# Towards a Conceptual Framework for Decomposing Non-functional Requirements of Business Process into Quality of Service Attributes

Camila F. Castro[1], Marcelo Fantinato[1], Ünal Aksu[2], Hajo A. Reijers[2] and Lucinéia H. Thom[3]

[1]School of Arts, Sciences and Humanities, University of São Paulo, São Paulo – SP, Brazil

[2]Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

[3]Institute of Informatics, Federal University of Rio Grande do Sul, Porto Alegre – RS, Brazil

Keywords: Non-functional Requirements, Service Level Agreements, Quality of Services, Web services.

Abstract: Non-functional Requirements (NFRs) of web services are defined by IT teams at the implementation level often as Quality of Service (QoS) attributes. Orchestrating web services to run business processes requires a rigorous definition of the NFRs of such web services. The definition of QoS attributes should consider the business process NFRs since misinterpretations of web service NFRs may affect the behavior of the web services and hence achieving the business goals. The approaches proposed so far are still heavily dependent on an IT expert's knowledge to identify the appropriate QoS attributes required to meet particular business process NFRs. Defining appropriate QoS attributes without reference to business process-level NFRs may be a costly, time-consuming task. We propose a conceptual framework for the hierarchical decomposition of NFRs from the business process level to the web service level. This framework seeks to reduce the dependence on a particular IT expert's knowledge by simplifying the dialog between the business and IT areas. The proposed framework relies on a structure of NFRs interdependence. The main reference was the ISO/IEC 25010 Product Quality Model, extended by additional software quality models and particular QoS attributes.

## 1 INTRODUCTION

To ensure the success of executing a business process through a web service orchestration, functional requirements and Non-functional Requirements (NFRs) of the web services should be considered. Web service NFRs are often defined as Quality of Service (QoS) attributes, formalized in Service Level Agreements (SLA) established between web service providers and consumers. QoS attributes defined in SLAs are propagated from specific business goals (O'Brien et al., 2005), for instance: a business goal related to agility may require QoS attributes such as adaptability, scalability and extensibility. Therefore, different web services require different QoS attributes, and what attributes are required depends on the business domain, intended use and user requirements (Abramowicz et al., 2008).

The definition of QoS attributes in SLAs should rely on business process NFRs. Business process NFRs can be formalized in Business Level Agreements (BLA), which should be defined by business or requirements analysts and capture business process-level NFRs useful later for web service provisioning (Bratanis et al., 2010; Salles et al., 2013; Barros et al., 2014; Salles et al., 2018). However, a decomposition of BLAs into SLAs depend on an Information Technology (IT) expert's knowledge to identify the proper QoS attributes required for a web service, based on implicit business process NFRs.

As an illustrative example, Figure 1 shows a fragment of a business process model for assessing loan against property applications (Dumas et al., 2013). Once received the customer application form from the *Loan Officer*, the *Financial Officer* needs to check the customer's credit history to assess the loan risk, while the *Property Appraiser* appraises the property. When both of them complete these activities, the *Loan Officer* is able to assess the customer's eligibility for the requested loan. This set of activities is susceptible to some NFRs. For example, the execution of these activities may include BLAs related to: *(i)* security, since private customer data in other institutions should be accessed; *(ii)* performance efficiency, since a rapid response must be sent to the customer so as not to lose this business opportunity; and *(iii)* compliance, as regulatory rules may need to be met. The web services implementing these activities are also suscepti-



Castro, C., Fantinato, M., Aksu, Ü., Reijers, H. and Thom, L.
Towards a Conceptual Framework for Decomposing Non-functional Requirements of Business Process into Quality of Service Attributes.
DOI: 10.5220/0007656504810492
In *Proceedings of the 21st International Conference on Enterprise Information Systems (ICEIS 2019)*, pages 481-492
ISBN: 978-989-758-372-8

ble to related NFRs since the effective execution of the business process depends directly on the effective execution of the web services.

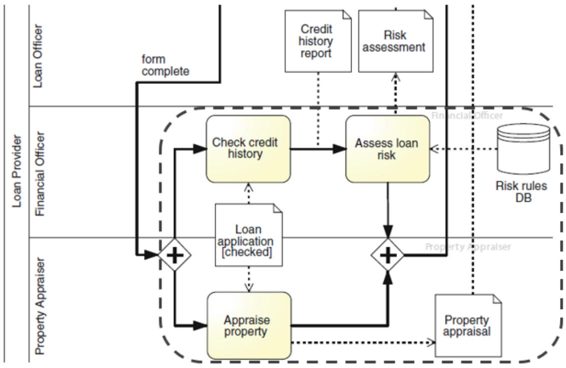


Figure 1: Fragment of a business process for assessing loan applications (Dumas et al., 2013).

Translating the business process NFRs exemplified above into appropriate NFRs to provision web services to support the execution of such a business process is a challenging job. For example, assuming the business team defines a BLA stating the execution of these three activities together should not exceed 5 minutes: what QoS properties for which web services should be defined to ensure achieving this business goal? The challenge addressed here is how IT teams can define which SLAs (and for what web services) are needed to meet a BLA defined by business teams.

In this paper, we present a conceptual framework to hierarchically decompose NFRs from the business process level to the web service level. The proposed framework relies on the interdependence of NFRs. Given a business process-level NFR, this framework describes the related NFRs that could be considered at the web service level to meet business goals. This decomposition is not automated, but an approach to help IT teams in breaking down preferences of business process NFRs into more detailed preferences related to web service QoS attributes. Motivation comes from the importance of an appropriate web service execution to meet business goals, which hence requires pertinent QoS attributes based on business needs.

To achieve this goal, we first developed a dictionary of NFRs, including both business process and web service levels. Then, we defined the interdependence among the NFRs at both levels via UML (Unified Modeling Language) class diagrams. The interdependence framework considers the relationships among business process-level NFRs, and between the business process and web service levels in a top-down strategy. The ISO/IEC 25010 Product Quality Model is the main reference, enriched by extra quality models on software and QoS attributes.

The main contributions of this work are threefold: *(1)* extensive gathering of NFRs related to business processes and technical aspects of web service provisioning, and the definition of their interdependence relationships aiming to support those who want to systematize their decomposition; *(2)* conceiving a conceptual framework that, despite primarily designed for the context of business process automation via web service orchestration, is generic enough to be used or adapted to other areas considering NFRs; and *(3)* presenting an approach that, while not automatic, can support semi-automated decision making.

