SOAQM: Quality Model for SOA Applications based on ISO 25010

Joyce M. S. França [1,2] and Michel S. Soares[3]
[1]*Federal University of Uberlândia, Faculty of Computing, Uberlândia, Brazil*
[2]*Federal Institute of Education, Science and Technology of Norte de Minas Gerais, Januária, Brazil*
[3]*Federal University of Sergipe, Department of Computing, São Cristóvão, Brazil*

Keywords: Software Quality, Service Oriented Architecture, ISO 25010, Quality Model.

Abstract: Service-Oriented Architecture (SOA) has been widely adopted to develop distributed applications with the promise of legacy systems integration and better agility to build applications by reusing services. Considering the important role of SOA in organizations, quality should be treated as a key issue. By observing the works proposed in the literature, it is possible to notice that there is a need for development of a specific quality model for SOA based on the latest ISO 25010. One of the proposals of this paper is to analyze which important contributions were aggregated into the new ISO 25010 regarding SOA applications when compared with ISO 9126. This paper provides the definition of a specific quality model for SOA based on quality attributes defined by ISO 25010. As a result, most quality attributes proposed by ISO 25010 may be applicable to SOA at some degree level. However, some of these quality attributes should be adapted when applied to SOA projects.

1 INTRODUCTION

Software systems are becoming increasingly complex over time and, thus, quality assurance is becoming increasingly important as well (Boehm, 2006) (Huang et al., 2012). To ensure adequate software quality, relevant quality characteristics must be specified, taking into account the intended use of a software product. In order to make a proper evaluation of software, relevant quality characteristics of a software product have been proposed in many quality models, including ISO standards.

Many software systems were developed in past years in an isolated manner, with little concerns about their integration (Erl, 2007), which leads to increasing complexity. Service-Oriented Computing is a paradigm that utilizes services as fundamental elements for developing applications/solutions (Papazoglou, 2003). A service is a capability of the business organization that is implemented and made available on the Internet so that other applications can access it. Service-Oriented Architectures (SOA) emerge as an attempt to integrate legacy systems by using Web Services. Within SOA, developers can combine and integrate internal legacy software assets with further components, with the purpose of creating new applications.

Several authors warn that activities related to software quality are fundamental to software product success (Sanders and Curran, 1994) (Sommerville, 2010). Evaluation of software quality is an extremely important activity in the software development process. Factors that affect quality, and can be directly measured, are called internal quality attributes (for example, lines of code). Factors that can only be indirectly measured are called external quality attributes such as maintainability.

ISO 9126 (ISO/IEC, 1991) has inspired several quality models. In 2011, ISO 9126 was replaced by ISO 25010 (ISO/IEC, 2011). There are many specific quality models for SOA already proposed in the literature. A systematic review proposed by Oriol et al. (Oriol et al., 2014) presented 47 quality models specific to Web services. However, most of these proposals did not take into consideration ISO standards. Only 6 out of 47 models were based on an ISO standard. In addition, none of these 47 models is based on ISO 25010. In a systematic mapping performed by the authors, Quality of SOA applications has been hardly mentioned.

One of the proposals of this paper is to analyze which important contributions were aggregated into the new ISO 25010 regarding SOA applications. In addition, this analysis aims to determine if limitations perceived in ISO 9126 were solved in the most recent ISO 25010. Another proposal of this paper is to inves-

M. S. França J. and S. Soares M..
SOAQM: Quality Model for SOA Applications based on ISO 25010.
DOI: 10.5220/0005369100600070
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 60-70
ISBN: 978-989-758-097-0

tigate the applicability of ISO 25010 to SOA applications. The aim is to analyze all the quality attributes (characteristics and sub-characteristics) proposed by ISO 25010 and determine which of them are directly applicable to define quality in SOA. In this direction, the research question is defined as follows:

RQ1 - What quality attributes proposed by ISO 25010 are relevant to SOA applications?

The answer to this question in this paper is the definition of a quality model specific for SOA based on quality attributes defined by ISO 25010.

2 BRIEF INTRODUCTION TO ISO 25010

Product quality model defined in ISO 25010 comprises eight quality characteristics: Functional Suitability, Reliability, Performance Efficiency, Usability, Security, Compatibility, Maintainability and Portability, and 31 sub-characteristics as depicted in Figure 1. Compared to ISO 9126, ISO 25010 is more comprehensive and complete. ISO 9126 (ISO/IEC, 1991) provides 6 characteristics and 27 sub-characteristics, while ISO 25010 provides 8 characteristics and 31 sub-characteristics. According to (Botella et al., 2004), ISO 9126 has some limitations due to its generic nature. Some concepts presented in ISO 9126 need to be refined before they can be actually applied in a real project. In addition, elements of software metrics were not clear when defining the standard.

New characteristics were inserted in ISO 25010 such as security and compatibility. Both characteristics were not defined in ISO 9126. In addition, the hierarchy of characteristics and sub-characteristics was reorganized with the purpose of improving understanding of related concepts. This effort of ISO 25010 to reorganize and create new features and improve the understanding of definitions is an attempt to address the limitations of ISO 9126 with respect to the abstract nature, incompleteness and lack of clarity as other authors have warned (Al-Kilidar et al., 2005).

