E.J.A.P.V.Shashikala, 2024, 12:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

Information Sharing Framework using Resource Oriented Architecture (REST) to Manage Quality of Service: A Comprehensive Review and Evaluation

E.J.A.P.V.Shashikala, W.M.J.I.Wijayanayake

Department of Industrial Management, University of Kelaniya,
Faculty of Science, Sri Lanka

Abstract- The research paper aims to identify the most important characteristics of Representational State Transfer (REST) that affect the quality and quality requirements of web services. A systematic literature review was conducted to identify the most frequently mentioned REST characteristics, which were then verified by industry experts in Sri Lanka. Based on the verified variables, a survey questionnaire was developed using a conceptual model. The survey was distributed to 33 developers and 34 consumers of RESTful web services, and their responses were analysed using Partial Least Square Structural Equation Modeling (PLS-SEM) in SmartPLS3 Version 3.1.6 to determine the linkages between REST characteristics and Quality of Service (QoS). Importance-Performance Matrix Analysis was executed to identify the REST characteristics with higher importance and performance towards QoS. Based on the findings, a framework was developed with REST characteristics that could improve QoS in the early development stages of RESTful web services. This framework is useful for service providers and developers to develop and implement high-quality RESTful web services that meet the QoS requirements of their customers.

Keywords- REpresentational State Transfer (REST), RESTful web services, Partial Least Square Structural Equation Modeling (PLS-SEM).

I. INTRODUCTION

REST has been evolving since the beginning of the Web and many developers are now implementing RESTful Web Services. However, this has led to an increase in the number of similar RESTful Web Services offering the same functionalities. Consumers may face difficulties in choosing the most suitable RESTful web service from a set of available options that provide the same functionality. This is because the quality of the services may vary, and there may be different criteria for evaluating the quality of each service. To address this issue, a model for managing the quality of RESTful web services needs to be defined by the service provider and developers. This framework should be defined by service providers and developers during the development and implementation stages to gain a competitive advantage.

While there have been numerous studies on the relationship between Service-Oriented Architecture (SOA), QoS, and web services, there is a lack of separate research papers specifically focusing on REST-based web services. Existing research has highlighted the importance of managing QoS for RESTful web services and extending the REST architecture to meet enterprise-level integration requirements.

Researchers such as Pautasso et al. (2008) have emphasized that RESTful web services are primarily suitable for ad-hoc integration and that security, reliable messaging, and transactions are key differentiators between RESTful and SOAP-based services. They argue that RESTful frameworks need to support these features to be viable for enterprise-level integration. Richardson and Ruby (2007) have demonstrated how security, reliable messaging, and transactions can be implemented in RESTful web services. However, further research is needed to define these concepts properly in the context of REST.

Webber et al. (2010) have noted that REST lacks support for security aspects, non-functional attributes, and composability, making it challenging to use as a core protocol for a comprehensive SOA infrastructure. Adamczyk et al. (2011) have stated that current RESTful web services primarily focus on providing functional interfaces and ignore QoS requirements. They propose the need for a language and mechanism to describe and incorporate QoS parameters in RESTful services, suggesting that a standard QoS description language could benefit from the work done in Semantic Web ontologies.

Zhang and Wijayanayake (2009) have identified the extension of REST to manage quality of services and develop a security framework for REST services as important future work. Overall, the advantages and limitations of REST architecture, particularly concerning QoS management, are still being assessed. Researchers have recognized the need to address the research problem of managing QoS in the context of the existing REST architectural style, and a comprehensive framework is required.

Although RESTful web services are popular among developers, they are not yet fully equipped to meet enterprise needs. The research problem lies in the lack of support for managing QoS requirements within the existing REST architectural style. QoS is crucial for adding value to service offerings. Therefore, it is necessary to conduct a study that develops an information sharing framework using REST architecture to effectively manage QoS.

II. LITERATURE REVIEW

1. ROA and REST Architecture

Resource-Oriented Architecture (ROA) encompasses three fundamental aspects: Resources, URIs, and Representations (Markey and Clynch, 2013). Resources, which refer to computer-stored objects, are accessed and stored using Uniform Resource Identifiers (URIs), while Representations define the state or format of the required resource. ROA serves as the foundation for REST, an architectural style introduced and defined by Roy T. Fielding in 2000. REST abstracts the architectural elements of distributed hypermedia systems, utilizing HTTP 1.1 and URIs as key components for its development. Initially serving as a data transfer transport model, REST has evolved to facilitate the construction of complex mashups and web services, offering an alternative to SOAP and other distributed computing specifications. Pautasso et al. (2008) highlight the strengths of REST, including its simplicity, lightweight nature, scalability, absence of additional configurations, and efficient discovery of web resources. These strengths have led to the widespread adoption of RESTful web services, ultimately improving the efficiency and performance of applications.