The rest of this paper presents the following sections: underlying concepts, research method, obtained results, related work and concluding remarks.

# 2 UNDERLYING CONCEPTS

## 2.1 Non-functional Requirements

In software engineering, Non-functional Requirements (NFRs) are defined as constraints on services or functionalities offered by a system, including characteristics related to software behavior and constraints imposed by standards (Sommerville, 2010). These NFRs are defined based on user needs, budget constraints or external factors, such as regulatory and legislative determinations (Sommerville, 2010). Examples are performance, security and availability.

In Business Process Management (BPM) and Service-Oriented Architecture (SOA), NFRs represent quality aspects for the provisioning of services responsible for executing business processes. These quality aspects set guarantee levels which allow comparison among distinct services with the same functionality (Abramowicz et al., 2006). Web service NFRs are often expressed as QoS attributes and specified usually through SLAs. SLAs refers to a commitment between web service providers and consumers, whereby the exact quality conditions that guide the web service provisioning are systematically defined (Salles et al., 2013)[1]. An SLA could include, for instance, a QoS attribute for availability with a target of 99% and a QoS attribute for response time with a target of 5 ms. In SLAs, penalties and rewards are defined and imposed depending on the breach of predefined guarantee terms.

SLA terms are defined by IT considering technical aspects of web service provisioning (Bratanis et al.,

[1]In this work, only technical aspects involved in a web service provisioning are considered in SLAs; i.e., IT outsourcing or out-tasking web services for higher-level tasks, including human tasks, are not part of the scope.

2010). However, business aspects should be also considered, mainly in the context of business process automation via web service orchestration (Borges et al., 2019). A different type of agreement is then required, and hence the use of BLAs is an alternative.

The structure of a BLA is like of an SLA, including penalties and rewards. The main difference lies in their scopes: while SLAs are associated with web services and consider mainly technical aspects involved in web service provisioning, BLAs are associated with business process activities that will be executed in the form of web services (Borges et al., 2019; Salles et al., 2013). BLAs are defined during business process analysis and modeling whereas SLAs are determined during business process implementation and execution (Salles et al., 2013).

The differences between BLA and SLA are illustrated by Salles et al. (2018) who exemplify a BLA goal with *"the business subprocess starting in the activity [1] and ending in the activity [4] must be concluded within 24 hours"* whereas the corresponding SLAs goals are exemplified with *"the web service invoked to execute the activity [1] must be completed within 2 hours"* and *"the web service invoked to execute the activity [4] must have 95% of availability"*.

A BLA can be mapped to a set of SLAs, each BLA related to a specific business process activity automated through a set of web services with their SLAs (Goel et al., 2011). Assuming that all the guaranteed terms of each SLA are satisfied, the corresponding BLA is expected to be satisfied accordingly.

### 2.2 Software Quality Models

According to the IEEE Standard Glossary of Software Engineering Terminology (IEEE, 1990), software quality means the degree to which a system, component or business process meets specific requirements. Specifying functional requirements and NFRs for software is not a trivial task; a common approach is to use software quality models as a reference to describe and assess software requirements.

Quality models support identifying relevant quality characteristics that can be further established as requirements and their corresponding satisfaction criteria and measures (ISO/IEC, 2010). Quality models provide the fundamentals for software evaluation, offering a consistent QoS terminology and supporting software measurement (Botella et al., 2003).

There are several software quality models proposed in the literature from international standards to several domain and company-specific models. One of the most popular approaches is the ISO/IEC 25010 System and Software Quality Model, which is a part of the ISO/IEC 25000 Software Product Quality Requirements and Evaluation (SQuaRe) model and results from evolving several other standards, especially the ISO/IEC 9126 (ISO/IEC, 2010). ISO/IEC 25010 addresses a set of QoS attributes for software product quality and software quality in use.

Figure 2 shows the structure of ISO/IEC 25010, depicting its eight main quality characteristics and 31 sub-characteristics. Alternative software quality models were proposed (McCall et al., 1977; Boehm et al., 1978; Dromey, 1999). More information about quality models can be found in Miguel et al. (2014); Sheoran and Sangwan (2015); Tomar and Thakare (2011).

Regarding BPM and SOA, quality models can be used as a reference to define requirements related to business processes and web services, allowing for the overall quality improvement of SOA-based applications. To the best of our knowledge, there is no general standard accepted as a quality model for web services being orchestrated to automate business processes. However, web services and software modules share the same set of properties; therefore, if software components can be replaced by web services, then the quality requirements of both solutions must be compatible with (Abramowicz et al., 2009). As a result, software quality models can also be used to address quality characteristics of web services.

Since the ISO/IEC 25010 quality model is a recognized quality standard for any type of software, it can also be used to provide QoS attributes for web services (Abramowicz et al., 2009). Alternative quality models used in the context of web services were proposed (OASIS, 2005; Abramowicz et al., 2008).

## 3 RESEARCH METHOD

This work was developed following principles of the *design science* research paradigm, which considers the creation and evaluation of artifacts to solve identified organizational problems (Hevner et al., 2004). These artifacts need to address an unsolved problem or propose an improvement for an existing solution to more significantly contribute with science and practice (Hevner et al., 2004). In this research, the problem refers to the lack of a systematic structure to support a straightforward decomposition of NFRs, from business to QoS attributes related to web services. However, contrasting the paradigm, this research did not include a validation work to ascertain the results, thus resulting in a theoretical research based on literature analysis. In this context, developing an conceptual interdependence framework for NFRs included two major activities: *(i)* the elaboration of a dictio-


Figure 2: ISO/IEC 25010 Product Quality Model (ISO/IEC, 2010).

nary of NFRs, considering NFRs for both business process and web service levels; and *(ii)* the definition of interdependence relationships between identified NFRs, taking relationships between NFRs at the same level (for the business process level) and relationships between NFRs at different levels (from the business process level to the lower levels).

Regarding the dictionary of NFRs, an exploratory literature study was conducted to elicit a set of quality characteristics related to business or technical aspects of web service provisioning. ISO/IEC 25010 was chosen as the main reference for this work primarily because it is already an official standard consolidated in the literature. In addition, because it is one of the most popular standards related to non-functional requirements used by researchers and business and IT practitioners (Abramowicz et al., 2009). Finally, because it addresses software generically, allowing a mapping to web services, which are specific types of software. The base The structure of characteristics and sub-characteristics provided by IS/IEC 25010 was expanded through extra research on software and web services quality models and studies related to SOA and QoS attributes. The dictionary and the details regarding its elaboration are fully described in a technical report (de Castro and Fantinato, 2018).