Each ISO 25010 characteristic is composed of a set of related sub-characteristics, as depicted in Figure 1. A brief summary of the definition of each characteristic is presented in this section as follows.

Functional Suitability represents the degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions. This characteristic is composed of sub-characteristics Functional Completeness, Functional Correctness, and Functional Appropriateness.

Performance efficiency represents the performance relative to the amount of resources used under stated conditions. This characteristic is composed of sub-characteristics Time Behaviour, Resource Utilization, and Capacity.

Compatibility is the degree to which a product, system or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment. This characteristic is composed of sub-characteristics Co-existence and Interoperability.

Usability is the degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use. This characteristic is composed of sub-characteristics Appropriateness Recognizability, Learnability, Operability, User Error Protection, User Interface Aesthetics, and Accessibility.

Reliability is the degree to which a system, product or component performs specified functions under specified conditions for a specified period of time. This characteristic is composed of sub-characteristics Maturity, Availability, Fault Tolerance, and Recoverability.

Security is the degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization. This characteristic is composed of sub-characteristics Confidentiality, Integrity, Non-repudiation, Accountability, and Authenticity.

Maintainability represents the degree of effectiveness and efficiency with which a product or system can be modified to improve it, correct it or adapt it to changes in environment, and in requirements. This characteristic is composed of sub-characteristics Modularity, Reusability, Analyzability, Modifiability, and Testability.

Portability is the degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another. This characteristic is composed of sub-characteristics Adaptability, Installability, and Replaceability.

3 SOAQM MODEL

ISO 25010 quality characteristics need to be studied and adapted in order to be applied to SOA applications. All characteristics and sub-characteristics proposed by ISO 25010 were analyzed regarding the real applicability to SOA. Accordingly, we establish for each characteristics the degree of applicability to SOA. Then, the next step of this analysis is to define

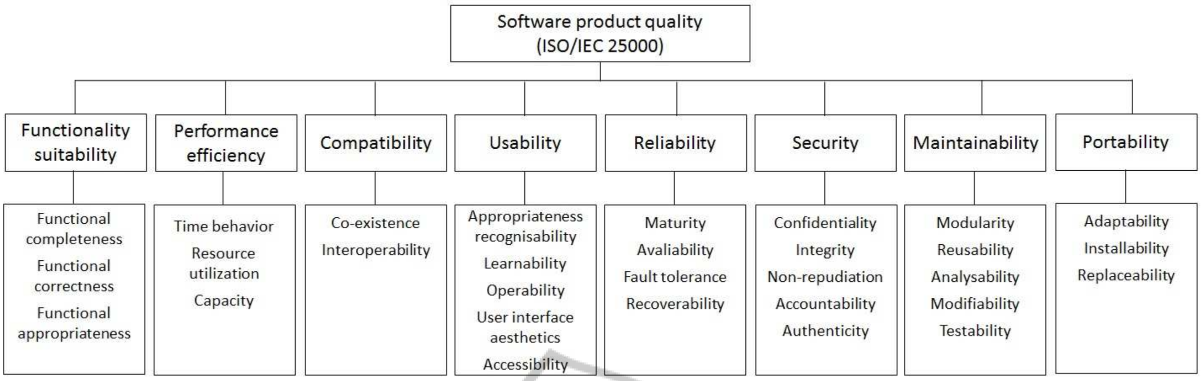

Figure 1: Quality model for external and internal quality by ISO 25010.

how these characteristics can be adapted to the SOA context.

We propose a degree of importance that defines which characteristics are more relevant during the SOA application development process. Seven volunteers answered a questionnaire to define the degree of importance. Two of them work in academia as researchers and also in industry projects, and the other five work only in industry, as developers, software architects or project managers. Each question represents a quality sub-characteristic and each volunteer answered how important this sub-characteristic is for SOA. The answer varies from 1 to 5 following the Likert Scale (1-Not Important, 2-Less Important, 3-Neutral, 4-Important, 5-Very Important). The results to this survey are described in Table 1.

Table 2 presents the results of the analysis of ISO 25010 characteristics and their applicability to SOA projects. First two columns of Table 2 shows all characteristics and sub-characteristics proposed by ISO 25010. Third column presents the degree of importance of each sub-characteristic with relation to the SOA context. Highlighted sub-characteristics in green are important or very important quality attributes for SOA applications. Sub-characteristics highlighted in yellow are not so important, but they are relevant for SOA and can not be disregarded. On the other hand, there are quality sub-characteristics considered less important or even irrelevant to SOA applications and therefore were highlighted in red. Finally, the last column presents the likely reasons that justifies the results.

The remainder of this section is divided into eight subsections. Each of the eight subsections addresses one ISO quality characteristic and also discusses how one can suit them in the context of SOA. These subsections are also important to present the reasoning behind each definition described in Table 2.

