SQuaRE based Web Services Quality Model

Witold Abramowicz[1] Radosław Hofman[2] Witold Suryn[3] Dominik Zyskowski[4]

Abstract—*.Growing Web Service usage, design and composition methods require precise and reliable information about Web Services quality. Such quality description has to be compatible with universal software quality models, so designers will be able to gather and decompose quality requirements addressing them to different IT solution components, including used Web Services. This article proposes Web Services Quality Model based on ISO/IEC Software product Quality Requirements and Evaluation (SQuaRE) model. Model consists of three perspectives Internal Web Service Quality, External Web Service Quality and Web Service Quality in Use. Prior to model definitions the authors present extended search on quality related issues in literature regarding Web Services and software in general. Authors refer to general need for measuring and publication of Web Service Quality measures considering limited trust and temporal character of measures. This article does not include technical solutions, but focuses on quality model showing its relevance to business needs.*

Index terms—*.Web Service, SQuaRE, QoS.*

I. Introduction

Software Engineering emerged in 1960's as an answer to software quality problems occurring at that time. Software product differed from other human industry products mainly because it was intangible and because its static attributes (attributes that can be marked for product without using it) were irrelevant while dynamic attributes (attributes describing behavior of product when used under certain conditions) were of highest importance. Software usage grew until it started to be used in almost every area of human activity and of course importance of software quality grew and is still growing. From 1970's until now many software quality models appeared – the newest SQuaRe model is still under development.

In the same way software and its quality was new set of product with its intangible factors in 1990's the new product appeared. It was Web Service, which was designed as an intangible product to be used dynamically by software. Main idea is to create framework, where many providers offer the same service and customer may take freedom of choice [44], [64]. This product is again an entrance into intangibleness - customer receives only output from invocation, although hopes and expectations from this product are vast. They are considered as next generation of software [48], [28], [25].

Considering Web Service as a product, and analyzing customers choosing procedure researchers discovered, that one of key selection factors beside semantic appropriateness is the quality [38].

Literature regarding Web Services quality is vast and will be discussed it in next session. Service quality is commonly named Quality of Service (QoS). This term originally relates to transportation, delays and distortions in service delivery, as defined by network providers [11] but some authors extend this term with cost, payments and other characteristics important for choice process but hardly related to quality (for example [35], [26], [7], [52]). What may be surprising is that authors hardly focus on the problem if Web Service actually does what is meant to do (appropriateness to its semantic description) and problems of quality in response omitting Quality of End-user Experience. Some authors add dimension for Quality of Response but most of authors refer only to technical parameters of QoS.

Good example, although in different context is provided in [31] where authors consider three different compositions for user objective which is translation from Byelorussian to Turkish language (first: Byelorussian to English and English to Turkish, second: Byelorussian to German and German to Turkish, third: Belorussian to German, German to English and English to Turkish). Their compositions of underpinning Web Services analyze reliability of Web Services showing how compositors should react if Web Service call fails. But what happens when Web Service answers, but answer is absolutely unacceptable. In the above example if user is unable to make translation between those languages he neither can verify and accept transition nor the final text of document. On the other hand if one limits Web Services usage only to simple services, which results may be verified by requester, then users may alternatively prepare their own software from the scratch to replace potential Web Service call (see [31]). In other case, Web Service requester is unable to verify response, and this response may have significant impact on software operation, accuracy of general results for user and may influence business of the requester.

Authors will show that majority of literature discuss (technically) how to pass information about service quality [2], [44], [5], [31], [38], [59] paying little attention to meaning of quality itself. In [29] authors declare will to define quality of service at requirements level, but again they enter level of details for restricted set of characteristics. Following this specification there are set of frameworks and algorithms proposed for supporting choice processes [1], [40], [45], [61]. The other important stream in literature focuses on reliability of descriptions. There are several approaches [1], [31], [27], [48], [54], [38], [60], [37], [52] for

[1] Professor and Chair of the Department of Information Systems at The Poznań University of Economics, email: witold.abramowicz@kie.ae.poznan.pl
[2] EUR ING, PhD Student Department of Information Systems at The Poznań University of Economics, Polish Standardization Committee Member, email: radekh@teycom.pl
[3] Professeur at École de Technologie Supérieure, Dept. de Genie Logiciel et TI, Montréal, Québec, Secretary of ISO/IEC JTC1/SC7 - Software and System Engineering committee, email: witold.suryn@etsmtl.ca
[4] PhD Student Department of Information Systems at The Poznań University of Economics, email: d.zyskowski@kie.ae.poznan.pl

gathering and trustworthiness of quality measures. Authors will not evaluate this frameworks neither they will propose technical solutions in this area, but they will concentrate on question "How to define Web Service quality". To answer this question authors will discuss purpose for quality definitions basing on experiences in software quality models definitions.

Motivation for proposing new Web Services Quality model comes from observation of software design process. Having quality requirements gathered from user perspective designer has to decompose them addressing requirements to components. If part of application is to be supported by Web Service then designer has to decompose software quality requirements to Web Service quality requirements. Growing usage of Web Services [64], [46] demands standardization in this area and requires compliance between software and Web Service quality models. Proposed model offers this compliance.

II. Background

A. Software Quality Models

From 1960's development of software was perceived as discipline of engineering and these was the beginning for all attempts to define goals and measures for software. One of the most important measures was software quality, but from the beginning it was more then difficult to create unambiguous definition of this quality.