Despite its advantages, managing the Quality of Service (QoS) of RESTful web services remains a challenge. This paper suggests that frameworks and models used to maintain QoS in SOA and SOAP-based web services can also be applied to REST.

2. SOA and REST Architecture

SOA is a set of design guidelines or principles that enable the creation of loosely coupled services for efficient utilization. SOA improves the simplicity of machine-to-machine communication by utilizing a single endpoint. It employs a set of standards for publishing, discovering, and invoking services. SOA focuses on providing schema and message-based interactions, often utilizing XML as the common message format, with SOAP as the adopted standard (Markey and Clynch, 2013).

REST and SOA share some fundamental building blocks such as the client/server model and the concept of a layered system. However, they differ in certain constraints and components. While SOA lacks constraints like the uniform interface and virtual machine, these are fundamental to REST. Rotem-Gal-Oz (2013) describes two approaches to enriching SOA with REST: building a RESTful service and extending it to be an SOA service, or taking an SOA service and extending it to be RESTful.

Therefore, to enrich REST with QoS, it is plausible to consider applying SOA-related characteristics that impact QoS. To initiate the development of a REST framework with QoS, reviewing the literature on Web Service Quality Models (WSQMs) for SOA would be a valuable starting point.

3 Quality Models for web services

A software Quality model consists of Quality Characteristics that describe the quality of software. Web Service Quality Characteristics (WSQCs) or Web Service Quality Factors (WQCFs) are specific quality characteristics related to web services. These WSQCs are also known as Quality Attributes (QAs), which are features or characteristics that impact the quality of an item (IEEE Std. 610.12).

Quality Models group these QAs into a hierarchy of quality characteristics. When applied to a web service, the combination of WSQCs and attributes is referred to as QoS. QoS encompasses the features and characteristics of a product or service that contribute to its ability to meet stated or implied needs.

Quality Metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project. Software metrics are classified into three categories: product metrics, which describe the characteristics of the product; process metrics, which describe software development and maintenance; and project metrics, which describe project characteristics and execution.

In recent years, several Quality Models have been developed for SOA. According to Rotem-Gal-Oz (2013), it is possible and simpler to extend an SOA service to become a RESTful one. Therefore, it is logical to analyze the QoS factors that affect SOA architecture and derive a new Quality model based on those factors.

SQuRE based Web Services Quality Model

Abramovicz et al. (2009) proposed a web services quality model that is compatible with traditional software quality models. They emphasized that quality requirements for web services should align with those of traditional software. The model consists of three parts: external web service quality, internal web service quality, and web service quality in use. External and internal qualities focus on the service provider's perspective, while quality in use pertains to the consumer's viewpoint. Each view can be further decomposed into sub-characteristics and quality measures. The quality characteristics include functionality, security, interoperability, reliability, usability, efficiency, maintainability, and portability. Additionally, quality in use characteristics encompass usability in use, context in use, safety in use, security in use, support in use, and adaptability in use.

Deploying and Managing Web Services: Issues, Solutions, and Directions

Yu et al. (2006) identified key dimensions for evaluating web service technologies in terms of deploying and managing web services. These dimensions include interoperability, security and privacy, quality of web services (QoWS), and management. They address aspects such as cooperation among web services, security concerns, runtime and business quality parameters, and control and monitoring of web service qualities and usage.

QoS for Web Services: Requirements and Possible Approaches - W3C Working Group Note 25 November 2003

Web services have several key quality attributes that are essential for their effective functioning. Performance measures the speed and efficiency of completing service requests, while reliability ensures consistent and error-free service performance over a specified time period. Scalability determines the system's ability to handle increasing user demands, while capacity refers to the maximum number of simultaneous connections a service can support. Robustness focuses on the service's ability to function correctly despite invalid inputs, and exception handling deals with proper handling of unforeseen errors. Accuracy emphasizes minimal error generation, and integrity ensures data and transactional security. Accessibility measures the service's availability to serve client requests, and interoperability enables seamless interaction across different platforms. Security is crucial to protect against unauthorized access and includes authentication, authorization, confidentiality, accountability, traceability, and encryption. Lastly, network-related QoS requirements align application-level QoS parameters with network-level mechanisms for optimal performance.

Quality Attributes and Service-Oriented Architectures

O'Brien et.al.,(2004)

The SOA approach has both positive and negative impacts on various quality attributes. It enables interoperability and extensibility, but semantic interoperability and security standards are still developing. Reliability and availability rely on standards and negotiated agreements, while usability and performance may be affected. Scalability requires careful analysis to avoid negative impacts, and adaptability depends on proper management. Testability and auditability can be challenging, while modifiability is supported but requires careful consideration. Overall, the management and tradeoffs of these attributes are crucial for successful implementation of an SOA.