3.1 Functional Suitability

Sub-characteristic Functional Completeness means the degree to which the set of functions covers all the specified tasks and user objectives. Observing from the SOA perspective, services must cover all the specified tasks and user objectives for which they were designed. The first activity during the development of SOA applications include defining requirements to develop services. This phase is commonly known as service-oriented analysis. Service-oriented analysis (Erl, 2005) is a process of determining requirements, scope of service and which services will be developed. These important definitions are incorporated into a document known as functional document, which is used to software validation.

Sub-characteristic Functional Correctness means the degree to which a product or system provides correct results with the needed degree of precision. This sub-characteristic can be applied in SOA by the fact that services should provide the correct response with the needed degree of precision. A service requests some information and a service provider is responsible to send the correct response. Service-oriented analysis (Erl, 2005), as mentioned in previous paragraph, is also very important to Functional correctness due to the relationship between what was requested by the customer and what the service offers as response.

Sub-characteristic Functional Appropriateness is the degree to which the functions facilitate the accomplishment of specified tasks and objectives. In the SOA context, services are designed to facilitate accomplishment of specified tasks, more precisely, the execution of a business process. Services perform functions that can be simple requests for activities or complex business processes. Therefore, services can be of simple nature or composite (Papazoglou, 2003). Composite services are construct of the orchestration

Table 1: Volunteers Opinion about importance of ISO 25010 for SOA - Likert Scale.

		Volunteers Opinion							Statistics	
Characteristic	**Sub-characteristics**	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**	**Average**	**Stand. Dev.**
Functional Suitability	Functional completeness	5	4	5	5	3	5	2	4.1	1.1
	Functional correctness	4	5	5	4	4	5	5	4.6	0.5
	Functional appropriateness	5	5	5	4	4	5	2	4.3	1.0
Performance efficiency	Time behaviour	5	5	4	5	4	5	2	4.3	1.0
	Resource utilization	3	5	4	4	4	5	4	4.1	0.6
	Capacity	1	5	3	4	3	5	1	3.1	1.6
Compatibility	Co-existence	3	5	3	5	3	5	4	4.0	0.9
	Interoperability	5	5	3	5	4	5	5	4.6	0.7
Usability	Appropriateness recognizability	5	2	4	4	3	4	1	3.3	1.3
	Learnability	5	2	4	4	4	4	3	3.7	0.9
	Operability	5	5	4	2	4	5	3	4.0	1.1
	User error protection	5	5	4	4	3	4	1	3.7	1.3
	User interface aesthetics	4	3	4	2	3	1	1	2.6	1.2
	Accessibility	4	2	4	4	3	1	1	2.7	1.3
Reliability	Maturity	4	5	4	4	4	5	4	4.3	0.5
	Availability	5	5	4	5	5	5	5	4.9	0.3
	Fault tolerance	5	5	4	5	4	5	4	4.6	0.5
	Recoverability	5	5	4	5	4	5	5	4.7	0.5
Security	Confidentiality	5	5	4	4	4	5	5	4.6	0.5
	Integrity	5	5	4	4	4	5	5	4.6	0.5
	Non-repudiation	4	3	4	4	4	4	4	3.9	0.3
	Accountability	5	5	4	4	4	4	4	4.3	0.5
	Authenticity	5	5	4	4	4	5	5	4.6	0.5
Maintainability	Modularity	5	5	4	4	4	5	5	4.6	0.5
	Reusability	5	5	4	4	4	5	5	4.6	0.5
	Analyzability	5	4	4	2	5	5	2	3.9	1.2
	Modifiability	5	5	4	4	4	5	5	4.6	0.5
	Testability	5	5	4	4	4	5	5	4.6	0.5
Portability	Adaptability	5	2	3	2	4	5	5	3.7	1.3
	Installability	4	1	3	2	4	5	5	3.4	1.4
	Replaceability	4	2	3	4	3	5	5	3.7	1.0

process, which allows services to be composed of other services to automating business processes.

3.2 Performance Efficiency

Performance efficiency characteristic can be applied in SOA applications to evaluate performance of a service.

Sub-characteristic Time-behaviour analyzes the response and processing times and throughput rates of a system, when performing its functions. Contextualizing for SOA, we can say response time is related to time spent by services to process a request and return a response. Generally, a service requests an information for a provider application and then this service provides responses for a consumer application. Thus, it is possible to measure response and processing times and throughput rates of this transaction.

Sub-characteristic Resource Utilization addresses the amounts and types of resources used by a product or system, when performing its functions. Generally, software should make a best use of resources such as processor capacity, memory usage, disk capacity and network bandwidth. With regard to SOA, service based applications interact with other systems exchanging messages over the network. Therefore, we can consider that services use resources such as server to access information of other applications.

Sub-characteristic Capacity means the degree to which the maximum limits of a product or system parameter meet requirements. This quality sub-characteristic was vaguely defined by ISO. It is difficult to set the real meaning of this concept. We suppose that capacity can be the application ability to support several accesses at the same time without performance variation. In this sense, capacity is related with availability of a service even with many access at the same time.

Performance efficiency is an important quality characteristic which must be constantly observed in SOA applications. According to O'Brien (O'Brien et al., 2007), performance is negatively affected in SOA. Therefore, the architecture should be carefully designed and evaluated prior to implementation to

Table 2: ISO 25010 characteristics mapped to SOA quality characteristics.