Defining and measuring quality was subject of discussion even in ancient time. Aristotle [4] proposed his own definition, Hammurabi Code and Roman trade rules [10] included quality of product as part of transaction requirements. Software product, despite these long lasing experiences, opened new set of questions. Any known measures (weight, size, durability, water resistance etc.) could not be applied. In 1970 there appeared two differing models of software quality: product oriented McCall model [39] and utility oriented Boehm model [6]. First quality model considered as a standard was developed and published by International Standardization Organization in 1991 as ISO/IEC 9126 [20]. Ten years later new edition of this standard reviewing quality model and introducing three perspectives: quality in use, external quality and internal quality appeared. In the same time new international initiative Software product QUality Requirements and Evaluation (SQuaRE) was set up aiming to develop set of norms ISO/IEC 25000 [17]. This new approach is perceived as new generation of software quality models [8].

Quality related issues are often consisting of mixture of different quality meanings. This problem was noticed since first publications about quality and important summary of five quality views was made in [30]:

1. Quality as an abstract, meta-physical term – unreachable ideal, which shows the direction where to products are heading but will never get there,
2. Quality as a perspective of user considering attributes of software in special context of use,
3. Quality as a perspective of manufacturer, seen as compliance with stated requirements and following ISO 9001:1994 [19] view,
4. Quality as a product perspective understood as internal characteristic, resulting from measures of product attributes,
5. Quality as a value based perspective, differing in dependence on stakeholder for whom is defined.

History and experience gathered within Software Engineering was significant input to proposed quality model. Mentioned above views of quality may be used as one of explanations why different authors focus on different issues of quality (see table 1) – depending on authors background and role in software related processes his starting point for quality definition may be based on certain perspective..

B. Service Description

A Web Service may be defined as a computational entity accessible over the Internet (using particular standards and protocols) [47]. The focus is assigned here to the way that the requester and provider interact with each other [32]. A Web Service is a technical implementation, an interface to a real-world service defined as a certain activity undertaken on behalf of a certain entity. Final users are in fact more interested in a real service they get, rather than in the interface itself. Hence, Web Services may be considered as two inseparable parts – a technical interface (described using Web Service Description Language (WSDL)) and a real (business) functionality (described by other means) the interface provides an access to.

The above perception of a service and Web Services entails a question what kind of a Web Services description is required. A Web Service is an interface used to request the actual provisioning of a real-world service fulfilling the requester needs. Therefore, e.g. in order to discover a Web Service and use it to request a required service, the technical description of a Web Service (the interface) is of course crucial, but not sufficient. What is also indispensable is the description of a real world service and non-functional properties of both a service and a Web Service. The way consumers interact with traditional services and their requirements regarding their description, are a result of social and economic interactions that have been taking place for many years. If Web Service providers do not consider this fact, they will fail. Therefore, a Web Service description should adhere to the well-established requirements of the consumers and cover not only functional, but also non-functional properties (NFP) of a service [2], [1].

When reviewing the Web Service description initiatives at least two things should be taken into account, namely the scope of such a description and formalism used to express it. There are many initiatives in the area of service description.

A Semantic Web Services representation is currently shaped by three main initiatives – OWL-S (Ontology Web Language for Web Services) [36], WSMO (Web Services Modelling Ontology) [49] and SAWSDL (Semantic Annotations for WSDL) [13]. The two first share a quite similar approach to a SWS description (in order to automate Web Services interactions they define the description of a service using standardized language) and in some sense are interoperable due to the activities of the research communities [3]

OWL-S has evolved from the early works of DAML-S [15] and uses the Web Ontology Language (OWL) as the description language upon which an ontology that models a Web Service is built. WSMO uses WSML (Web Services Modeling Language) for defining concepts and ontology building. The difference between OWL-S and WSMO lies in the scope they describe additional interactions that are envisioned and supported. For instance, WSMO foresees a need of providing mediators that serve a purpose of making two or more Web Services interoperable when they differ e.g. in a number and type of parameters.

The distinct approach to Web Service description, which has recently gained the momentum, is SAWSDL [13]. It does not point to a concrete ontology or description language, but allows adding semantic annotations to the WSDL elements. It should be observed that it abandons the paradigm of a template that can be enhanced in favor of in-place semantic annotations done with a use of favorite description tool. This kind of light-weighted solution seems to be quite popular, as adding semantics to WSDL is easier than creating WSMO or OWL-S descriptions from scratch and does not impose in fact additional learning.

The work in the area of SWS description is in advanced stage, yet not all issues have been addressed (e.g. description of the non-functional side of a service [2]).

Functional properties represent capability of a Web Service. These properties are mainly related to the input and output parameters as well as constraints/a state of the world before and after service execution. Therefore, in most cases either the functionality is expressed as only information on inputs and outputs (like in WSDL where input and output parameters required by a service are defined) or as the semantically annotated quadruple IOPE (inputs, outputs, preconditions and effects) in OWL-S or pre- and post-conditions defined within WSMO.

In turn, the non-functional properties play a crucial role in almost all service interactions (to mention only selection, discovery and filtering). Non-functional properties of a service may be defined as anything that exhibits a constraint over the functionality [26]. In fact, non-functional parameters are distinctive criteria for the success of businesses offering their services using Web Services technology. They allow differentiating between Web services offering the same (or quite similar) functionality as in most cases service substitutes differ when it comes to the values of specific non-functional properties. Their role became even more important, as nowadays Web Services are not only used internally, but also support collaboration between various organizations. In consequence, final users (especially business users) desire to know in advance the real quality and non-functional properties of external services they are to use.