WSQM of OASIS 2005,2012

The OASIS WSQM consists of three parts: Quality Factors, Quality Activities, and Quality Associates. The focus of this study is on Quality Factors, which define the characteristics and attributes of quality for Web Services. The model is divided into three layers: Business Level Layer, Service Level Layer, and System Level Layer. The Business Level Layer considers the business value perceived by users, while the Service Level Layer focuses on the measurable performance quality of Web Services. The System Level Layer includes factors such as standards/interoperability, business processing, manageability, and security. However, criticisms of the model highlight its lack of consideration for domain-specific qualities and the need for a broader view of quality in the context of SOA. Additionally, the interrelationships between layers are not adequately discussed in the model (Kim et.al., 2005).

Table 1: Comparison between Quality Models

	A	B	C	D	E	F
Maintainability	*					
Correctness/Accuracy			*			
Reliability	*	*	*	*		
Efficiency	*					
Usability	*			*		
Portability	*					
Interoperability	*		*	*		
Modifiablity				*		
Functionality	*					
Performance			*	*		
Scalability			*	*	*	
Capacity			*			
Robustness			*			
Exception Handling/Usability			*			
Integrity		*	*			
Accessibility		*	*			
Availability		*	*	*		
Security	*		*	*	*	*
Network related			*			*
Extensibility				*		
Adaptability				*		
Testability				*		
Auditability				*		
Operability and Deploying/Installable				*		
Business Value					*	*
Service level meas.					*	*
Suitability for standards					*	*
Business Process					*	*
Manageability					*	*
Price		*				
Penalty		*				
Reputation		*				
Regulatory		*				
RTT		*				
Environment quality						*
Device quality						*
Domain specific						*
Dynamic capabilities						*
User Profile modeling						*
User Requirements modeling						*

A Semantic End-to-End QoS Model for Dynamic Service Oriented Environments

Mabrouk, 2009 developed a QoS model for Service Oriented Computing that focuses on dynamic service environments and end-to-end QoS. The model incorporates the WSQM and considers factors such as user mobility and context awareness. It consists of four ontologies: QoS Core, Infrastructure QoS, Service QoS, and User QoS. The QoS Core ontology provides general constructs for QoS description, while the Infrastructure QoS ontology focuses on the environment and underlying network infrastructure. The Service QoS ontology extends WSQM with factors supporting dynamic application services. The User QoS ontology addresses user concerns and specifies their QoS requirements. The model aims to provide modularity, flexibility, and manageability in QoS engineering.

4. A Comparison between Quality Models

According to research, there is a lack of consensus among researchers regarding fixed general quality attributes for web services. Terms like accuracy and correctness are used interchangeably by different researchers, and similar issues arise with terms like compliance and regulatory. Additionally, some sub-attributes, such as testability, interoperability, and understandability, are treated as separate quality characteristics in some models but are considered sub-characteristics in ISO 9126. For instance, testability is part of maintainability, understandability falls under usability, and interoperability is associated with functionality. The SQuRE-based Web Services Quality model by Abramovicz et al. (2009) is primarily based on ISO 9126, although other models like the OASIS proposals can also be used. However, the OASIS proposals are still in the form of a working draft and may not be widely known in the community. In summary, various Web Services quality models exist, each focusing on different perspectives and viewpoints of service quality. A holistic modeling approach, such as a view-based approach, is required to structure QoS models and establish connections between them, but further research is needed to develop such an approach (Lupeikiene et al., 2013).

III. METHODOLOGY

The main objective of this research is to come up with a framework using REST architecture to manage quality of web services. To do this need to identify the implication of REST characteristics and QoS. Therefore, required to select measurements for measure Web Service Quality and clearly identify main characteristics of REST.

1. Characteristics of REST

This study selected a limited number of quality characteristics (QCs) for SOA-based web services from the existing literature. The validity of these QCs was confirmed by gathering feedback from industry experts with practical experience. The identified QCs are categorized into seven layers: User Based Layer, REST Business Value Layer, REST Dynamic Capability Layer, REST Interoperability Layer, Business Processing Layer, REST Security and Privacy Layer, and Infrastructure Layer.