Characteristic	Sub-characteristics	Importance for SOA	SOA Perspective
Functional Suitability	Functional completeness	4.1	Services must cover all the specified tasks and user objectives which were designed
	Functional correctness	4.6	Services should provide the correct results with the needed degree of precision
	Functional appropriateness	4.3	Services are designed to facilitate accomplishment of specified tasks, more precisely, the execution of a business process.
Performance efficiency	Time behaviour	4.3	Time spent by a service to process a request and return a response.
	Resource utilization	4.1	Services use resources such as servers to access information of other applications.
	Capacity	3.1	Service capacity can be defined as the ability to remain working even with large number of accesses at the same time.
Compatibility	Co-existence	4.0	Different composite services can share the use of same service operations.
	Interoperability	4.6	Services are interoperable. Services allow interaction between systems through the use of interfaces (WSDL) and communication protocols (SOAP).
Usability	Appropriateness recognizability	3.3	Users can recognize whether this service is appropriate for their needs through service description that relate information such as service functionality and data types transmitted.
	Learnability	3.7	Degree to which a service can facilitate the understanding of its operation.
	Operability	4.0	A service has a WSDL document that allows exchange messages between services.
	User error protection	3.7	The WSDL structure of a service should not allow making errors from wrong inputs.
	User interface aesthetics	2.6	Not the focus of SOA.
	Accessibility	2.7	Not the focus of SOA.
Reliability	Maturity	4.3	Whenever a service consumer requests some information, it is expected that a response is returned.
	Availability	4.9	Services must be available when they are requested.
	Fault tolerance	4.6	Services can create strategies that may be performed when a failure happens on some hardware or software.
	Recoverability	4.7	Service ability to recover data when occurs some interruption or failure.
Security	Confidentiality	4.6	Information shared by a service provider can be accessed only to an authorized service client.
	Integrity	4.6	Services must be developed to prevent unauthorized access to, or modification of private data.
	Non-repudiation	3.9	Service provider constructs strategies to prove that an information have been delivered to a service consumer
	Accountability	4.3	Service are autonomous.
	Authenticity	4.6	The identity of the external service provider should be authenticated.
Maintainability	Modularity	4.6	Service provider hosts a network accessible software module.
	Reusability	4.6	Services are reusable.
	Analyzability	3.9	Analyze change impact when services need to be modified
	Modifiability	4.6	Services are loosely coupled. This characteristic reduces the dependency between services, increasing modifiability.
	Testability	4.6	services can be tested, for instance, through automated tools for functional testing
Portability	Adaptability	3.7	Although web services run remotely on a server, it can happen a change of platform.
	Installability	3.4	Although web services run remotely on a server, it can happen a change of platform.
	Replaceability	3.7	Although web services run remotely on a server, it can happen a change of platform.

avoid performance pitfalls.

Service application performance depends on the combined performance of cooperating components and their interactions. Beyond consumers and providers applications and the need to communicate over the network, services can be composed. Composite services allows services to be composed of other services, in such a way the logic of the process is centralized by orchestration that enables extensibility (composition of new services). Thus, time behaviour depends on several factors. To deal with so many factors that can affect system performance, companies need constantly to monitor the health of their SOA applications (Papazoglou et al., 2008).

3.3 Compatibility

Sub-characteristic Co-existence is concerned with the degree to which a product can perform its required functions efficiently while sharing a common environment and resources with other products, without detrimental impact on any other product. From the SOA perspective, co-existence concept can be redefined/extended as follows. A composite service matches a set of services that perform operations and together form a specific task. As a service can be reused by several composite services, some efforts are needed to ensure what some authors have been called service conformance (Papazoglou et al., 2008). Service conformance ensures the integrity of a composite service matching its operations with those of its constituent component services, imposes semantic constraints on the component services and ensures that constraints on data exchanged by component services are satisfied. This means the co-existence concept can be extended to SOA due to different composite services that share the use of same service operations.

Sub-characteristic Interoperability refers to the degree to which two or more systems can exchange information and use the information that has been exchanged. Services are interoperable (Papazoglou et al., 2008). SOA utilizes services to construct an application that allows exchanging messages through networking protocol. This means services are interoperable and they allow interaction between systems through the use of interface (WSDL) and communication protocols (SOAP). To allow increased interoperability is the most prominent benefit of SOA (O'Brien et al., 2007).

3.4 Usability

Within the SOA context, usability is a measure of the quality of a user's experience in interacting with information or services. Usability as a concept has been rarely addressed in the context of SOA. Few studies have been published addressing this issue (Liu, 2009). One research named this issue as service oriented usability (Liu, 2009), suggesting that some procedures may increase usability of services, such as development of diagrams for service and definition of life cycle and lifetime of services.

Sub-characteristic Appropriateness recognizability refers to the degree to which users can recognize whether a system is appropriate for their needs. In this sense, it is important to have access to some description that relate the service operations. Relationship between services is based on an understanding that for the services to interact, they must be aware of each other and this awareness is achieved through the use of service descriptions (Erl, 2005). A service description establishes the name and location of the service, and data exchange requirements. SOA uses a standard format for service description known as WSDL (Web Services Description Language) (WSDL, 2014). Therefore, sub-characteristic Appropriateness recognizability can be applied in SOA because service description relates service functionality and data types transmitted, and so users can recognize whether this service is appropriate for their needs.