The non-functional parameters may be represented as qualitative and quantitative measures of a Web Service (or a service). The non-quantitative ones include security or transactions, whereas quantitative ones include such attributes as cost or time. NFP should of course include business constraints and inter-service dependencies, if possible. However, different types of services require different properties describing them and which properties are necessary depends on the domain, intended use and users' requirements. If services are to be used to automate B2B and B2C models, they have to be described in a proper manner and meet specific business requirements.

The non-functional model for Web Services is still under development. Each of the already mentioned service description initiatives or standards like WSDL, UDDI, OWL-S, WSMO or SAWSDL treats non-functional properties in different ways. No non-functional properties can be expressed using WSDL. A list of non-functional parameters provided by UDDIs includes only some attributes such as e.g.: provider name, service name and category. In turn, OWL-S and WSMO take into account wider range of NFP (than e.g. UDDIs) including not only information on service providers, but also some performance related information like e.g. execution time etc.

C. Web Services Quality in literature

Web Services Quality attributes, quality characteristics or quality models have appeared in literature since beginning of 21st century. Majority of authors define quality of Web Services as Quality of Service using QoS as abbreviation. In many cases proposed quality attributes are analogical to those defined for low level telecommunication services or generic services definitions [27], [11]. Proposed quality characteristics and approaches to define Web Service quality that occur most often in literature are presented in table 1.

No	Re	Quality attributes / model
1.	[2]	execution price, latency time, average and maximum response time, robustness, availability, reliability, charging method, payment method Remark: Example attributes
2.	[44]	completeness, flexibility, interoperability, reliability, scalability, accuracy
3.	[5]	availability, throughput, latency, security, broadness of future set, price, location Remark: Example attributes
4.	[64], [63]	execution price, execution duration, reputation (end user experience), successful execution rate, availability Remark: Presenting different user QoS preferences giving different aggregation functions
5.	[35]	two main groups: 1. Generic: Execution price, execution duration, Reputation 2. Business: Transaction, Compensation rate, Penalty rate
6.	[26]	price, availability, reliability, reputation
7.	[41]	availability, security, response time, throughput
8.	[29]	availability, accessibility, integrity, performance, reliability, regulatory compliance, security
9.	[5], [51]	time, cost, fidelity, reliability Remark: Authors propose different functions to compute aggregated value of attributes
10.	[27]	response time, availability, performance, throughput Remark: Attributes regarded as the criteria for quality driven selection of Web Services
11.	[48]	four groups of attributes: 1. runtime: scalability, capacity, performance, reliability, availability, robustness/flexibility, exception handling, accuracy; 2. transaction: integrity (two phase commit) 3. configuration and cost: regulatory, supported standard, stability/change cycle, cost, completeness 4. security: authentication, authorization, confidentiality, accountability, non-repudiation
12.	[65]	main categories: response time, cost, reliability Remark: Each Web Service may have different QoS metrics to evaluate and describe its QoS

ISBN: 978-988-98671-8-8

No	Re	Quality attributes / model
13.	[53]	server performance, network performance, security, transaction, pricing and customer defined issues
14.	[60]	three groups of factors: 1. dynamic factors: service availability, network availability, execution duration 2. statistical factors: service reliability, network reliability, execution reliability, reputation 3. static factors: regulatory, security
15.	[45]	three groups of attributes: 1. general: performance (latency), performance (throughput), reliability, cost 2. internet service specific: availability, security, accessibility, regulatory 3. task specific: task specific
16.	[56]	performance, throughput, reliability, availability, trust
17.	[62]	response time, service cost, network delay, service availability
18.	[43]	three views with characteristics and sub-characteristics 1. user view: business value quality (cost / penalty, metering / billing, reputation, suitability), service level measurement quality (performance (response time, maximum throughput), stability (availability, successability, accessibility)) 2. interoperability view: conformability, interoperability, reliable messaging, message context, transaction 3. management & security view: management (manageable level, managed levels), security (data confidentiality, data integrity, user authentication, access control, non-repudiation, accessibility, audit trail, privacy)

Tab. 1, Search on literature for Web Service quality attributes and quality models

Analysis of different sets of quality attributes considered in literature as key quality indicators present different quality, business and functionality attributes regarded as QoS.

Problem with different presentation of quality attributes, followed by defined requirement for common Web Service quality understanding can be found in [29], but according to best knowledge of the authors, there was no publication in this area considering presented views nor Software Engineering quality modeling experience [8].

There are many approaches of discovery and selection of Web Services based on their quality information. Algorithms [31], [14], [34], [61], frameworks [48], [37], [27], [54], [43] and other ideas present different solutions for this task with respect to information reliability issues. Focus of this article is on how to define quality of Web Service, how to design Web Service for specified and implied quality needs, how to select relevant Web Service but leaving technical solutions aside.

D. Trust and verification

In this section the authors consider information reliability assuming existence of common Web Services quality model. Information and services available on an anonymous site (without signed agreement with provider) should be considered as trust unworthy. Even if third parties are considered in model providing additional information about service providers this information is another anonymous service and from this point should be considered as trust unworthy. This problem will be discussed with Web Service Quality evaluation issues in section IV.C.

Another problem for potential customer is related to business risks implicated by limited reliability of anonymous service provider. If user requires real time answers and the service call fails then even if alternative service is called time requirements are violated and user may suffer serious impact on his business. The same problem is when a result of Web Service provides incorrect data but user cannot verify it unless he uses alternative service with same query.

III. What is the meaning of "Quality of Service"

A. Layers and drivers of service quality

Quality considered as product attribute is difficult to trace and unambiguously define. Quality is commonly considered as complex measure. In ISO/IEC 25000 series SQuaRE[5] model (following previous ISO/IEC 9126) authors made distinction between software quality and overall quality for user (Software Quality in Use and Quality In Use). Such distinction along with internal and external software quality distinction emphasizes different layers and different responsibilities in providing demanded quality for users. Web Services can also be perceived as set of value (and quality) adding layers as shown on fig 1.