The definition of the interdependence relationships among the dictionary's NFRs was based on the studies of Zulzalil et al. (2008); McCall et al. (1977), used as the main references to describe the relationships between the business process-level NFRs. Although they predate the publication of the ISO/IEC 25010, both share the same evaluated characteristics.

With respect to the relationships between NFRs from the business process level to the web service level and also between NFRs at the web service level, the structure of characteristics and sub-characteristics of the ISO/IEC 25010 was also used as the main reference. For most of the NFRs got from other references during the elaboration of the dictionary, the corresponding studies already incorporated some hierarchical classification that could be the basis to define the decomposition structure.

Remaining relationships were determined via logical inference based on empirical analysis. The authors conducted iterative brainstorming meetings to discuss potential relationships between the NFRs mapped in the dictionary. The ideas that came up during these meetings were refined resulting on a final set of relationships, presented as follows.

# 4 NFR DECOMPOSITION

Considering business processes being automated through web service orchestration, a conceptual NFR decomposition framework is proposed to support a straightforward definition of web service QoS attributes. The definition of QoS attributes is carried out based on constraints determined by business areas at process modeling time. Using this approach, IT teams are given hints of which QoS attributes they can assign to a web service to meet a business demand. Thus, the expected users for this framework are IT teams working on the perspective of a web service provider and hence involved in the definition of web service SLAs to be executed by business units.

The designed NFR decomposition framework is presented as follows. An explanation of the framework's structure is given, with details on the dictionary of NFRs and the interdependence diagrams. Then, the decomposition diagrams are shown.

## 4.1 Conceptual Framework Overview

The set of NFRs is organized into a dictionary structure. Two sections form the dictionary of NFRs: one for business process NFRs (cf. Table 1) and another for web service NFRs (cf. Table 2), as detailed by de Castro and Fantinato (2018). The structure of both sections comprises four attributes: **ID**, a numerical NFR identification; **Name**, the NFR name; **Definition**, a brief description of the NFR; and **Reference**, the references of the works from which the NFR was extracted from. Synonyms are identified and grouped using a unique ID. Specifically for web service NFRs, there is an extra attribute, **Measurement Unit**, which identifies the primary unit used to measure a quantitative NFR. The measurement unit was filled in the dictionary only when found in the literature.

Considering the relationships between the NFRs, they are represented through UML class diagrams (cf. Figures 3–11). Each class in the diagram represents an NFR included in the dictionary. Relationships between NFRs at the same level are represented using the *association* bidirectional connector (e.g., Figure 6, association between *Confidentiality* and *Access Control*). Relationships between NFRs of different levels are defined through *aggregation* connectors, i.e., lower-level NFRs contribute to those at a higher level, although they exist independently (e.g., Figure 6, aggregation between *Confidentiality* and *Encryption*).

Each class in the diagram includes a configurable attribute denominated **relevance**, which considers three values: high, medium or low. Business and IT areas should use this attribute to show which NFRs are how likely relevant when creating SLAs in an organization, based on the business domain and previous experiences. As the proposed decomposition framework has been developed to be generic enough to be considered in different organizational contexts, no prior definition of relevance for each NFR is provided. As a result, each organization willing to use this framework should define its own relevance values considering its own context and historical data.

## 4.2 Decomposition Diagrams

The NFR decomposition framework considers attributes to be defined at business process and web service levels, by business and IT areas, respectively. Regarding business process NFRs, the first section of the dictionary includes eight attributes (cf. Table 1).

To identify the business process-level NFRs, the characteristics proposed in the ISO/IEC 25010 (ISO/IEC, 2010) were considered as describing generic aspects of product quality to be selected by business areas (de Castro and Fantinato, 2018). From eight characteristics in the ISO/IEC 25010 model (cf. Figure 2), seven were adapted to be added in the dictionary as business process NFRs: *Performance Efficiency*, *Compatibility*, *Usability*, *Reliability*, *Security*, *Maintainability* and *Portability*. Only the first — i.e., *Functional Suitability* — was not considered as it addresses functional requirements and not NFRs which is the purpose of this dictionary.

ISO/IEC 25010 and other related software quality models describe quality characteristics only from the product perspective. Aiming at completeness for the business context, the dictionary of business process NFRs was extended with an attribute addressing regulatory, legislative and operational aspects involved in business process enactment. This NFR is *Compliance*, adapted from the types of NFRs for software systems presented by Sommerville (2010).

The attributes in Table 1 relate to each other. For instance, the conversion from standard protocols to ensure compatibility may affect performance efficiency (McCall et al., 1977), while a fast maintenance implies in higher recoverability in the presence of errors, improving reliability levels. Identifying interdependence relationships between business process NFRs is relevant to recommend a more complete set of web service NFRs to be defined in SLAs. For example, when business areas define a constraint related to compatibility, the IT team could take care of performance NFRs as well. Figure 3 shows the interdependence relationships between the business process NFRs that were identified in this work.

Some relationships in Figure 3, such as *performance efficiency* vs. *compatibility* and *reliability* vs. *maintainability*, were defined based the literature (Zulzalil et al., 2008; McCall et al., 1977). Others, such as *reliability* vs. *security* and *compliance* vs. *security*, were defined via logical inference based on an empirical analysis by the authors of this work. Table 3 presents a brief explanation of the meaning of the interdependence relationships shown in Figure 3.

Regarding the web service NFRs, the characteristics and sub-characteristics proposed in ISO/IEC 25010 (ISO/IEC, 2010) were considered describing technical aspects of web service provisioning to be defined in SLAs by IT teams. This model was refined by extra works on software quality models and QoS attributes and, as a result, the dictionary of web service NFRs is formed by 93 requirements, each of them related to at least one business process NFR and also related among them. Web service NFRs are also interrelated through a requirements hierarchy. The dictionary of web service NFRs, with definitions, references, measurement unit and additional details regarding its elaboration, is fully described in a technical report (de Castro and Fantinato, 2018).

The NFR decomposition framework is split into eight decomposition diagrams (cf. Figures 4-11), one for each business process NFR (cf. Figure 3).