Sub-characteristic Learnability is related to the degree to which a system can be used by specified users to achieve specified goals of learning to use the system in an easy way. From the SOA perspective, this sub-characteristic refers to the degree to which a service can facilitate the understanding of its operation.

Sub-characteristic Operability refers to the degree to which a product or system has attributes that make it easy to operate and control. A service has a WSDL document that allows exchanging messages between services. These messages are transmitted through a standard protocol called SOAP (Simple Object Access Protocol) (SOAP, 2014). The standards defined for SOA make it easy to develop, operate and control services.

Sub-characteristic User error protection is related to the degree to which a system protects users against making errors. Within the SOA context, this quality sub-characteristics may be appropriate to SOA by the fact that a service has a description document (WSDL) and its structure should not allow different inputs from the specified ones.

Sub-characteristic User interface aesthetics refers to the degree to which a user interface enables pleasing and satisfying interaction for the user. User interface aesthetics or graphical interface is not the focus of SOA. Researches in SOA are hardly concerned with this sub-characteristic. Probably, the reason is attributed to the fact that there it is research in other areas such as human computer interaction that can be leveraged to SOA applications.

Sub-characteristic Accessibility is related to the degree to which a product or system can be used by people with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use. Accessibility is not very important to services because this issue is not the focus of SOA knowledge and research.

3.5 Reliability

Sub-characteristic Maturity is related to the degree to which a system meets needs for reliability under normal operation. Clearly, maturity can be applied in SOA because whenever a service consumer requests some information it is expected that a response is returned. Therefore, Maturity is a very important sub-characteristic that can be addressed in SOA applications to maintain quality.

Availability is a very important sub-characteristic in SOA applications. Availability consists in defining the degree to which a system, product or component is operational and accessible when required for use. This concept can be associated with SOA by the fact that a service provider must be available when a service consumer requests some information. Downtime could affect the provider's finances and reputation. According to O'Brien (O'Brien et al., 2007), external service providers usually agree to provide services under a service level agreement (SLA), which defines the contract for the provision of the service with details such as who provides the service, the guaranteed availability of the service, the escalation process, and the penalties to the provider if a service level is not met.

Sub-characteristic Fault Tolerance refers to the degree to which a system operates as intended despite the presence of hardware or software faults. From the SOA perspective, fault tolerance can be applied in SOA applications as meaning that strategies must be performed when a failure happens on some hardware or software. This ability is not SOA native and should be designed and implemented by SOA developers.

Sub-characteristic Recoverability is related to the degree to which, in the event of an interruption or a failure, a product or system can recover data directly affected and re-establish the desired state of the system. This quality sub-characteristic can be adjusted to SOA as regards the service ability to recover data when any interruption or failure occurs. Auto recovery tools have been developed to provide the functionality of recover the faulted instances and Invoke and Callback messages during business process execution.

3.6 Security

Services are designed to limit information in the service contract to what is really necessary for the service to be functionally useful to consumers (Erl, 2007). Information beyond that is published in a service contract, is considered private, and should not be made available for creating potential consumers of service.

Confidentiality is the degree to which a system ensures data are accessible only to those authorized to have access. This quality sub-characteristic must be found in SOA applications because information shared by a service provider can be accessed only to an authorized service client. Business process interactions are always controlled from the (private) perspective of one of the business parties involved in the process (Papazoglou et al., 2008).

Another security quality sub-characteristic is Integrity. Integrity refers to the degree to which a system, product or component prevents unauthorized access to, or modification of, software or data. Services must be developed to prevent unauthorized access to, or modification of private data. Integrity is related with one of the design principles of software engineering proposed in SWEBOK (Software Engineering Body of Knowledge) (IEEE Computer Society, 2004): Encapsulation/information hiding. Encapsulation/information hiding means grouping and packaging elements and internal details of an abstraction and making those details inaccessible. In this way, services must publish in a registry, or repository such as UDDI. Ensuring implementation details are inaccessible provides the guarantee that this information is not modified or corrupted.

Sub-characteristic Non-repudiation is related to the degree to which actions or events can be proven to have taken place, so that the events or actions cannot be repudiated later. In the SOA context, this concept can be the ability of a service provider to prove that information has been delivered to a service consumer.

Sub-characteristic Accountability refers to the degree to which actions of an entity can be traced uniquely to the entity. According to (Papazoglou et al., 2008), services are autonomous. That way, services have a contract that expresses a well-defined functional threshold which should not involve other services. In addition, Thomas Erl (Erl, 2007) defined Service Autonomy as a design principle. Service Autonomy refers to the ability to govern itself. When something is autonomous, it has freedom and control to make their own decisions without the need for external validation or approval. Accountability in the SOA context refers to the service state of being accountable, in other words, a service has the obligation of render an account for failure to perform as expected.

