ability to maintain a specified level of performance in case of faults
recoverability	capability of reestablish level of performance after faults
reliability compliance	adherence to application of reliability related standards or conventions
Usability	
understandability	effort for recognizing the logical concept and its applicability
learnability	effort for learning software application
operability	effort for operation and operation control
attractiveness	capability of the product to be attractive to the user
usability compliance	adherence to application of usability related standards or conventions
Efficiency	
time behavior	response and processing times; throughput rates
resource utilization	amount of resources used and the duration of such use
efficiency compliance	adherence to application of efficiency related standards or conventions
Maintainability	
analysability	identification of deficiencies, failure causes, parts to be modified, etc.
changeability	capability to enable a specified modification to be implemented
stability	capability to avoid unexpected effects from modifications
testability	capability to enable for validating the modified software
maintainability compliance	adherence to application of maintainability related standards or conventions
Portability	
adaptability	opportunity for adaptation to different environments
installability	effort needed to install the software in a specified environment
co-existence	capability to co-exist with other independent software in a common environment sharing common resources
replaceability	opportunity and effort of using software in the place of other software
portability compliance	adherence to application of portability related standards or conventions