A illustrative excerpt of the dictionary of web service NFRs is shown in Table 2. The NFRs in Table 2 are related to *performance efficiency*, with which seven other NFRs describing web service's time behavior, resource utilization and capacity are associated. Contrasting the dictionary of business process NFRs, shown in Table 1, the measurement unit is de-

Table 1: Dictionary of business process NFRs (de Castro and Fantinato, 2018).

| ID | Name | Definition | Reference |
|---|---|---|---|
| 1 | Performance Efficiency | Degree to which a business process can efficiently use an amount of resources (such as software, products, hardware and generic materials) under stated conditions. | (ISO/IEC, 2010) |
| 2 | Compatibility | Degree to which a business process can exchange information with other business processes, and/or perform its activities while sharing the computing environment. | (ISO/IEC, 2010) |
| 3 | Usability | Degree to which a business process can be used by specified users to achieve specific goals with effectiveness, efficiency and satisfaction. | (ISO/IEC, 2010) |
| 4 | Reliability | Degree to which a business process performs specified activities under specified conditions for a period. | (ISO/IEC, 2010) |
| 5 | Security | Degree to which a business process can protect information and data from unauthorized access. | (ISO/IEC, 2010) |
| 6 | Maintainability | Degree of effectiveness and efficiency with which the activities of a business process can be modified. | (ISO/IEC, 2010) |
| 7 | Portability | Degree of effectiveness and efficiency with which a business process can be configured in an environment and transferred from one environment to another. | (ISO/IEC, 2010) |
| 8 | Compliance | Degree to which a business process is compliant with internal procedures of an organization and external guidelines. | (Sommerville, 2010) |

Table 2: Excerpt of the web services' dictionary of NFRs — performance efficiency (de Castro and Fantinato, 2018).

| ID | Name | Definition | Meas. Unit | Ref |
|---|---|---|---|---|
| 1 | Time behavior | Degree to which the response and processing times and throughput rate meet requirements in web service provisioning. | — | (ISO/IEC, 2010) |
| 2 | Resource utilization | Degree to which the amount and type of resources used meet requirements in web service provisioning. | — | (ISO/IEC, 2010) |
| 3 | Capacity | Maximum limits of a web service (i.e., concurrent users, stored data etc.) for which performance is guaranteed. | — | (ISO/IEC, 2010) |
| 4 | Latency time | Round-Trip Delay (RTD) between the dispatch of a request and receive of a response for a web service. | Time | (Abramowicz et al., 2006) |
| 5 | Execution time | Time for a web service to execute a sequence of activities and process a request. | Time | (Lee et al., 2003) |
| 6 | Response time | Time necessary to complete a certain web service request, from the moment it is dispatched until a response is received. | Time | (Lee et al., 2003) |
| | [Average and maximum response time] | The average time needed for the packet of control data to get to the provider's server and return to the requester. | Time | (Abramowicz et al., 2006) |
| | [Execution duration] | Expected delay from the dispatch of a web service request until the result is received by the client. | Time | (Zeng et al., 2003) |

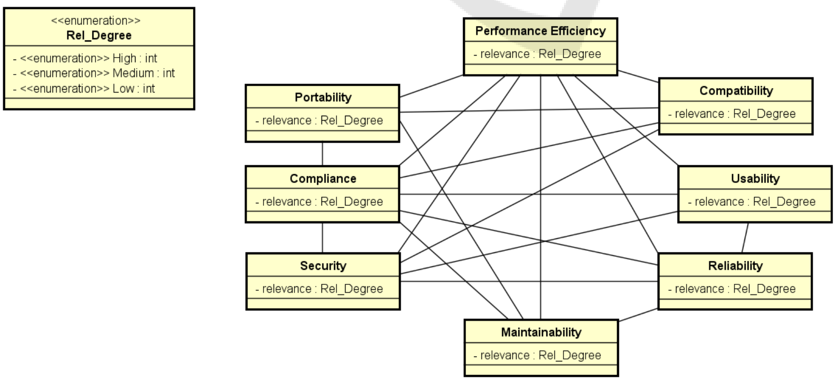


Figure 3: Interdependence relationships for business process NFRs.

fined here for some web service NFRs. Synonyms from different references are grouped in a unique ID, which is the case of *Response time*, *Average and maximum response time* and *Execution duration*.

Figure 4 shows the decomposition diagram for *performance efficiency*. The NFRs and most of the relationships between them were extracted from ISO/IEC (2002, 2010); Abramowicz et al. (2006); Lee

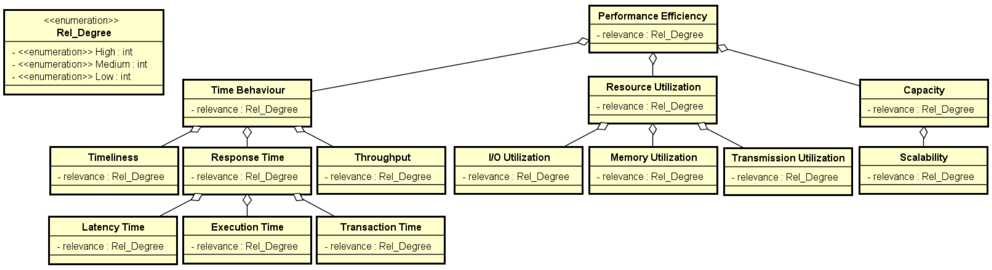


Figure 4: NFR decomposition diagram — performance efficiency.

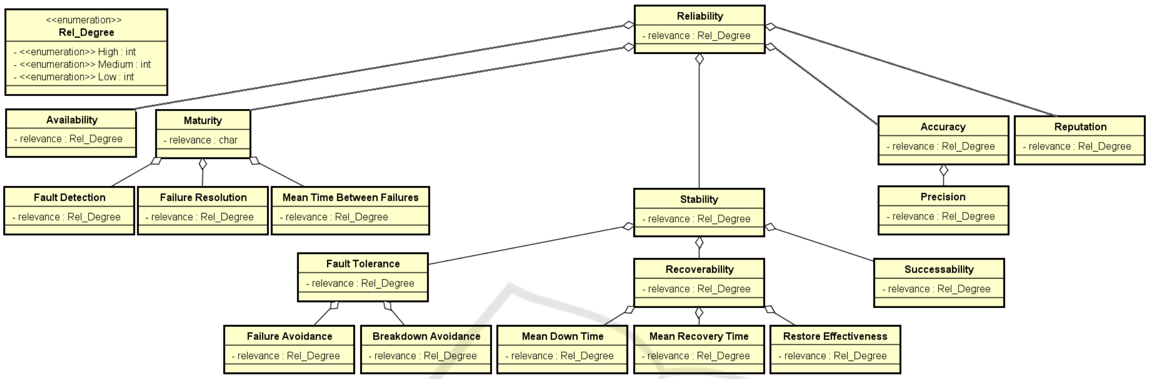


Figure 5: NFR decomposition diagram — reliability.