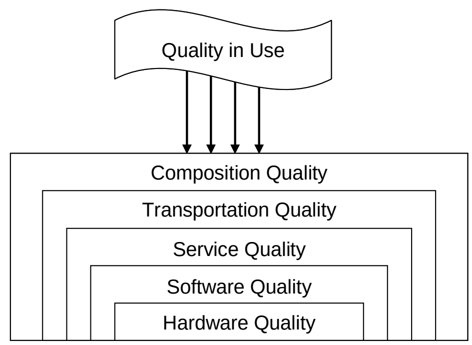

Fig. 1, Layers of quality regarding Web Services

Hardware term is used not only for computer hardware used to run Web Service, but whole infrastructure used to provide it (including LAN passive and active elements, UPSs, etc.). Software can be further divided into system software (operating system, database engine etc.) and software designed to provide service, although for purpose of this article such distinction is not needed. Authors distinct between internal and external software quality following SQuaRE model. Service layer includes all means and efforts taken by service provider in order to deliver service (eg. operational procedures). Transportation layer is term used for WAN layer description. Later on in this article it will be considered as part of service provider and broker responsibility. Composition layer is understood as responsibility of broker/service compositor making decisions on which service to use in certain context of user request.

Quality in use, or saying in another words *ability of system* to meet stated and implied user needs in specific context of use [17] should be defined exactly the same way as it is for tailored or commercial-of-the-shelf (COTS) software. User does not pay attention to internal architecture nor to the way the system is operating, but is interested in quality delivered

[5] SQuaRE model and ISO/IEC 25000 series are under development. According to ISO/IEC 25001, section 6 set of ISO/IEC 9126 and ISO/IEC 14598 should be used as SQuaRE model until new standard is completely published

ISBN: 978-988-98671-8-8

by system he is provided with.

Such approach allow defining user quality requirements independently from chosen implementation. It is also compatible with IEEE1061 standard defining two fundamental requirements for quality models:

- top-down decomposition – after quality requirements are defined model needs to support their decomposition to measurable attributes
- bottom-up measurement – basing on primitive quality indicators measures quality can be predicted

Quality lifecycle for Web Service is presented on fig 2. Transportation layer is omitted – this layer is addressed (divided) between attributes supervised by service provider and service broker.

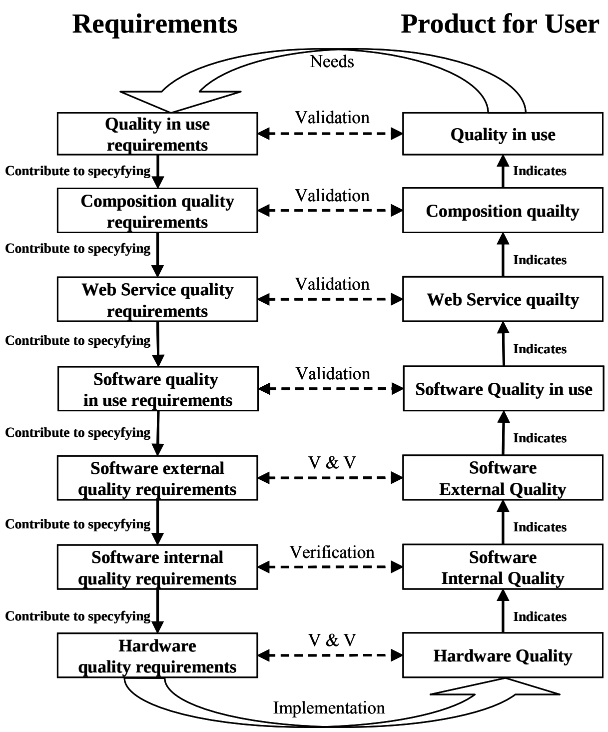

Fig. 2, Service based product quality lifecycle

Authors assume, that Web Services are mainly designed for composition and usage by other software systems, applications or service providers (underpinning service), by compositing software etc. This perspective follows [31], [38] regarding automatic services discovery and composition.

In terms and in compliance to SQuaRE model, Web Service quality can be defined as the extent to which a service used by specific users (brokers/compositors) meets their needs to achieve specific goals including quality in use delivered to software used by end-user.

B. Layers and responsibilities

1) Hardware

Hardware quality characteristics are not considered as main interest of the authors, although authors find it necessary to mention some important information regarding these characteristics.

Hardware quality characteristics, analogically to software quality characteristics, can be divided into two areas: static quality characteristics (static performance indicators, static throughput indicators, static reliability factors etc.) and dynamic quality characteristics (following architecture, interrelations etc.). Answering question which characteristics are important requires pointing out the role interested in quality measures of hardware. This role is obviously Service Provider.

Service Provider needs to deliver reliable, effective and correctly performing service [45]. This indicates needs for hardware quality in these areas. Another aspect of service quality and Service Provider tasks relate to scalability along with recovery issues (i.e. [48]), that is the reason for Service Provider to require replaceable, compliant with standards and expandable hardware.

Above set of main characteristics may not be exhaustive, but it covers quality requirements indicated by upper levels in order to deliver quality with Web Service.

2) Software internal and external characteristics

Software static and dynamic quality characteristics, described in ISO/IEC 9126-1[6] [21] as internal and external quality characteristics are described in six areas: functionality, reliability, usability, efficiency, maintainability and portability. Again if we consider interests and point of view of Service Provider we may see that all SQuaRE characteristics deliver important information about software for purpose of ensuring high quality service.