Table 2 REST User based layer characteristics

Factor	Description
Gap of User Requirements	User Requirements: The description of a constraint made by the user [Mabrouk et.al.,2010]
Types of Users' Mobility	Stationary Users: users with constrained movements that does not affect service provisioning. QoS-driven users: QoS-driven users are mostly stationary but they move when their perceived QoS level drops below an acceptable threshold. Mobile users: The mobile users are characterized by continuously moving positions [Mabrouk et.al.,2010]
Types of Users' Mobility	Low-load: users who are using the network for e-mailing and light Web browsing Medium-load: users who are using the network for intensive Web browsing, file downloading, audio streaming, and so on High-load- users: who make an intensive use of the network, such as video streaming [Mabrouk et.al.,2010]

REST User Based Layer

This separate layer could be affect the QoS since the provider and consumer of services could belong to different ownership domains so that there are many cases that a service cannot meet the consumer's service requirements in respect of service quality and content [Kim et.al., 2005].

REST Business Value Layer

Table 3 Business Value Layer

Factor	Description
Cost	Measures the units of money that a service requestor needs to pay to invoke service [Yu et.al., 2006]
Penalty or compensation	Measures the financial compensation for business losses due to non fulfilment of a contract or failure to meet promised quality. [Yu et.al., 2006]
Reputation	Measures the trustworthiness of a service operation based on user feedback. Users are prompted to rate service operations on a scale after using them. The reputation corresponds to the average of collected ratings. [Yu et.al., 2006]
Regulatory	Measures whether a Web service in conformance with the rules, the law, compliance with standards, and the established service level agreement. [Yu et.al., 2006]

REST Dynamic Capability Layer

Automation Support: Dynamic service environments focuses on fulfilling user tasks on-the-fly (i.e., at runtime) by dynamically locating and integrating available services. This is the ability of a service to support automated management.

Table 4 REST Dynamic Capability Layer

Factor	Description
Automation Support	Semantic Information Offerability - Ability of a service to provide semantic description of its functional and non-functional features. 32 [Mabrouk et.al.,2010]

Adaptation and Context Awareness: Services operating in highly dynamic environments need to continuously adapt themselves in order to react to changing conditions such as environmental conditions and user requirements with the context awareness.

Table 5 Adaptation and context awareness characteristics

Factor	Description
Adaptation	Ability of a service to adapt itself to changing conditions and to reconfigure itself accordingly[Mabrouk et.al.,2010]
Context Awareness	Ability of a service to gather, manage, use and disseminate context information [Mabrouk et.al.,2010]

Table 5 Management based characteristics

Factor	Description
Management based characteristics	Web service management refers to the control and monitoring of Web service qualities and usage. [Yu et.al., 2006]. Web service management mechanisms are highly coupled with the QoWS of a Web service.

Control: Typical control mechanisms include Web service transaction, Web service change management, and Web service optimization.

Monitoring Management: rates the behavior of Web services in delivering its functionalities in terms of each QoWS parameter.

REST Interoperability Layer

Web services are designed to bring together applications from geographically distributed and heterogeneous environments and provide interoperability among them (Yu et.al., 2006).

Table 6 Standard-Protocol based characteristics

Factor	Description
Standards	A specification or format for Web Service that has been approved by a recognized standardization organization or is accepted as a de facto standard by the industry [Yu et.al.,2006]
Conformability	Evaluate to which degree the

	standard technology of Web Services are conformed. Evaluation inspects whether a Web service implemented reflects the standard specifications [Kim et.al.,2005
Interoperation	Evaluate whether both conformable Web service systems are Interoperable and suggest profiles of Web services specifications. The profile suggests the guidelines of the applicable Web services standard [Kim et.al.,2005]

Business Processing layer

Performance indicators needed to represent functionality for collaboration among two or more web services.

Table 7 Business Logic Execution

Factor	Description
Reliable Messaging	Providing reliability functions which guarantee the level of reliability of messaging. Reliability functions are, Transmitting the message at least once (guaranteed delivery) Transmitting the message at most once (guaranteed duplicate elimination) Transmitting the message sequentially (guaranteed delivery order) [Kim et.al.,2005]
Transaction Processing Capability	Short-term Transaction: transaction which requires a service locked for a short period of time Long-term Transaction: transaction which requires a longer processing time or its resources cannot be locked exclusively during processing [Kim et.al.,2005]
Collaborability	Orchestration : relates to the execution of specific business processes Choreogrphy : Relates to describing externally observable interactions between web services.[Kim et.al.,2005]

Security & Privacy Layer

Security is an important issue for deploying Web services. Web services enable interoperation at the risk of letting outside intruders attack the internal applications and databases since they open up the network to give access to outside users to these resources [Yu et.al.,2006].