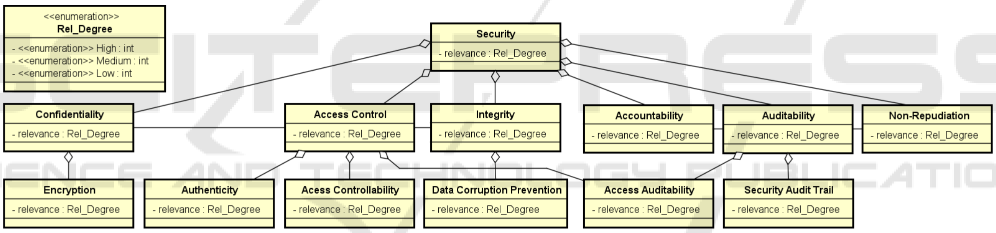


Figure 6: NFR decomposition diagram — security.

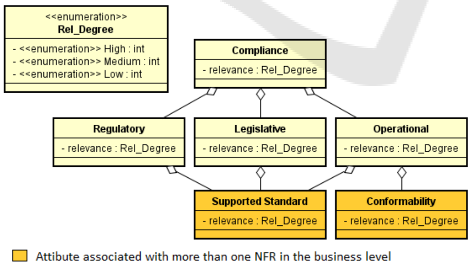


Figure 7: NFR decomposition diagram — compliance.

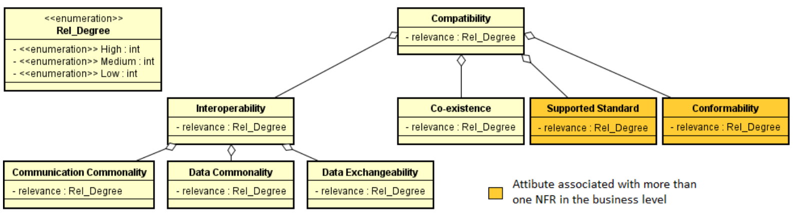


Figure 8: NFR decomposition diagram — compatibility.

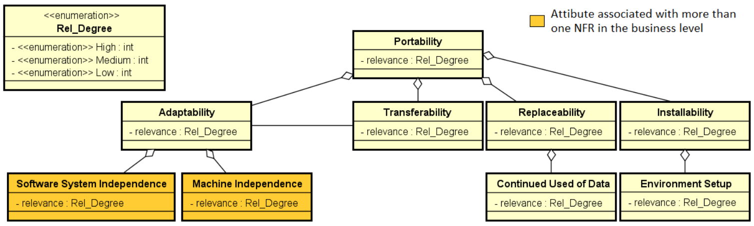


Figure 9: NFR decomposition diagram — portability.

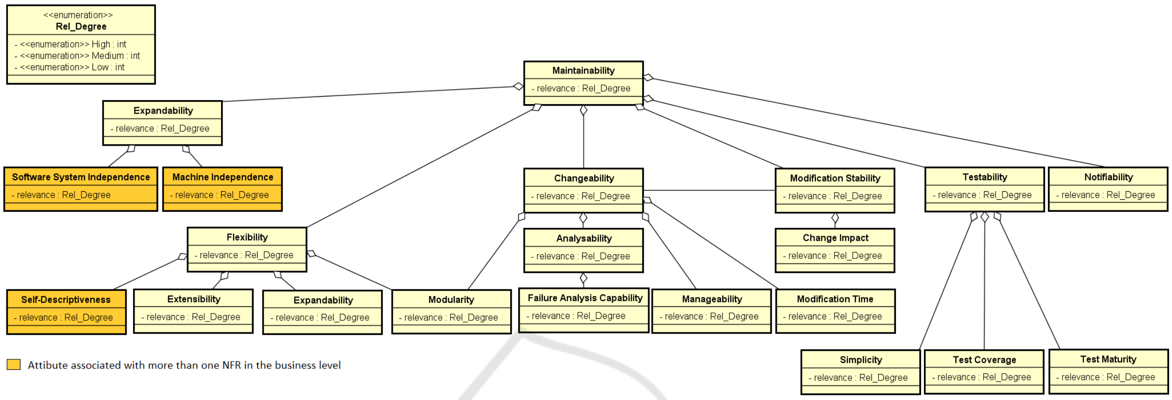


Figure 10: NFR decomposition diagram — maintainability.

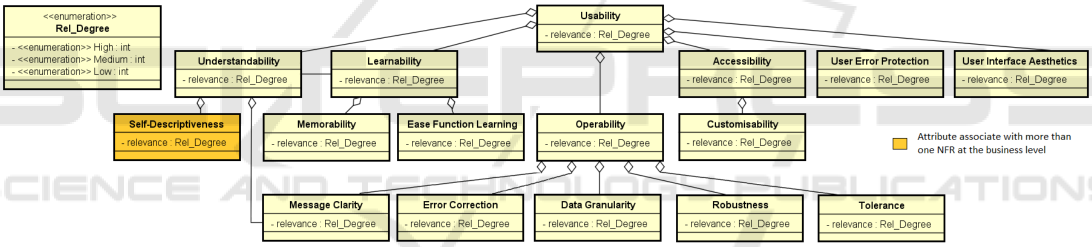


Figure 11: NFR decomposition diagram — usability.

et al. (2003); OASIS (2005); O'Brien et al. (2005). For instance, the relationships between the attributes related to *resource utilization*. Other relationships were mapped from related studies or defined via logical inference. The attributes *latency time*, *execution time* and *transaction time*, for instance, are associated with *response time* considering the definition of the latter: "response time is defined as the time required to complete a web service request" (Lee et al., 2003). Thus, response time should include the round-trip delay for the network propagation (i.e., the latency time) plus the execution time required to process the request in the provider (i.e., the execution time). If transactions are processed, it should also consider the time to complete the transaction (i.e., the transaction time).