Another security quality sub-characteristic is authenticity. Authenticity refers to claim and identification of a subject or resource requests access to a certain information. A system built using a SOA approach may encompass services provided by third-party organizations. The identity of the external service provider should be authenticated.

According to Papazoglou (Papazoglou et al., 2008), security is a grand challenge within SOA applications. Services must be developed to be self-protecting, which can anticipate, detect, identify and protect against threats. Self-protecting components can detect hostile behaviors, e.g., unauthorized access and use, virus infection and proliferation, and denial-of-service attacks, as they occur and take corrective actions to make themselves less vulnerable. Self-protecting capabilities allow businesses to consistently enforce security and privacy policies. One of the strategies used to increase security is encryption. Encryption should be used to preserve confidentiality and preserve integrity. However, as mentioned in (O'Brien et al., 2007), encryption has the effect of increasing messages size.

3.7 Maintainability

Maintainability quality characteristic is very important in SOA due to its primary proposal of easy maintaining distributed applications. Before emergence of SOA, several distributed solutions were proposed, with varying degree of success and limitations. Many systems were developed with little elaboration and integration point to point was created according to the needs that were emerging. This approach produced a tangle of complex connections, lack of stability, difficult maintenance, and problems associated with extensibility and interoperability frameworks (Erl, 2005). SOA has emerged with the purpose of development of rapid, low-cost and easy composition of distributed applications.

Modularity is a quality sub-characteristic that defines the capacity of a system to be composed of components such that a change to one component has minimal impact on other components. This definition is similar to a service concept. In a typical service-based scenario employing the service foundations plane a service provider hosts a network accessible software module (an implementation of a given service) (Papazoglou et al., 2008).

Reusability is an important quality sub-characteristic which indicates that an asset can be used in more than one system, or in building other assets. This definition can be extended to SOA. Service reusability is a SOA design principle that means services are reusable (Erl, 2007). Services encapsulate the logic that is useful for more than one service request. Logic encapsulated by services is generic enough to allow many usage scenarios and can be used for different types of service consumers.

Sub-characteristic Analyzability indicates the degree of effectiveness and efficiency with which it is possible to assess the impact on a product or system of an intended change to one or more of its parts, or to diagnose a product for deficiencies or causes of failures, or to identify parts to be modified. Within SOA context, Analyzability can be an important quality attribute for services. In situations that a service need to be modified, analysis of which composite services use this service are necessary.

Modifiability is a quality sub-characteristic related to the capacity of software to be modified without introducing defects or degrading existing product quality. Modifiability increases as the independence between modules of the system increases because a change in one module can affect all modules that are dependent. By definition, services are loosely coupled (Papazoglou et al., 2007). This means there are few well-known dependencies between services. Thus, the cost of modifying the implementation of services is reduced and the overall system modifiability increases (O'Brien et al., 2007).

Testability is a quality sub-characteristic found in a system or component in which test criteria can be established and tests can be performed to determine whether those criteria have been met. Services can be tested through automated tools for functional testing. According to O'Brien (O'Brien et al., 2007), testing a system based on SOA is more complex than an isolated software. For instance, if a runtime problem occurs, it may be difficult to find the cause. It can be within the service user, the service provider, the communication infrastructure, the discovery agent (UDDI), or it can be due to the load on the platform where the service executes. Trying to replicate problems in a test environment may be difficult or even not possible within a time frame.

3.8 Portability

Portability is a quality characteristic concerned with the effectiveness and efficiency with which a system can be transferred from one hardware or usage environment to another.

Sub-characteristic Adaptability refers to the degree to which a product or system can effectively and efficiently be adapted for different or evolving hardware, software or other operational or usage environment.

Sub-characteristic Installability is related to the degree of effectiveness and efficiency with which a product or system can be successfully installed and/or uninstalled in a specified environment.

Sub-characteristic Replaceability indicates the degree to which a product can replace another specified software product for the same purpose in the same environment.

In general, some authors consider that this quality characteristic is not so important in the SOA context because web services run remotely on a server. For instance, according to (Oriol et al., 2014), Installability is usually not applicable to web services because they are executed remotely at the server side. On the other hand, a change of a server platform of an application provider can be performed and then services should normally maintain their situation.

4 COMPARISON WITH RELATED WORKS ON QUALITY MODELS FOR SOA

Quality models are elaborated to describe, evaluate and predict quality of a product (Deissenboeck et al., 2009). According to software quality standard ISO 25010, quality models are useful for specifying requirements, establish measures and assessments of quality (ISO/IEC, 2011). There are several general purpose quality models proposed for software systems. They differ on the terminology, the set of attributes that define quality, and the structure of the quality model.

Generic quality models, as the ones proposed in the literature, need to be adapted to be applied to SOA. Generic standards such as ISO 9126 can not completely conform to the web services domain (Oriol et al., 2014). For this reason, there are many specific quality models for SOA already proposed in the literature.

