Insights into API Management Quality Characterization, Measurement and Evaluation

Eder dos Santos[1][2]

[1]Instituto de Tecnología Aplicada – Universidad Nacional de la Patagonia Austral (UNPA)

[2]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Río Gallegos – Santa Cruz – Argentina

esantos@uarg.unpa.edu.ar

Abstract. *APIs have become a cornerstone in software ecosystems: organizations have been increasingly connecting software applications to share complex digital assets. However, managing APIs is non-trivial, and companies have been struggling with different quality-related issues in managing their APIs. The main goal of this work is to create a framework that allows characterizing, assessing, and improving API Management and API governance, particularly focusing on API Gateways as the core elements of API Management ecosystems.*

1. Introdcution

In recent years, information systems have increasingly adopted Everything-as-a-Service (XaaS) models [Duan et al. 2015], which are primarily built on microservice architectures. These architectures provide organizations with the flexibility to break down their systems into highly scalable services for more efficient management [Gamez-Diaz et al. 2017]. Within this framework, Application Programming Interfaces (APIs) [De 2023] have become key elements in the development of modern software products [Iyer and Subramaniam 2015]. APIs have extended to enterprise information systems [Yamamoto et al. 2018] and form the foundation of today's digital business, where industries share their commercial assets through new business models. This shift has given rise to what is known as the API Economy, where APIs serve as essential enablers for monetization, collaboration, and integration across diverse platforms and industries [Tan et al. 2016].

In general terms, an API is defined as a software-to-software interface through which applications communicate with each other over a network, without user interaction. Software developers access APIs as interfaces to code libraries, frameworks, or data sources, enabling them to avoid low-level programming tasks and/or accelerate development [De 2023]. Currently, several authors [De 2023] [Preibisch 2018] suggest that the term API refers to Web APIs built using REST principles, and the expressions API, Web API, REST API, and RESTful API are considered synonymous. Therefore, the term API will be used throughout this document.

APIs are managed through API management platforms, whose central component is the API Gateway. An API management platform provides the essential capabilities for creating, analyzing, and managing APIs in a secure and scalable environment. The services and functions of the API Gateway enable the inspection of all incoming and outgoing traffic and provide different levels of support for fulfilling a wide range of functional and non-functional requirements related to APIs, such as access control, network-level security, encryption, validation, transformation and routing of messages, resource availability, logging, caching, threat protection, monetization, documentation, developer registration, and tracking of products consuming the API, lifecycle management, auditing, and operational and performance analysis [De 2023] [Preibisch 2018] [Gamez-Diaz et al. 2017]. As a result, the API Gateway facilitates the operation of API, contributes to the separation of concerns and decoupling, facilitates service integration and scaling, and accelerates development.

The growing adoption of APIs has led to the proliferation of various API Gateway products. The global API Management market grew by 13.7% and reached 3.3 billion USD in 2023[1]. Additionally, current industry "state of the market" reports[2][3][4] indicate that there are 65+ API Gateway offerings available nowadays. In such a competitive landscape, it is essential to understand how software quality can be defined, and what these products provide to satisfy the stakeholders' requirements. Nonetheless, several authors agree that, in general, the quality of API is largely overlooked, and it is simply assumed that they are sufficiently "good", "secure", and/or "stable" for adoption [Bermbach 2017], and there is scarce evidence of research in the API Management field.

The remainder of this work is structured as follows: section 2 describes related work; section 3 presents the research questions and objectives that the work aims to achieve; section 4 outlines the research methodology used; section 6 details the proposed solution under development; and finally, section 6 concludes with the final considerations and limitations.

2. Related Work

2.1. Software Product Quality

There is a wide range of standards applicable to various areas of knowledge in Software Engineering, with a common objective related to quality. In general, software quality can be viewed from two perspectives: the first is based on the ISO 9001 standard, which focuses on quality policy across the entire organization. The terminology of this standard may be unfamiliar to software professionals, and quality management auditors may not be well-versed in software engineering vocabulary. The second perspective seeks to decompose software quality into a set of desired characteristics of a software product and select measures and evaluation methods to determine if the desired level of those characteristics has been achieved. Over time, various quality models have been developed from this standpoint, including FURPS, Boehm, Dromey, and McCall, among others [Nistala et al. 2019].