Table 8 Security & Privacy Layer

Factor	Description
Authentication	Verify a claimed identity [Yu et.al.,2006]
Authorization	Check whether a user is authorized to perform a requested action[Yu et.al.,2006]
Confidentiality	Ensure that information is disclosed only to authorized recipients[Yu et.al.,2006]
Integrity	protection of the information from being tampered [Yu et.al.,2006]
Privacy	During service interactions personal data or business secrets must protect [Yu et.al.,2006]

Infrastructure Layer

Network based factors: Web services are usually invoked through networks, so the network performance critically affects the overall web service quality. Addresses the communication infrastructure in dynamic service environments.

Table 9 Network based factors

Factor	Description
Network Performance	Bandwith - Rate of data transfer [Mabrouk et.al.,2010]
	Latency - Total time taken to deliver a message [Mabrouk et.al.,2010]
	Loss - Rate of message units lost during the delivery of a message
	Jitter - variation in Latency [Mabrouk et.al.,2010]

Device based factors: Addresses hardware devices hosting application services (e.g., Server) or supporting end-users (e.g., PDA, SmartPhone, PC)

Table 10 Device based factors

Factor	Description
Device Performance	Common device capabilities; CPU Memory of the device Storage Capacity of the device Power Consumption of the device [Mabrouk et.al.,2010

Environment based factors: intrinsically related to the environment where users and services exist. Such environments are populated with networked services supporting a wide variety of user applications. The quality of these environments can be assessed by evaluating their degree of support to user applications.

Table 11 Environment based factors

Factor	Description
Sustainability	Environment's ability to sustain employed services if they fail [Mabrouk et.al.,2010]
Scalability	Ability of the environment to support a large number of active users and to handle their requests in a satisfying manner [Mabrouk et.al.,2010]

Measure the Web Service Quality

In the literature, it has been observed that there are numerous Web Service Quality Models (WSQMs) available for Service-Oriented Architecture (SOA) based Web Services. However, there is a lack of separate models specifically designed for RESTful Web Services. Therefore, it is common practice to consider the quality measurements used for measuring SOA based Web Services as applicable to RESTful Web Services as well. These quality measurements are often specified in various literature sources, including ISO9126, OASIS, and W3C Web Service Requirements.

Table 12 Selected Quality Requirements of Web Services

Factor	Description
Response Time	Measures the expected delay between the moment when a service operation is initiated and the time the operation sends the results. Response time = Response Completion Time – User Request Time [Kim et.al.,2005]
Throughput	Measures the max number of services that a platform providing web services can process for a unit time, Maximum throughput = max complete requests / unit time [Kim et.al.,2005]
Reliability	Measures the ability of a service operation to be executed within the maximum expected time frame.
Availability	Measures the probability that the service operation is operating at any given moment and is available to perform its function on behalf of its users. In another word, a high available service operation is one that will most likely be working at a given instant time. Availability = 1 – (Down Time/Unit Time) [Kim et.al.,2005]
Accessibility	Accessibility in Web services refers to the capability of a service operation to successfully serve a request. It is often measured as the ratio of acknowledgments received to the total number of request messages. High accessibility can be achieved through scalability, which ensures consistent service despite varying request volumes. [Kim et al., 2005]
Accuracy or Successibility or Correctness	Measures the extent to which Web services yield successful results over request messages. Successibility = number of response messages / number of request messages [Kim et.al.,2005]
Integrity	Transaction correctness in Web services refers to the proper execution and completion of a sequence of activities treated as a single unit of work. If a transaction fails to complete, all changes made are rolled back. [Kim et al., 2005]

Using identified REST characteristics and web service quality requirements the conceptual framework was developed as shown in Fig. 2. The Conceptual framework was validated using expert knowledge in the industry through a face to face interview.

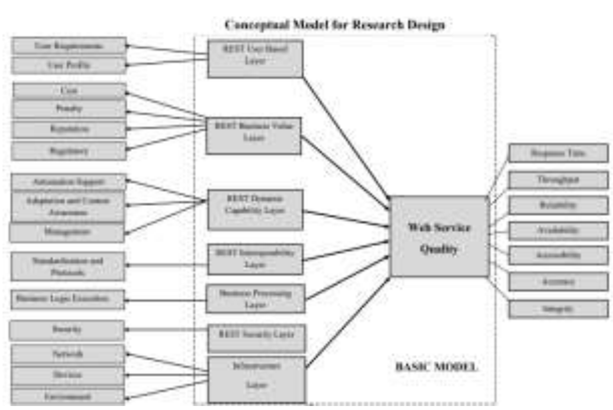

Fig. 2 Conceptual Model of Research Design

Data Collection

Out of three main research approaches Qualitative, Quantitative and Mix (Teddie and Tashakkori 2009) in this research, used the quantitative approach. The questionnaire used in the research study was designed to gather qualitative data, which means that it aimed to collect information about the users' personal experiences and knowledge related to the quality of web services. That needs to be converted into a numerical value by giving weights to each response and justify each choice in order to analyse.