Figures 5 and 6 show the decomposition diagrams for *reliability* and *security*, respectively. As for reliability, 17 web service NFRs are presented, mainly related to web service availability and stability in the presence of failures. The NFRs and most of the relationships were extracted from ISO/IEC (2002, 2010); McCall et al. (1977); OASIS (2005); Sheoran and Sangwan (2015); Zeng et al. (2003). For instance, the attributes *fault detection*, *failure resolution* and *mean time between failures* as associated with *maturity*. Other relationships were defined via logical inference, such as associating *fault tolerance*, *recoverability* and *successability* in meeting requests as attributes related to web service *stability*.

Regarding security, 12 web service NFRs are presented, mainly related to data confidentiality and integrity, access control and traceability. The NFRs and some relationships were extracted from the literature (ISO/IEC, 2002, 2010; McCall et al., 1977; OASIS, 2005; O'Brien et al., 2005; Pettersson, 2007). Other relationships, such as the bidirectional association between *confidentiality* and *access control*, were defined via logical inference. For the latter, we considered that confidentiality requires that data should be read only by those with access to it, implying in access

Table 3: Details of interdependence relationships between business process NFRs.

| NFR1 | NFR2 | Relationship explanation | Reference |
|---|---|---|---|
| Perf. Effic. | Compatib. | The conversion from standard protocols to ensure compatibility between web services may affect performance efficiency. | (McCall et al., 1977) |
| Perf. Effic. | Usability | The additional code and processing required to ease an operator's task or provide more usable output may affect performance efficiency. | (McCall et al., 1977) |
| Perf. Effic. | Maintain. | The need of using modularity, instrumentation and well commented high-level code to increase maintainability may affect performance efficiency. | (McCall et al., 1977) |
| Perf. Effic. | Security | The additional code and processing required to control the access of a web service or data may affect performance efficiency. | (McCall et al., 1977) |
| Perf. Effic. | Portability | Using direct code or optimized system software to increase performance may affect a web service's portability. | (McCall et al., 1977) |
| Perf. Effic. | Reliability | The implementation of strategies to increase web service's availability may affect performance efficiency. | — |
| Compatib. | Security | Coupled systems or web services can be accessed by different users, increasing the potential for accidental access of sensitive data and thus affecting security requirements. | (McCall et al., 1977) |
| Compatib. | Portability | The guarantee of the web service's compatibility may affect portability requirements in terms of platform independence. | — |
| Usability | Reliability [+Security] | The implementation of error prevention functions in a web service's interface may affect its maturity and stability in terms of fault detection and tolerance. | (Zulzalil et al., 2008) |
| Reliability | Maintain. | Increasing maintainability usually affects a web service's reliability, as it turns easier for a web service to be maintained in case of breakdown. | (Zulzalil et al., 2008) |
| Reliability | Security | The security of a web service may affect its reliability in terms of stability and reputation. | — |
| Maintain. | Portability | Increasing maintainability may affect the effort to transfer a web service from one operating environment to another. | (Zulzalil et al., 2008) |
| Compliance | * | Regulatory, legislative or operational guidelines can be applied to all seven remaining NFRs included in the framework. | — |

control. The association between *access control* and *integrity*, on the other hand, was proposed in the literature with a similar argument (McCall et al., 1977).

Figures 7 and 8 show the decomposition diagrams for *compliance* and *compatibility*, respectively. Regarding *compliance*, the NFRs were extracted from the literature (OASIS, 2005; Ran, 2003; Sommerville, 2010; Yoon et al., 2004) and all the relationships were defined via logical inference. On the other hand, the NFRs and most of the relationships for *compatibility* were extracted from ISO/IEC (2002, 2010); McCall et al. (1977), as the attributes associated with *interoperability*. The attributes *supported standard* and *conformability* were identified by the authors as being related to both *compliance* and *compatibility* and hence are shown in both diagrams in orange. The definition of this dual association considered a scenario where technical standards must be addressed in web service provisioning, as a demand of regulators, organizations or the government itself. When these standards are related to the communication between systems, this attribute may also be defined in terms of compatibility. Conformability is the degree to which the pre-defined standards are met and hence considered in both cases.

Figures 9, 10 and 11 show the decomposition diagrams for *portability*, *maintainability* and *usability*, respectively. The NFRs and some relationships defined in these diagrams were mainly extracted from Abramowicz et al. (2006); ISO/IEC (2002, 2010); McCall et al. (1977); Miguel et al. (2014); OASIS (2005); O'Brien et al. (2005); Pettersson (2007); Sheoran and Sangwan (2015). The attributes *software system independence* and *machine independence* are related to both *maintainability* and *portability*, and shown in both diagrams in orange. This dual classification considered that, the more a system is independent of the computational environment, the easier it is to adapt its operation to different environments and reuse its components in different contexts. Likewise, the attribute *self-descriptiveness* is classified as related to both *maintainability* and *usability*, considering that the clearer a web service and its documentation is, the easier it is to maintain and operate it.

## 4.3 Examples of NFR Decomposition

Using the illustrative example of business process presented in Figure 1, some NFRs can be defined to the web services that will implement such a process.

Consider the BLA related to *performance efficiency* associated with the set of highlighted activities. Some levels for web service QoS attributes should be pursued in order to ensure that this BLA is met, and the decomposition framework proposed herein can be used for this purpose. Considering that for each activity one or more web services can be used, different QoS attributes can be defined for each web service, depending on the needs identified by the analysts involved. Per Figure 4, to meet the BLA *performance efficiency* for these three activities, 13 distinct QoS attributes may be associated with the web services that will be used to implement them. For example, for some web services, QoS attributes related to *response time* or *throughput* may be defined, associated with some target values; i.e., the QoS levels. IT teams may also understand that, for some web services, they should use a more specific QoS attribute related to *latency time*, *execution time* or *transaction time*.

Besides the direct relationships addressed in the previous example, indirect QoS attributes can also be defined as they can also interfere in the performance efficiency of these three activities. For example, according to Figure 3, *performance efficiency* is related to *reliability*. Thus, web services that implement one or more of the three activities in Figure 1 may also have QoS attributes associated with *reliability*, as they may also affect the performance efficiency of the business process, as explained in Table 3. An example would be to define a QoS attribute related to *availability* (cf. Figure 5) because the business process will be delayed if the web service is not available.