3) Software quality in use

Another important set of characteristics of software is described in ISO/IEC 9126-1 [21] as software quality in use. This set contains: effectiveness, productivity, safety and satisfaction. These characteristics may be considered as not crucial for Service Provider, because assumption made previously defines Web Service as provided for other software, automatic agents etc., but not end-users directly. It is then difficult to define "satisfaction" for automatic call of Web Service. If one looks back on quality definition he will find that it is defined as ability of a product to meet stated and implied user needs. These needs for Web Services are expressed through agents and other software representing user query or gathering data for a user, so quality delivered in Web Service influences those characteristics.

Service Provider has to consider Web Service Quality broader then only through perspective of technical output. If Web Service is considered as part of some business process [52] with their goals then quality in use characteristics may be defined and evaluated for the software.

4) Service provider

Having best quality of hardware and software is not sufficient for service provider to ensure high quality of Web Services he is providing. Important part of delivered Web Service Quality is an output from operation and maintenance procedures including procedures for exceptional situations, faults, failures and planned service shutdowns along with responsibility for providing reliable connection to backbone network allowing accessing Web Service.

[6] Will be replaced by ISO25010 [18]

5) Service broker/dispatcher

This responsibility is beyond the scope of Web Service Quality following assumptions made. However it is mentioned in order to have coherent and complete view on quality delivered to end-user for usage in quality lifecycle. Service broker [27] or a software composing process [64] for end-user is responsible for delivering high quality user interface and decomposition of user request into Web Services call in way, that compliant with its specification result from Web Service will be satisfying for user.

IV. Proposed Web Service Quality Model

A. Overview of the model

In proposed model authors define Quality of Web Service as ability of Web Service to provide specific users with specific service in defined context of use. Web Services quality model proposed in this article is based on SQuaRE model. Quality perspective consist of three views:

1. *Internal Web Service Quality* - capability of a set of static attributes of a Web Service design to satisfy stated and implied needs when the Web Service is used under specified conditions
2. *External Web Service Quality* - capability of a Web Service to provide responses satisfying stated and implied needs when the Web Service is used under specified conditions
3. *Web Service Quality In Use* - the extent to which the Web Service used by specific users meets their needs to achieve specific goals with effectiveness, productivity, safety and satisfaction in specific contexts of use

Web Service internal quality provides "white box" information about how the Web Service is designed, built and what is its ability to provide quality. External quality defines dynamic behavior of "black box" measuring quality of service understood as overall quality indicator for running service. Web Service Quality in Use summarizes quality in aspect of utility for specific user. This means that quality is measured basing on user-perspective and as at users site (for example transmission issues).

For such defined quality perspectives model also defines their relations to layers of responsibility presented in section III.A.

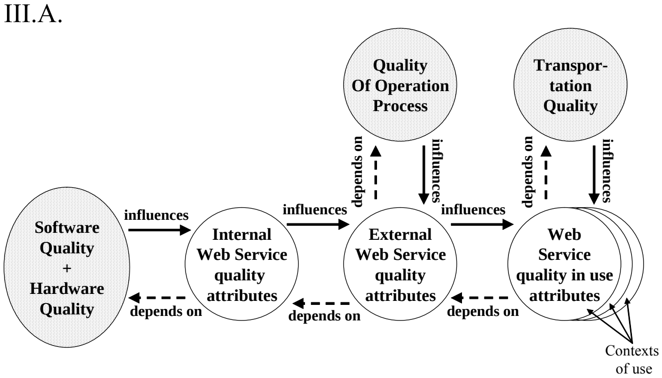

Fig. 3, Web Service quality views dependencies in lifecycle

Each view is then decomposed into three levels. Top level – quality characteristics are sets of main characteristics for perspective. Each characteristic have defined set of sub-characteristics. Model allows a user to define his own custom sub-sub-characteristics (same as SQuaRE model) but before this future is used one has to consider fact, that Web Services and their quality description are designed for automatic service discovery and selection – additional characteristics may be omitted by unknown Web Service selection engines. For each sub-characteristic there is a set of measures which can be used to express quality and may be aggregated to obtain overall sub-characteristics quality mark. Model does not require usage of each measure for each sub-characteristic. Service provider before making decision about quality measures he will provide for Web Service should consider user quality requirements and quality measures relevant for those requirements, as most probable in discovery and selection mechanisms used by users.

All of above make this model compliant to IEEE1061 requirements, because when quality requirements are defined they can be decomposed from those representing user/customer perspective, to external (operational) quality requirements, decomposed into requirements for internal quality requirements and also requirements for operational procedures. Characteristics may be decomposed into sub-characteristics and those to simple measures allowing in reverse direction to measure quality based on primitive measures.

B. Quality characteristics

1) Internal Web Service Quality

Internal Web Service Quality is measure purposed mainly for Service Provider. It is guidance for his preparations of Web Service. Acquisition or preparation of hardware, software, design of architecture and other a'priori parameters such as for example scalability (mentioned in [44], [48], [33]).

Main characteristics for this perspective are functionality, security, interoperability7, reliability, usability, efficiency, maintainability and portability.

In this article we do not focus on explicit listing of sub-characteristics nor primitive measures set. We refer to ISO/IEC 9126-2, -3 and ISO/IEC 25010 definitions, which are recognizable as quality standard for any kind of software, what means also software for delivering Web Services.