In this study, the researchers used two questionnaires to collect data. Questionnaires are a common technique for gathering data in research studies, and many previous studies have used this method as well. The online questionnaires were designed to gather information from two different perspectives: developers and consumers of web services. The researchers created separate online Likert scale-based questionnaires for each perspective.

The researchers selected companies that have developed RESTful web services based on an online survey conducted initially. From these companies, they selected 50 developers to participate in the study. After conducting a pilot survey, the researchers distributed the first online questionnaire among the selected developers. However, only 33 developers responded back. Based on the responses from the first questionnaire, the researchers selected 40 RESTful web service consumers and distributed the same questionnaire to them. Only 34 of these consumers responded back.

The questionnaires used in the study measured the same variables from two different perspectives; developers and consumers, in order to improve the reliability of the results. The variables are listed in Table 1 and the researchers gathered data on these variables from both developers and consumers to gain a more comprehensive understanding of the impact of REST QAs on web service quality and performance.

Table 13 Variables used to measure QoS

Layer	Factor/variable	Italic
User Based Layer (Mabrouk et al. 2010)	Gap of Requirements	UB01
	User Mobility	UB02
	Traffic of Users	UB03
Business Value Layer (Yu et al. 2006)	Cost	BV01
	Penalty	BV02
	Reputation	BV03
	Regulatory	BV04
REST Dynamic Capability and Change management Layer (Mabrouk et al. 2010)	Automation Support	DC01
	Adaptation	DC02
	Context Awareness	DC03
	Change Management	DC04
REST Interoperability Layer (Yu et al. 2006)	Standards	I01
	Conformability	I02
	Interoperability	I03
Business Processing Layer (Kim et.al. 2005)	Reliable Messaging	BPR01
	Transaction Processing Capability	BPR02
	Collaborate-ability	BPR03
REST Security Layer (Yu et al. 2006)	Authentication	S01
	Authorization	S02
	Confidentiality	S03
	Integrity	S04
	Privacy	S05
Infrastructure Layer (Mabrouk et al. 2010)	Network Performance	IF01
	Device Performance	IF02
	Sustainability	IF03
	Scalability	IF04

Data Analysis

To conduct a quantitative research, selecting the appropriate statistical techniques is crucial. According to (Hair et al. 2011) for a quantitative research there are two data analytical methods; univariate and multivariate. In this research, use the multivariate analysis since need to measure multiple dependent variables simultaneously.

Structural Equation Modeling (SEM) is a multivariate analysis method that enables researchers to investigate complex relationships among multiple independent and dependent variables simultaneously. In this research all the dependent/independent variable relationships postulated and use the SEM technique as the data analysis technique. Previous literature has suggested that PLS is advantageous because it requires fewer assumptions about the distribution of the data and works well with smaller sample sizes (Qureshi 2009). This is due to its component-based nature. Kwong and Wong (2013), PLS-SEM becomes a good alternative to Covariance Based-SEM when the following situations are encountered,

Sample size is small.
Applications have little available theory.
Predictive accuracy is paramount.
Correct model specification cannot be ensured.

Since the sample size of this research is relatively small and distributional assumptions cannot be made, the PLS approach was chosen to find the linkages between REST characteristics and QoS of web services and to validate the conceptual model. PLS data analysis was done using the software package SmartPLS 3 (Version: 3.1.6).

The research study evaluated two components of SEM the measurement model and the structural model. The researchers assessed the reflective measurement model for reliability and validity, which involved testing the consistency and accuracy of the variables measured using composite reliability and Cronbach α coefficients. These coefficients needed to be greater than 0.7 to ensure internal consistency reliability of the variables. The variables were tested separately for the consumer and developer samples, as well as for the whole sample.

Table 14 Reliability of the model

Sample	Construct	Cronbach's α	Composite Reliability
Whole	REST characteristics	0.886	0.897
	QoS	0.796	0.852
Consumer	REST characteristics	0.907	0.917
	QoS	0.802	0.856
Developer	REST characteristics	0.858	0.883
	QoS	0.796	0.852