Still taking the example in Figure 1, another BLA associated with the three activities being addressed is related to *security*. Using Figure 6 as the main reference for the decomposition of security-related NFRs, 12 QoS attributes are suggested as ideas for the IT team to include in the corresponding SLAs. From these 12 QoS attributes, eight are leaf nodes, i.e., considered the most specific attributes, whereas four are intermediate nodes, i.e., more generic ones. Any attribute level can be used, depending on the needs perceived by the IT team. The business team should provide detailed information related to the security BLA, explaining exactly what the requirement means, so IT teams can choose the most appropriate QoS attributes to be associated with the web services, such as *access control*, *encryption*, *auditability* and so on. The information from the business area is also relevant to allow identifying which web services will need QoS attributes or not. Other QoS attributes can be chosen by referring to Figures 4, 5, 7, 8 and 11 as the attributes related to *performance efficiency*, *reliability*, *compliance*, *compatibility* and *usability* are indirectly related to *security* (cf. Figure 3).

# 5 RELATED WORK

Several studies have addressed business process automation with SOA. However, only a few of them discusses the relationship between business process NFRs and web service QoS attributes.

Still in 2005, the particularities involved in using quality requirements originally defined as software quality models were discussed for SOA (O'Brien et al., 2005). SOA and underlying concepts were explained in detail, examining their impact on meeting business goals in organizations. A structured list of QoS attributes used in SOA was provided.

Then, in 2008, the relationship of ISO/IEC 9126 quality characteristics (ISO/IEC 25000 predecessor) was investigated to develop web-based applications (Zulzalil et al., 2008). Eliciting information from stakeholders with an online survey, interactions between pairs of quality characteristics were identified, considering three possible relationships: positive, negative and independent. This approach enabled to understand how software quality aspects influence each other, contributing to the elaboration of a quality model that combines individual QoS attributes based on specific relationships.

In 2009, a related approach was proposed to address quality requirements expressed at the level of SOA applications and break them down to the level of components used to create the applications, i.e., at the web service level (Abramowicz et al., 2009). The structure used for decomposition considers two ontologies: *(i)* SQuaRE-based SOA Quality Ontology, with 14 high-level quality characteristics extracted from ISO/IEC 25000 SQuaRE quality model and is used by business users; and *(ii)* Semantic Web Service QoS Ontology, with a set of qualitative and quantitative QoS attributes related to web services. This approach supports a more direct decomposition of high-level QoS attributes into detailed preferences, in a task less dependent on an IT expert's knowledge.

Discussions for the business process level were introduced by showing the need for defining functional requirements and NFRs of web service provisioning in a different type of agreement, which is the aforementioned BLA (Bratanis et al., 2010).

The Strategic Alignment with BPM (StrAli-BPM) framework was proposed in 2013 to foster the strategic alignment between business and IT, when executing web service-based business processes (Salles et al., 2013, 2018). StrAli-BPM was extended (Carmo et al., 2017; Borges et al., 2019) and is formed by four components, one of which is particularly related to the work being presented herein — the BLA2SLA component. BLA2SLA considers a top-down strategy for a generic decomposition of business process NFRs (represented by BLAs) into web service QoS attributes (represented by SLAs). The definition of BLAs and SLAs is supported by meta-models, each including a set of attributes related to the business process and web service levels, respectively. BLA2SLA enables the use of a standardized structure to define NFRs for web services based on business needs.

Many of the related work focuses on only discussing individual quality aspects of software components and web services, regardless of their relationship to NFRs of the business processes being automated. The BLA2SLA (Salles et al., 2013, 2018) component is closer to the aim sought in this work but using a loose relationship between the addressed concepts: business areas define a BLA, but IT teams only receive this BLA as a reference and must use their experience to define the SLAs considered more appropriate to address this BLA.

The decomposition model proposed by Abramowicz et al. (2008) considers the relationships between high-level and low-level quality characteristics to semi-automate the derivation of QoS attributes. However, they assume NFRs defined from scratch by business users in the SOA development. This assumption contrasts with the work being presented herein, which assumes that specific activities of business processes (to be automated through web services) are associated with business goals. Their proposed structure considers mainly product quality characteristics to be selected by business users, disregarding organizational and external requirements (i.e., regulatory and legislative) that might be demanded by business areas in software applications. For some of those characteristics, no related QoS attribute was mapped to be defined at the web service level. Ultimately, the relationships among high-level quality characteristics themselves were not considered, resulting in each characteristic being examined only individually.

In the approach of Zulzalil et al. (2008), only the relationships among high-level quality characteristics themselves were explored, disregarding their relationship with web service QoS attributes. Finally, despite presenting an analysis of QoS attributes in SOA, O'Brien et al. (2005) did also not consider the relationships with business process NFRs.

## 6 CONCLUDING REMARKS

In this paper, we presented a conceptual framework to support a straightforward decomposition of business process NFRs (which can be formalized in BLAs) into web service NFRs (which can be formalized in SLAs grouping QoS attributes). Our main goal was to provide a systematization to support IT teams in breaking down NFRs defined by business areas into technical aspects for web service provisioning.

Due to the lack of standardized quality models specific to business processes and web services, software quality models had to be used to build the dictionary. Since web services are a type of software component, general software product quality models should work properly also for this particular case. Thus, the quality models from which NFRs were elicited are expected to have no negative influence on the results of this study. In addition, due to the lack of studies regarding the interdependence relationships between software and web service quality attributes, some relationships on the decomposition diagrams were designed considering logical inferences on the NFR interdependence, based on empirical analysis.

This preliminary study presents a promising model to provide a common understanding between the business side of an organization and the IT team eligible for developing web services to automate the business processes necessary by business units. To the best of our knowledge, this is the first attempt to present a detailed model for decomposing business process NFRs into QoS attributes, considering an extensive dictionary with about 100 attributes. This framework provides a basis for facilitating the decomposition of BLAs into SLAs and attenuating the dependence of an IT expert's knowledge in defining which attributes should be considered in a web service based on the constraints of a business process.

Future studies should be devoted to further adjusting and validating the proposed framework. We plan to investigate the NFR Framework (Chung et al., 1995) as a candidate to replace the UML Class Diagrams. For the validation, we intend to extend the prototype proposed in Salles et al. (2013, 2018) with the decomposition diagrams proposed herein, addressing the relationships of interdependence between NFRs, in the context of the StrAli-BPM approach. Other possibilities are using experts and case studies.