2) External Web Service Quality

External Web Service Quality is again quality perspective designed mainly for Service Provider. Main difference between External and Internal Web Service Quality is difference between static and dynamic attributes of Web Service. For example static reliability is based on probability of failure for components used for Web Service delivery. Dynamic reliability measure would be based on real observation of used components during their operation. Although we have traced additional quality influence from Operation Processes at Service Provider site we assume that set of eight characteristics is extensive enough to fully describe dynamic Web Service quality. These characteristics are the same as defined for Internal Web Service Quality: functionality,

[7] Security and interoperability are added to quality model following current works within SQuaRE project

ISBN: 978-988-98671-8-8

security, interoperability, reliability, usability, efficiency, maintainability and portability.

3) Web Service Quality in Use

Web Service Quality in Use is quality perspective of Web Service user. It was said above that such user acts through special software designed to discover and use Web Services. Quality perceived by user is also influenced by transportation layer and context of use. For example weather forecast service will be perceived differently by user intending to decide how to spend his weekend and by user planning flight schedule for region. Quality of data, availability of the service would have had different quality requirements for these two contexts of use.

Figure 2 presented quality lifecycle expressing root definitions for Web Service Quality in Use requirements. These are users-specified and implied needs which should be recognized and addressed by Service Provider with Web Service design.

Web Service Quality in Use is composed out of six characteristics: usability in use, context in use, safety in use, security in use, support in use and adaptability in use[8].

This article does not provide definitions of sub-characteristics nor measures assuming usage of SQuaRE definitions.

C. Objective evaluation of Web Service Quality

Evaluation of Web Service Quality is a complex issue. Assuming that Web Services are offered in the Internet one has to use roman trade rule *caveat emptor*[9]. In some approaches proposed in literature we may find concepts:

- a'priori [48] – one assumes that quality declarations are trust worthy
- based on some certifier [48] or RES agency [37] – one assumes that certification service is trust worthy
- based on service broker [27] [54] – one assumes that brokering service is trust worthy,
- a'posteriori, based on user feedback [48] – one assumes that other users opinions are trust worthy

Other approaches are: Q-components [40], QUEST [14], agent-based [31], OGSA [52], OASIS [43] etc. If we apply limited trust consequently then all of above approaches fail. From the other hand one may have trust, but it has to be based on the evidence [58]. Service Provider may publish false quality information, certifier, RES agency or broker may be paid to prefer some Service Providers, reference users may be virtual objects generating false feedbacks promoting certain service.

Is there a simple and universal solution? At this moment it is not known. The above issues may become the beginning of new service focused on pointing reliable information about public Web Services. To have trust in such service authors assume that this should be non-anonymous service, bought from carefully selected provider. Such an information provider should be specialized in discovery and evaluation of Web Services. This approach can also provide solution for different approaches to Web Services Quality Description.

Difference between thr above perspective and for example RES agency, certifier or broker-based approaches does not negate any of these models. Authors aim to show the problem, define its complexity and point out that breaking limited trust assumption requires careful selection of a provider and agreement between a customer and Web Services Evaluator to be sure, that quality information is reliable. These two requirements were not addressed in literature regarding provision of objective quality information.

There should be defined Web Services quality evaluation process in order to provide reliable information. In this point the Authors point out general framework of evaluation process described in ISO/IEC 25000 [17], ISO/IEC 14598-1 [16]. This framework describes evaluation process consisting of: establishing evaluation requirements, specifying of the evaluation, design the evaluation, execution of the evaluation and conclusion of the evaluation. Authors assume that evaluators should have user needs defined, they should define evaluation process to meet those needs and publish information relevant to those needs.

D. Quality description usage and on-line updates

Web Service Quality measures are purposed to be used in Web Services discovery and selection processes. In previous section the problem of trust in declared quality measure was discussed and in this section the authors consider another important issue regarding dynamic quality measures. These are temporal (momentum) values (we consider general problem similar to stated for reliability in [38]). For example if one knows that some service is generally very efficient, but in selection process he finds out that at that certain moment service is totally unstable or overloaded he probably is more interested in this temporal value then general statistic. This relation react with same strength for both negative and positive deviations from historical quality measures values. It is of course less likely that someone examines generally non-efficient Web Service so he might not discover that at certain moment it is perfectly efficient, but if somehow this information was delivered it could be used. In another words user is interested in the most probable parameters for his intended service call not in historical values.

Authors have proposed Web Service Quality model and assume that Web Service Quality information should be published in a way, allowing user to choose service accurate for his quality needs. Achieving this model of selection process (user states quality requirements according to quality model, Web Services discovery processes use user preferences and Web Services Quality measures organized quality model compliant to user quality model) Web Service Provider should be interested in publishing as many measures for each sub-characteristic of quality as possible.

Gathering reliable momentum information is important issue by itself. Considering various approaches and their foreseen costs one observes, that solution should be dependant from user requirements. If, as in example given previously, user requires fast, accurate and reliable information for activities ensuring safety and economic efficiency in aerospace industry then he will probably be willing to pay

[8] Quality characteristics differ from ISO9126-1 [21] Quality in use set following current works within SQuaRE project

[9] Latin: Let the buyer beware

for reliable information about momentum Web Services Quality measures. But if Web Service usage is aimed to give information for non important activities than reliable information about Web Services Quality momentum values would be nice to have, but certainly without any additional costs. It is then classic trade-off problem of information quality and its cost.

V. Summary

This article and quality model proposed inside summarizes current development of Web Services description and quality models. Referring to general software quality models the authors have proposed to new quality model based on Software Engineering and SQuaRE project achievements.

Justifying this approach authors point out that Web Services similarly to software are intangible objects purposed to be used by automata or other applications, having real impact on End User Experience, and overall quality delivered.