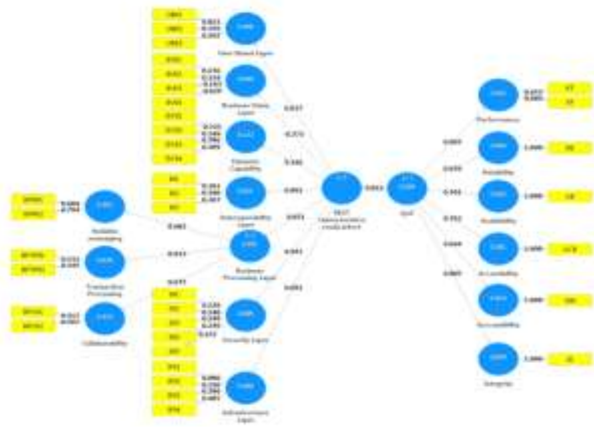

Fig.3 PLS values for Whole sample

According to (Hair et al. 2011) the boundary value of the indicator loadings is 0.4. Analysed each indicator loadings from REST characteristics and QoS constructs towards each indicator and as well as constructs established for seven layers specified earlier. Several indicator loadings were less than 0.4 for whole, consumer and for developer samples. Cross loadings also analysed for whole sample as in Fig.4, developer and consumer samples and several indicators associated with latent construct not higher than its loadings with all the remaining constructs were recognized.

Then the Structural model evaluated, according to (Hair et al. 2011) the primary evaluation criteria for the structural model are the R^2 measures and the level and significance of the path coefficients. R^2 measures resulted in between substantial and moderate level for three samples.

Table 15 R^2 square values for samples

Sample	R^2 Dependent variable - QoS
Whole	0.659
Consumer	0.673
Developer	0.681

PLS SEM does not presume that the data are normally distributed, hence cannot use that parametric significance tests used in regression analyses. Therefore, in this research use the bootstrap procedure 500 iterations selected. 0.05 significance level, the hypothesized paths of the constructs are considered to be significant (t-value is greater than 1.96). Since, all t-values from REST characteristics to QoS higher than 1.96, path coefficients in the inner model are statistically significant.

Table 4 Total effects [Mean, STDEV, T- Values]

Sample	Original Sample (O)	Sample Mean (M)	Standard Error (STERR)	T Statistics (|O/STERR|)
Whole	0.820	0.778	0.171	4.802
Consumer	0.825	0.758	0.180	4.572
Developer	0.812	0.790	0.109	7.465

Indicators with the lessen t values than 1.96 were recognized and all the indicators could not satisfy the outer loadings, cross loadings and t-values for whole, developer and consumer samples were listed. all the indicators which led to have minus or less values many times were removed and those are Mobility and Traffic of User Based Layer, Price, Penalty and Reputation of REST Business Value Layer, Automation Support, Adaptation and Change Management of REST Dynamic Capability Layer, Reliability and Fault Tolerance, Orchestration and Choreography related to Business Processing Layer, Network Performance of Device Performance of Infrastructure Layer, Response Time related to measure the Quality requirements for web services.

The validated model was again tested using PLS. All the outer loadings are greater than 0.4 (Hair et al. 2011) and are therefore significant. For the whole sample PLS gives a strong relationship between REST characteristics and the QoS of the organization.

By listing the t values according to whole sample in the descending order we can find out the indicators with higher corporation towards the REST characteristics and QoS. For REST characteristics: Interoperability, Standardization, Privacy, Authorization, Regulatory and Confidentiality. For QoS; Integrity, Accessibility, Availability, Throughput, Success-ability and Reliability. Fig. 4.

Importance – Performance Matrix

According to (Ahamad 2014), Importance-performance matrix analysis (IPMA) is useful in extending the findings of the basic PLS-SEM using the latent variables score. The goal to conduct IPMA is the evaluation of key target constructs in QoS. Once, we execute the IPMA analysis in SmartPLS it resulted the importance of each independent indicator towards the QoS.

IPMA of QoS reveals, as shown in Fig.6 that Integrity has the highest importance and performance for establishing QoS. Identification of Gap of Requirements has higher importance and relevance to keep the QoS but the performance is very poor.

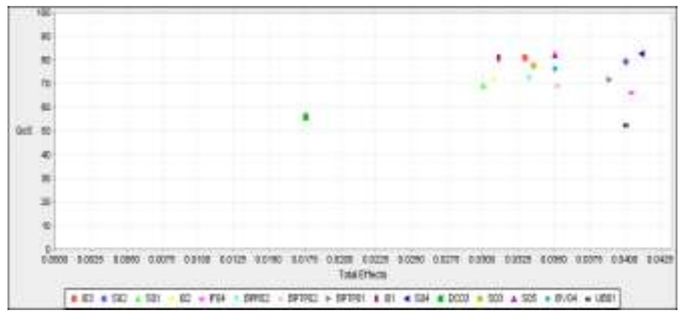

Fig. 4 IMPA Analysis for QoS

IV. DISCUSSION

Qualitative research approach was undertaken and the statistical tools used for data analysis was Partial Least Squares, which is a structural equation modeling technique. After analyzing the results obtained from first validation process, revalidated model has presented by the researcher. Some variables had to remove from the model as there was no support from those variables to the model. It may be due to irrelevancy of the variables, the particular variable might have been measured already or the less support given by these variables to REST characteristics and QoS. The validated model is giving all the variables that impact towards the model.