## ACKNOWLEDGEMENTS

This work was funded by Fapesp, Brazil (grants 2017/26491-1 and 2017/26487-4, holder Marcelo Fantinato) and Capes, Brazil (grants 88881.172071/2018-01, holder Lucinéia H. Thom).

## REFERENCES


Abramowicz, W., Haniewicz, K., Hofman, R., Kaczmarek, M., and Zyskowski, D. (2009). Decomposition of SQuaRE-based requirements for the needs of soa applications. In *Int. Conf. on Advances in Commun. Techn. and Engin. Sci.*, pages 81–94.

Abramowicz, W., Hofman, R., Suryn, W., and Dominik, Z. (2008). SQuaRE based web services quality model. In *Int. MultiConf. of Engin. and Comput. Scient.*

Abramowicz, W., Kaczmarek, M., and Zyskowski, D. (2006). Duality in web services reliability. In *Adv. Int. Conf. on Telecom. and Int. Conf. on Int. and Web Applic. and Serv.*, pages 165.1–165.6.

Barros, V. A., Fantinato, M., Salles, G. M. B., and de Albuquerque, J. P. (2014). Deriving service level agreements from business level agreements: An approach towards strategic alignment in organizations. In *16th Int. Conf. on Enter. Inf. Syst. (ICEIS)*, pages 214–225.

Boehm, B. W., Brown, J. R., Kaspar, H., and Lipow, M. (1978). *Characteristics of software quality*. TRW Softw. Technol. NH, Amsterdam.

Borges, E. S., Fantinato, M., Aksu, U., Reijers, H. A., and Thom, L. H. (2019). Monitoring of non-functional requirements of business processes basedon quality of service attributes of web services. In *21st Int. Conf. on Enter. Inf. Syst. (ICEIS)*.

Botella, Pereand Burgués, X., Carvallo, J. P., Franch, X., Pastor, J. A., and Quer, C. (2003). Towards a quality model for the selection of ERP systems. In Cechich, A., Piattini, M., and Vallecillo, A., editors, *Component-Based Software Quality: Methods and Techniques*, pages 225–245.

Bratanis, K., Dranidis, D., and J. H. Simons, A. (2010). Towards run-time monitoring of web services conformance to business-level agreements. In *5th Int. Acad. and Ind. Conf. on Testing – Practice and Research Techniques*, pages 203–206.

Carmo, A., Fantinato, M., Thom, L., Prado, E. P. V., Spínola, M., and Hung, P. C. K. (2017). An analysis of strategic goals and non-functional requirements in business process management. Master's thesis.

Chung, L., Nixon, B. A., and Yu, E. (1995). Using non-functional requirements to systematically support change. In *2nd IEEE Int. Symp. on Req. Eng.*, pages 132–139.

de Castro, C. F. and Fantinato, M. (2018). Dictionary of non-functional requirements of business process and web services. Technical Report 003/2018, Graduate Program of Information Systems, Univ. of São Paulo.

Dromey, R. G. (1999). Software product quality: Theory, model, and practices. Technical report, Soft. Quality Institute, Griffith Univ., Brisbane, Australia.

Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. A. (2013). *Fundamentals of Business Process Management*. 2 edition.

Goel, N., Kumar, N. V. N., and Shyamasundar, R. K. (2011). SLA monitor: A system for dynamic monitoring of adaptive web services. In *9th Eur. Conf. on Web Serv.*, pages 109–116.

Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–105.

IEEE (1990). IEEE standard glossary of software engineering terminology, IEEE Std 610.12-1990.

ISO/IEC (2002). ISO/IEC 9126 software product quality, IEEE Std 9126:2002.

ISO/IEC (2010). ISO/IEC 25010 system and software quality models.

Lee, K., Jeon, J., Lee, W., Jeong, S.-H., and Park, S.-W. (2003). QoS for web services: Requirements and possible approaches. Technical Report NOTE-ws-qos-20031125, W3C Korea Office.

McCall, J. A., Richards, P. K., and Walters, G. F. (1977). Factors in software quality. volume-iii. preliminary handbook on software quality for an acquisiton manager. Technical Report RADC-TR-77-369, Defense Technical Information Center.

Miguel, J. P., Mauricio, D., and Rodriguez, G. D. (2014). A review of software quality models for the evaluation of software products. *Int. J. of Soft. Eng. & Applic.*, 5(6):31–54.

OASIS (2005). Quality model for web services.

O'Brien, L., Bass, L., and Merson, P. (2005). Quality attributes and service-oriented architectures. Technical Report CMU/SEI-2005-TN-014, Soft. Engin. Inst., Carnegie Mellon Univ.

Pettersson, A. (2007). Service-oriented architecture (SOA) quality attributes – a research model. Master's thesis, Dept. of Informatics, Lunds Univ., Sweden.

Ran, S. (2003). A model for web services discovery with QoS. *ACM SIGecom Exchanges*, 4(1):1–10.

Salles, G. M. B., Fantinato, M., Barros, V. A., and de Albuquerque, J. P. (2018). Evaluation of the strali-bpm approach: Strategic alignment with bpm using agreements in different levels. *Int. J. of Bus. Inf. Syst.*, 27(4):433–465.

Salles, G. M. B., Fantinato, M., de Albuquerque, J. P., and Nishijima, M. (2013). A contribution to organizational and operational strategic alignment: Incorporating business level agreements into business process modeling. In *IEEE Int. Conf. on Serv. Comp.*, pages 17–24.

Sheoran, K. and Sangwan, O. P. (2015). An insight of software quality models applied in predicting software quality attributes: A comparative analysis. In *4th Int. Conf. on Reliab., Infocom Techn. and Optim.*, pages 1–5.

Sommerville, I. (2010). *Software Engineering*. Pearson, Addison-Wesley, 9 edition.

Tomar, A. B. and Thakare, V. M. (2011). A systematic study of software quality models. *Int. J. of Soft. Eng. & Applic.*, 2(4):61–70.

Yoon, S., Kim, D., and Han, S. (2004). WS-QDL containing static, dynamic, and statistical factors of web services quality. In *Int. Conf. on Web Serv.*, pages 808–809.

Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., and Sheng, Q. Z. (2003). Quality driven web services composition. In *12th Int. Conf. on WWW*, pages 411–421.

Zulzalil, H., Ghani, A. A. A., Selamat, M. H., and Mahmod, R. (2008). A case study to identify quality attributes relationships for web based applications. *Int. J. of Comp. Sci. and Network Security*, 8(11):215–220.