A systematic review proposed by Oriol et al. (Oriol et al., 2014) presented 47 quality models specific to Web services. As a result of this study, contrary to what could be expected, most of the proposals did not take into consideration ISO standards. Only 5 out of 47 models were based on ISO 9126: WSQM (WSQM, 2005), S-Cube Quality Reference Model (The European Network of Excellence in Software Services and Systems (S-Cube), 2008), GESSI (Ameller and Franch, 2008), Yin et al. (Yin et al., 2010), and Nadanam and Rajmohan (Nadanam, P. and Rajmohan, R., 2012). Therefore, none of the 47 quality models were based on the new ISO 25010.

In the next paragraphs, we propose a comparison between those 5 quality models for SOA based on ISO 9126 and the SOAQM model proposed in this paper, which is based on ISO 25010. The main objective is to present what has been proposed in the literature and detach the novelty we are proposing in this new quality model for SOA.

WSQM is a Quality Model for Web Services proposed by OASIS in 2005 (WSQM, 2005). WSQM comprises a set of quality attributes that are important for Web Services domain. WSQM model was structured into 6 categories called Web Services Quality Factors: Business Value, Service Level Measurement, Interoperability, Business Processing, Manageability and Security. Sub-factors are defined to represent quality attributes. Besides a hierarchical structure with categories and sub-factors, there is no concern to develop the same structure and use of all the attributes set by ISO 9126. In addition, WSQM does not present a mature state because many of the contained definitions lack precision (Goeb and Lochmann, 2011).

S-Cube Quality Reference Model (The European Network of Excellence in Software Services and Systems (S-Cube), 2008) consists of a set of quality attributes for service-based applications. S-Cube was structured into 10 categories which are Performance, Dependability, Security, Data-related Quality, Configuration-related Quality, Network- and Infrastructure-related Quality, Usability, Quality of Use Context, Cost and Other. Then, 78 quality attributes were defined and distributed in these categories. S-Cube was based on ISO 9126 but some definitions proposed by ISO 9126 were not mentioned. For instance, Functionality, which is an important characteristic for ISO 9126, was not considered as a category. In addition, the classification of attributes is considered by the authors as follows in ISO 9126, which is not always true. S-Cube is considered a model too complex to be used in an intuitive way (Goeb and Lochmann, 2011).

GESSI is a quality model based on ISO 9126 proposed to focus on the quality characteristics that are observable (Ameller and Franch, 2008). This study proposes a tool called SALMon to monitor services to detect Service Level Agreement violations. However, just a small set of quality attributes proposed by ISO 9126 are observable. Therefore, just three quality attributes were addressed by GESSI, which are related to sub-characteristics Availability, Time Behaviour, and Accuracy.

In (Yin et al., 2010), an ontology is proposed to support quality of services considering the following non-functional requirements: expressiveness, robustness, flexibility, scalability, performance, completeness, friendliness and interoperability. A mapping of these quality requirements and concrete requirements is proposed, with focus on only part of ISO 9126.

Nadanam and Rajmohan proposed in 2012 (Nadanam, P. and Rajmohan, R., 2012) a quality model for evaluating QoS (Quality of service) for web services. This quality model also considered cloud computing issues. As a result of this study, the quality model proposed some attributes and metrics considered relevant for cloud services. Two quality characteristics defined by ISO 9126 were considered in this model: Efficiency and Reliability. These quality characteristics are extended to cover cloud computing peculiar features. Based on the derived key features, Reusability, Adaptability, Availability, Composability, Understandability and Scalability are newly defined.

Contrary to the result of the systematic review proposed in (Oriol et al., 2014), we found a quality model for SOA that is actually based on ISO 25010. A Software Quality Model for SOA proposed by Goeb et. al (Goeb and Lochmann, 2011) was inspired in ISO 25010. However, this model analyzes only 5 quality characteristics: Consumption, Understanding, Service Reuse, Support, and Extension. Furthermore, this model was not comprehensive enough to consider the 8 quality characteristic and 31 sub-characteristics proposed by ISO 25010. In addition, two out of five considered characteristics (Support and Extension) were not defined by any version of ISO. Another issue with this quality model is that it did not follow the standardization of names established by ISO 25010 because two characteristics were renamed: Appropriateness Recognizability was renamed to Understanding, and Functional Appropriateness was called Consumption.

Furthermore, existing models are not comprehensive and do not address all quality attributes proposed by ISO 25010, as they are based on ISO 9126. Even in this case, they do not always follow the ISO 9126 standard. Although the models do follow ISO 9126, such as the S-Cube, the classification of attributes does not follow strictly ISO 9126, and important characteristics such as Functionality is not considered. In another case, WSQM, there is a hierarchical structure with categories and sub-factors, but not all attributes of the ISO 9126 structure is used. This happens in GESSI as well, in which only 3 quality attributes are addressed, and in the model proposed in (Yin et al., 2010).

By observing the works proposed in the literature, it is possible to notice that there is a need for development of a specific quality model for SOA based on the latest ISO 25010, as described in this paper.

5 CONCLUSION

This paper addresses one gap in researches about SOA. There is a lack of quality models specific for SOA based on most recent ISO 25010. In this sense, in this paper we propose a quality model for SOA applications based on quality attributes proposed in ISO 25010. ISO 25010 is a standard for software quality that presents a quality model that may be applied for any kind of software products. ISO 25010 is currently the most complete version of quality models of ISO standards due to the great amount of proposed quality attributes.