Designing new software solution ordered by customer analysts and developers gather quality requirements and describe them using quality model. The most common model for this moment is the ISO/IEC 9126-1:2001 [21] model and this model will be replaced with SQuaRE ISO/IEC 25010 model. User quality requirements will be decomposed into characteristics, sub-characteristics and measures and such description will be further contributing to establishment of external and internal software quality requirements. If one of components is to be replaced by Web Service then designer of the solution has to decompose quality requirements for this Web Service call as for traditional software module. This means that definition of quality requirements starts from the same set of requirements both for Web Service and software module. These requirements will define "user profile", which in case of Web Services will be used for discovery and selection of Web Service delivering desired quality to end user.

That is one of the reasons quality model for Web Services should be compatible with software quality model.

Further research should evaluate approaches to publicating quality characteristics for proposed model, evaluate accuracy of sub-characteristics and measures proposed for software quality model. Another important area is trust to quality description – this subject should also be investigated pointing characteristics vulnerable from dishonest Web Service providers with methods for overcoming this problem.

VI. References

[1] Abramowicz W., Godlewska A., Gwizdała J., Jakubowski T., Kaczmarek M., Kowalkiewicz M., Zyskowski D., A Survey of QoS Computation for Web Service Profiling, 18th International Conference on Computer Applications in Industry and Engineering, Honolulu, Hawai, USA, 2005
[2] Abramowicz W., Kaczmarek M., Zyskowski D., Duality in Web Services Reliability, Guadeloupe, French Caribbean, 2006
[3] Ankolekar A., Martin D., et al., OWL-S' Relationship to Selected Other Technologies., World Wide Web Consortium, 2004
[4] Aristotle, Metaphysics VIII 1043a 10-30
[5] Bilgin A. S., Singh M. P., A DAML-Based Repository for QoS-Aware Semantic Web Service Selection, pages 368-375, 2nd ICWS, San Diego, IEEE Press, 2004
[6] Boehm B., Brown J., Lipow M., MacCleod G., Characteristics of software quality, NY, American Elsevier, 1978
[7] Cardoso J., Sheth A., Miller J., Workflow Quality of Service, 4th International Conference on Information Integration and Web-based Applications and Services, Bandung, Indonesia, 2002
[8] Côté M-A, Suryn W., Georgiadou E., Software Quality Model Requirements for Software Quality Engineering, Software Quality Management & INSPIRE Conference (BSI), 2006
[9] Eenoo C., Hylooz O., Khan K., Addressing non-functional properties in software architecture using ADL, Paper presented at the 6th Australian Workshop on Software and Systems Architectures, 2005
[10] Encyclopedia Britannica
[11] Evans J., Filsfils C., Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice, Morgan Kaufmann, 2007
[12] Farrel J., Lausen H., Semantic Annotations for WSDL, DERI Innsbruck, 2006
[13] Farrell J., Laursen H., Semantic Annotations for WSDL and XML Schema, 2007
[14] Gu X., Nahrstedt K., Chang R.N., Ward C., QoS-Assured Service Composition in managed Service Overlay Networks, Distributed Computing Systems, 23rd International Conference on (ICDCS), Providence, RI, USA, 2003
[15] Horrocks I., DAML+OIL: a Description Logic for the Semantic Web, IEEE Data Engineering Bulletin 25 (1), 2002
[16] ISO/IEC 14598:1999 - JTC1/SC7, Information technology - Software product evaluation, International Standardization Organization, 1999
[17] ISO/IEC 25000:2005 – JTC1/SC7, Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE), International Standardization Organization, 2005
[18] ISO/IEC 25010-CD – JTC1/SC7, Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE), Internal ISO/IEC JTC1/SC7 Document, Currently at Commission Draft Stage, International Standardization Organization, 2007
[19] ISO 9001:1994, International Standardization Organization, 1994
[20] ISO/IEC 9126:1991 – JTC1/SC7, Information Technology – Software Product Quality, International Standardization Organization, 1991, International Standardization Organization, 1991
[21] ISO/IEC 9126-1:2001 - JTC1/SC7, Software engineering - Product quality, International Standardization Organization, 2001, International Standardization Organization, 2001
[22] ISO/IEC 9126-2:2003 - JTC1/SC7, Software engineering - Product quality, External Metrics, International Standardization Organization, 2003
[23] ISO/IEC 9126-3:2003 - JTC1/SC7, Software engineering - Product quality, Internal Metrics, International Standardization Organization, 2003
[24] ISO/IEC 9126-4:2004 - JTC1/SC7, Software engineering - Product quality, Quality in use metrics, International Standardization Organization, 2004
[25] Iverson W., Real World Web Services, O'Reilly, 2004
[26] J. O'Sullivan, D. Edmond, and A. ter Hofstede, What's in a Service?, Distributed and Parallel Databases, vol 12, 2002
[27] Kalepu S., Krishnaswamy S., Loke S-W., Reputation = f(User Ranking, Compliance, Verity), Proceedings of the IEEE In-