Most significant variables of REST characteristics are Interoperability, Standardization, Privacy, Authorization, Regulatory, Confidentiality, Transaction Processing Capability-Short Term, Integrity, Conformability and Authentication. Most significant variables which affect QoS are Integrity, Accessibility, Availability, Throughput, Success-ability and Reliability.

According to IPMA, importance of each exogenous variable towards the QoS, from high to low can be listed as , Integrity, Scalability, Gap of Requirements, Authorization, Transaction Processing –Short, Transaction Processing –Long, Regulatory, Privacy, Confidentiality, Reliability Functions, Interoperability, Standardization, Conformability, Authentication and Context Awareness.

Performance of each exogenous variable towards the QoS, from high to low can be listed as, Integrity, Privacy, Interoperability, Standardization, Authorization, Confidentiality, Regulatory, Reliability Functions, Conformability, Transaction Processing-Short, Authentication, Transaction Processing-Long, Scalability, Context Awareness and Gap of Requirements.

Using these two lists related to importance and performances towards QoS, determine what are the factors really important towards the QoS are those factors perform well to improve the QoS.

V. CONCLUSION

Through the overall analysis derived that REST characteristics have a higher impact on QoS. Attention towards the factors could affect the QoS is very much important to improve the QoS. This will develop high quality web service from the scratch in development and implementation stages. Major contribution of this research is to come up with a framework which can be focused by the developers of RESTful web services and giving directions for future research.

Declarations

Availability of Data and Materials

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing Interests

Not Applicable

Funding

No funding was received for conducting this study.

Authors' contributions

All authors equally contributed to the study conception and design.

Acknowledgement

The authors very much appreciate the valuable comments and suggestions given by the editor and the reviewers.

Abbreviations

REST: Representational State Transfer
PLS-SEM: Partial Least Square Structural Equation Modeling
QoS: Quality of Service
SOA: Service Oriented Architecture
QCs: Quality Characteristics
QAs: Quality Attributes
W3C: World Wide Web Consortium
SEM: Structural Equation Modeling

REFERENCES

1. Adamczyk,P., Smith,P.H., Johnson,R.E., and Hafiz,M. (2011) REST: From Research to Practice.
2. Ahamad,S. (2014) The Importance-Performance Matrix Analysis In Partial Least Square Structural Equation Modeling (Pls-Sem) With Smartpls 2.0 M3, International Journal of Mathematical Research.
3. Hair,J.F., Ringle, C.M. and Sarstedt,.M., PLS-SEM: Indeed a silver bullet, Journal of Marketing Theory and Practice, vol. 19, no. 2 (spring 2011), pp. 139–151.
4. IEEE Std. 610.12, Glossary of Software Engineering Terminology.
5. Kim,E. and Lee,Y., Quality Model for Web Services, OASIS, 2005
6. Kwong, K. and Wong, K., Partial Least Squares Structural Equation Modeling (PLS-SEM) Techniques Using SmartPLS, Marketing Bulletin, 2013, 24, Technical Note 1, 2013
7. Mabrouk,N.B., Georgantas,N. and Issarny,V., A semantic End-to-End QoS Model f or Dyanamic

Service Oriented Environments, France, March 2010

8. Oz, R.G. A.,RESTful SOA- SOA Patterns by Arnon, 2013.
9. Pautasso, C., Zimmermann, O., and Leymann, F. (2008) RESTful Web Services vs. "Big" Web Services: Making the Right Architectural Decision, International World Wide Web Conference Committee.
10. Qureshi,.I, Assessing Between-Group Differences In 144 Information Systems Research: A Comparison Of Covariance- And Component-Based Sem 2009.
11. Richardson, L. And Ruby, S., RESTful Web Services, 2007.
12. Teddie.C. and Tashakkori, A., Foundations of Mixed Methods Research: Integrating Quantitative and Qualitative Approached in the Social and Behavioural Science, Sage Publication.Inc , 2009.
13. Webber, Jim., Parastatidis, S. and Robinson, I., REST in Practice, Hypermedia and Systems Architecture, O'Reilly, 2010.
14. w3.org (2002) Web Services Description Requirements Available at: https://www.w3.org/TR/2002/WD-ws-desc-reqs-20021028/
15. Yu, Q., Liu, X., Bouguettaya, A., and Medjahed, B., Deploying and managing Web services: issues, solutions, and directions, 2006.
16. Zhang,T.I. and Wijayanayake, J.I.(2009) A Framework for REST Service Registry, Faculty of Business and Law, Victoria University, Footscray Park Campus, Australia.