All quality attributes proposed by ISO 25010 were analyzed in this research from the SOA perspective using both experts opinion and literature review. As a result, most of quality attributes proposed by ISO 25010 may be applicable for SOA. However, these quality attributes should be adapted to SOA contexts, as discussed in Section 3. On the other hand, ISO 25010 has a generic nature and therefore some quality attributes proposed do not apply to SOA domain, as described in this paper.

One threat to validity is the small number of subjects that responded the survey. However, as they are experts in the SOA context, their opinion can be highly considered.

Generally, quality models propose methods to measure the quality attributes in applications. In this sense, a future work of this research is the definition of metrics to evaluate quality in SOA applications. In addition, currently our research group is using the SOAQM model as a guide to spend more resources on specific characteristics that were considered more important than others.

ACKNOWLEDGEMENTS

We would like to thank the Brazilian research agencies CNPq (grant 445500/2014-0) and CAPES/ FAPITEC (grant AUXPE 0517/2014) for supporting this work.

REFERENCES

Al-Kilidar, H., Cox, K., and Kitchenham, B. (2005). The Use and Usefulness of the ISO/IEC 9126 Quality

Standard. In *International Symposium on Empirical Software Engineering*.

Ameller, D. and Franch, X. (2008). Service Level Agreement Monitor (SALMon). In *Proceedings of the Seventh International Conference on Composition-Based Software Systems (ICCBSS)*, pages 224–227.

Boehm, B. (2006). A View of 20th and 21st Century Software Engineering. In *Proceedings of the 28th International Conference on Software Engineering*, ICSE '06, pages 12–29.

Botella, P., Burgus, X., Carvallo, J. P., Franch, X., Grau, G., Marco, J., and Quer, C. (2004). ISO/IEC 9126 in Practice: What Do We Need to Know. In *Proceedings of the 1st Software Measurement European Forum*.

Deissenboeck, F., Juergens, E., Lochmann, K., and Wagner, S. (2009). Software Quality Models: Purposes, Usage Scenarios and Requirements.

Erl, T. (2005). *Service-Oriented Architecture Concepts, Technology, and Design*. Prentice Hall, Upper Saddle River, NJ, USA.

Erl, T. (2007). *SOA Principles of Service Design (The Prentice Hall Service-Oriented Computing Series from Thomas Erl)*. Prentice Hall, Upper Saddle River, NJ, USA.

Goeb, A. and Lochmann, K. (2011). A Software Quality Model for SOA. In *Proceedings of the 8th International Workshop on Software Quality*, WoSQ '11, pages 18–25.

Huang, C.-Y., Leung, H., Leung, W.-H. F., and Mizuno, O. (2012). Software quality assurance methodologies and techniques. *Advances in Software Engineering*.

IEEE Computer Society (2004). *Software Engineering Body of Knowledge (SWEBOK)*. Angela Burgess, EUA.

ISO/IEC (1991). Software Engineering - Product Quality, ISO/IEC 9126. Technical report, International Organization for Standardization.

ISO/IEC (2011). ISO/IEC 25010 - Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models. Technical report.

Liu, L.-L. (2009). Design principles and measurable service oriented usability. In *IEEE International Conference on Service-Oriented Computing and Applications, (SOCA)*, pages 1–4. IEEE.

Nadanam, P. and Rajmohan, R. (2012). QoS Evaluation for Web Services in Cloud Computing. In *Proceedings of the Third International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–8.

O'Brien, L., Merson, P., and Bass, L. (2007). Quality Attributes for Service-Oriented Architectures. In *Proceedings of the International Workshop on Systems Development in SOA Environments*, SDSOA '07.

Oriol, M., Marco, J., and Franch, X. (2014). Quality Models for Web Services: A Systematic Mapping. *Information and Software Technology*, 56(10):1167–1182.

Papazoglou, M. P. (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering*, WISE '03.

Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. (2007). Service-Oriented Computing: State of the Art and Research Challenges. *Computer*, 40(11):38–45.

Papazoglou, M. P., Traverso, P., Dustdar, S., and Leymann, F. (2008). Service-Oriented Computing: a Research Roadmap. *International Journal of Cooperative Information System*, 17(2):223–255.

Sanders, J. and Curran, E. (1994). *Software Quality: a Framework for Success in Software Development and Support*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

SOAP (2014). Simple Object Access Protocol (SOAP) 1.1. W3C Note. . online.

Sommerville, I. (2010). *Software Engineering*. Addison Wesley, Essex,UK, 9 edition.

The European Network of Excellence in Software Services and Systems (S-Cube) (2008). Quality Reference Model for SBA. Gehlert, A. and Metzger, A., S-Cube Consortium.

WSDL (2014). Web Services Description Language (WSDL) 1.1. W3C Note. . online.

WSQM (2005). Quality Model for Web Services (WSQM2.0). Working draft, OASIS.

Yin, B., Yang, H., Fu, P., and Chen, X. (2010). A Semantic Web Services Discovery Algorithm Based on QoS Ontology. volume 6335 of *Lecture Notes in Computer Science*, pages 166–173. Springer.