The terminology for quality attributes of software products differs from one model to another; each model has a different number of hierarchical levels and a distinct set of characteristics [Nistala et al. 2019]. A milestone in the definition of quality standards was the publication of the ISO/IEC 9126:1991 standard, which was later replaced by

[1] https://www.gartner.com/en/documents/5594559

[2] https://www.gartner.com/en/documents/4829031

[3] https://www.postman.com/state-of-api/

[4] https://apilandscape.apiscene.io/

the ISO/IEC 9126:2001 and ISO/IEC 14598:1999 standards. Currently, the international ISO/IEC 25000:2014 standards series, known as SQuaRE (Software Product Quality Requirements and Evaluation), unifies and replaces the ISO/IEC 9126 and ISO/IEC 14598 series. SQuaRE is considered the current de facto standard for software product quality and consists of a series of 27 standards that address aspects related to quality management, requirements, software and data quality models, as well as measurement and evaluation processes for quality.

2.2. API management quality

With the growing proliferation of APIs and the subsequent recognition of their importance, several technical challenges have been identified [Espinha et al. 2015]. In line with this, [Yamamoto et al. 2018] argues that the quality of APIs has become a highly relevant issue, asserting that research in Software Engineering focused on Web APIs is urgent. In this regard, various studies indicate that formal evaluation methods are rarely used in practice for the evaluation and selection of API products, leading to the need for empirical research on the evaluation and selection practices adopted in the industry for both open-source products [Ayala et al. 2013] [Sarrab and Rehman 2014] and commercial “off-the-shelf” products [Ayala et al. 2011], while also addressing international standards related to software product quality.

Nevertheless, studies addressing the quality of API Gateways are scarce. [Ofoeda et al. 2019] highlights the limited research on the topic: out of a total of 104 papers analyzed, 20 main research areas were identified, and none of these areas were explicitly related to the terms “software quality” or “API Gateway”. Finally, only four papers were related to the research area of “API Management”.

3. Research Questions and Goals

The main goal of this doctoral research is to develop an comprehensive, platform-neutral API Gateway quality framework that can be used to create open tools for the characterization, measurement, evaluation, and improvement API Management software quality.

In the search for effective strategies for both providers and consumers of API Management solutions, the following research questions (RQ) have been proposed for exploration:

RQ1: *What quality characteristics can be identified and how can they be specified in a formal basis?* This question aims to explore what dimensions and properties are essential to evaluate API Gateway software quality and how they can be articulated in a precise, formalized manner. This might involve defining these characteristics using a systematic framework, a set of metrics, or through formal models to ensure clarity and avoid ambiguity in quality assessment.

RQ2: *What relevance do these characteristics have from the perspective of the different stakeholders?* This question seeks to assess the degree of importance of each quality characteristic from various perspectives. By examining the priorities of different stakeholders, businesses or organizations can make more informed decisions about which aspects of quality to emphasize, ensuring that they strike a balance between customer satisfaction, technical feasibility, cost-efficiency, and compliance with regulations.

RQ3: *How can these characteristics be measured and assessed?* The goal of this RQ is to establish reliable, repeatable, and valid ways of assessing quality that allow for consistent evaluations over time, across different versions, or across different API Gateway products. This ensures that quality can be tracked, improved, and communicated effectively across all stages of development and use.

RQ4: *How are these quality characteristics related to API Management practices and capabilities?* This RQ aims to explore the relationship between software quality and API management practices and capabilities, providing insights for developers and organizations to align their practices with desired quality outcomes, particularly in the context of increasingly complex and distributed architectures.

Based on this, the following specific objectives are outlined: a. To study and analyze quality models, as well as measurement and evaluation methods for quality applied to API Management in general and API Gateways in particular. b. To propose a theoretical and conceptual metamodel for the standardized specification, measurement, and evaluation of quality characteristics in line with the capabilities of API Gateways. c. To identify the quality characteristics applicable to API Gateways and specify them based on the proposed metamodel. d. To determine the relevance of the identified quality characteristics from the perspective of different stakeholders. e. To design and/or implement tools and frameworks that support the proposed approach.

To provide a consistent and integrated solution, the ISO/IEC 25000 international standards series will be adopted as a reference, and the Goal / Question / Metric (GQM) approach [Basili et al. 1994] will be applied as the evaluation framework.

4. Research Methods and Activities

This research combines different empirical and non-empirical research strategies: solution-seeking strategies [Stol and Fitzgerald 2018] will be adopted for the generation of artifacts and for the adoption of models and tools, while knowledge-seeking strategies [Stol and Fitzgerald 2018] will be used for the analysis, validation, and quality evaluation of the results. As a general framework, the Design Science Research (DSR) framework [Peffers et al. 2007] will be applied. The activities and methods to be applied in each case are outlined below.