ternational Conference on Web Services, San Diego, California, 2004
[28] Khalaf R., Mukhi N., Tai S., Weerawarana S, The next step in Web Services, Communications of the ACM, vol 46, no 10, 2003
[29] Kim H. M., Sengupta A., Evermann J., MOQ: Web Services Ontologies for QOS and General Quality Evaluations, European Conference on Information Systems (ECIS), Regensburg, Germany, 2005
[30] Kitchenham S., Pfleeger, Software Quality: The Elisive Target, IEEE Software 13(1), 1996
[31] G, Kokash N., Dandrea V., Evaluating Quality of Web Services - A Risk-Driven Approach, Technical Report # DIT-06-099, 2006
[32] Lara R., Olmedilla D., Discovery and Contracting of Semantic Web Services, Paper presented at the Position Paper for the Workshop on Frameworks for Semantics in Web Services, Innsbruck, Austria, 2005
[33] Lee K., Jeon J., Lee W., Jeong S., Park S., QoS for Web Services Requirements and Possible Approaches, World Wide Web Consortium (W3C) note, 2003
[34] Lin M., Xie J., Guo H., Wang H., Solving Qos-Driven Web Service Dynamic Composition as Fuzzy Constraint Satisfaction, Internation Conference on e-Technology, e-Commerce, e-Service (EEE05), Hong Kong, 2005
[35] Liu Y., Ngu A. H. H., Zeng L. Z., QoS Computation and Policing in Dynamic Web Service Selection, 13th International World Wide Web Conference (WWW), New York, USA, 2004
[36] Martin D., et al., OWL-S: Semantic Markup for Web Services, 2004
[37] Maximilien E. M., Singh M. P., Reputation and Endorsement for Web Services, ACM SIGecom Exchanges, 2001
[38] Maximilien E. M., Singh M. P., Conceptual Model of Web Service Reputation, SIGMOD Record, Vol. 31, No. 4, pp. 36-41, 2002
[39] McCall J., Richards P., Walters G., Factors In software quality, Griffiths Air Force Base, NY, Rome Air Development Center Air Force Systems Command, 1977
[40] Menasce D. A., QoS-aware software components, IEEE Internet Computing, Vol. 8, No. 2, 2004
[41] Menasce D. A., QoS Issues in Web Services, IEEE Internet Computing, Vol. 6, No. 6, 2002
[42] OASIS, UDDI Version 3.0.2, UDDI Spec Technical Committee, 2004
[43] OASIS, Web Services Quality Model (WSQM) TC, http://www.oasis-open.org/committees/wsqm/charter.php, 2005
[44] Papaioannou I. V., Tsesmetzis D. T., Roussaki I. G., Anagnostou M. E., A QoS Ontology Language for Web-Services, 20th International Conference on Advanced Information Networking and Applications - Volume 1 (AINA'06), 2006
[45] Patel C., Supekar K., Lee Y., A QoS Oriented Framework for Adaptive Management of Web Service based Workflows, Database and Expert Systems Conference, Prague, Czech Republic, 2003
[46] Plummer D., Andrews W., The Hype Is Right: Web Services Will Deliver Immediate Benefits., http://www3.gartner.com/DisplayDocument?id=344028&ref=g_search, 2001
[47] Preist C., A conceptual architecture for Semantic Web Services, Paper presented at the International Semantic Web Conference, 2004
[48] Ran Sh., A Model for Web Services Discovery With QoS, ACM SIGecom Exchanges, Vol. 4, No.1, 2003
[49] Roman D., et al., Web Service Modeling Ontology, Applied Ontology Journal 1 (1), 2005
[50] Roman D., et al., WWW:WSMO, WSML and WSMX in a nutshell, Paper presented at the First Asian Semantic Web Conference, 2006
[51] Rosa N., Cunha P., Freire L., Justo G., Process NFL: A language for describing non-functional properties, Paper presented at the 35th Annual Hawaii International Conference, 2005
[52] Sheth A., Cardoso J., Miller J., Kochut K., Kang M., Qos for Service-oriented middleware, 6th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida, 2002
[53] Tian M., Gramm A., A Concept for QoS Integration in Web Services, 1st Web Services Quality Workshop (WQW 2003), Rome, Italy, 2003
[54] Tian M.,Gramm A., Ritter H., Schiller J., Efficient Selection and Monitoring of QoS-aware Web services with the WS-QoS Framework, IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004), Beijing, China, 2004
[55] Toma I., Non-functional properties in Web services, DERI Innsbruck, 2006
[56] Vu L., Hauswirth M., Aberer K., QoS-based Service Selection and Ranking with Trust and Reputation Management, International Conference on Cooperative Information Systems (CoopIS), Agia Napa, Cyprus, 2005
[57] W3C Organization, WSDL 2.0, 2007
[58] Wang Y., Singh M. P., Formal Trust Model for Multiagent Systems, Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI), 2007
[59] Wikipedia, http://en.wikipedia.org/wiki/Quality_of_Service
[60] Yoon S. H., Kim D. J., Han S. Y., WS-QDL containing static, dynamic, and statistical factors of WS quality, Proceedings of the IEEE International Conference on WS, 2004
[61] Yu T., Lin K. J., Service Selection Algorithms for Composing Complex Services with Multiple QoS Constraints, 3rd International Conference on Service-Oriented Computing, Amsterdam, the Netherlands (ICSOC-05), 2005
[62] Yu T., Lin K. J., Service Selection Algorithms for Web Services with End-toend QoS Constraints, Journal of Information Systems and E-Business Management, Volume 3, Number 2, 2005
[63] Zeng L., Benatallah B., Dumas M., Kalagnanam J., Sheng Q.Z, Quality Driven Web Services Composition, 12th International Conference on the World Wide Web, Budapest, 2003
[64] Zeng L., Benatallah B., Ngu A. H. H., Dumas M., Kalagnanam J., Chang H., QoS Aware Middleware for Web Services Composition, IEEE Transactions on Software Engineering Vol. 30, No. 5, 2004
[65] Zhou Ch., Chia L-T., Lee B-S., QoS-Aware and Federated Enhancement for UDDI, International Journal of Web Services Research, Vol. 1, No. 2, 2004