1. A review of the literature and other sources of information related to metamodels, models, and quality methods applied to API in general and to API Gateways in particular. To achieve this, a systematic mapping study (SMS) following the guidelines proposed by [Petersen et al. 2015] [Shaw 2003] has been planned.
2. Identification of the quality characteristics applicable to API Gateways and their specification based on the proposed metamodel. To achieve this objective, the GQM paradigm, which is conceptually compatible with SQuaRE and specific to the software development domain, will be adopted.
3. Evaluation of the relevance of the identified quality characteristics from the perspective of different stakeholders. For the preliminary definition, sampling studies such as questionnaires and interviews [Stol and Fitzgerald 2018] will be conducted.
4. Development of a metamodel for the standardized specification, measurement, and evaluation of quality characteristics. To achieve this, the theoretical formu-

lation method proposed by [Ralph 2018] will be applied, and the design will be supported by the principles of the Model Driven Architecture (MDA) approach. This framework allows the establishment of analysis and evaluation models for API Gateway products; as a result, a set of relevant quality attributes and associated metrics will be obtained.

5. Design of a quality model as a set of instances of the specified metamodel.

The validation of the described activities will be carried out using different methods, namely: a. Expert judgment studies, as described in [Stol and Fitzgerald 2018]. b. Development of artifacts to support the proposed approach, following the DSR principles. c. Design of case studies to evaluate a set of at least 10 real API Gateways and statistically analyze the results, as proposed in [Kitchenham et al. 1995] [Pfleeger 1995]. d. Communication of the obtained results through scientific publications and the doctoral thesis.

5. Results

Our research has made significant strides in advancing the understanding of quality traits in API Management, supported by empirical studies and key publications. Four of these publications notably highlight these contributions.

Initially, an SMS was conducted [dos Santos and Casas 2024] following the rigorous framework presented in [Petersen et al. 2015]. The primary objective of this work was to examine the definitions, measurement methods, and evaluation criteria used to assess API Management software quality, with a particular emphasis on methodological perspectives [Shaw 2003], API Management capabilities, and software quality attributes. From an initial set of 1205 papers, 21 identified studies conducted between 2000 and 2023 were identified, highlighting the novelty of the research topic.

The findings indicated that current research landscape is heterogeneous and primarily focuses on technical analysis and specific solution development. Key quality characteristics included performance efficiency, reliability, functional suitability, and security, which align with various API Management capabilities and current industry trends. The results also suggested that research is predominantly informal or lacking in theory, highlighting the need of defining a tailored quality metamodel to define, measure, and assess API Management software quality. Finally, threats to validity were addressed, and research gaps to encompass future research were identified.

As the second main contribution of this research, [dos Santos and Casas 2023] presented a methodical exploration of how practitioners perceive the quality traits associated with API management requirements in alignment with the quality characteristics outlined in ISO/IEC 25010. A comprehensive survey targeted professionals in various roles in Rio Gallegos City, Argentina, including developers, system administrators, and functional analysts. The outcomes of this study revealed that Functional Suitability, Performance Efficiency, and Security emerged as the most prominent quality characteristics for API management, corroborating the findings obtained in [dos Santos and Casas 2024]. These insights are critical for future research and practical applications in API management platforms, guiding developers to enhance functionality and security in their platforms. The study also provided a data-driven approach to assist platform providers in refining their offerings based on user needs, ultimately aiming to create more efficient and reliable API Management solutions.

The third advancement of this research introduced API-MQM, a tailored quality metamodel that builds on the ISO/IEC 25010 standard and integrates the GQM framework [dos Santos and Casas 2025], following the principles of Model-Driven Architecture (MDA). The proposed metamodel encompasses API Management capabilities and practices as key quality requirements, along with software quality characteristics, measurement and assessment, and tools that support the evaluation framework. The proposal was validated through both theoretical and empirical methods [Cachero et al. 2007] [Shaw 2003].

The last contribution of the current research is under peer review. The primary artifact is a catalog of 59 metrics, compiled from 68 industry-leading API Gateway products. These metrics were categorized into distinct groups based on clear definitions found in recent literature [De 2023] [Beyer et al. 2016] [Gregg 2013]. The DSR approach was adopted to design the catalog, and a quantitative analysis method was proposed and systematically performed. This study revealed that most metrics focus on specific API Management capabilities, namely latency and response time, API performance, error capturing, and traffic monitoring. Other features such as caching, saturation, resource utilization, and system health were scarcely addressed by the examined products, which highlights potential gaps in the current research landscape. The proposed artifacts in this study provide an objective foundation for a deeper understanding of API Management software quality and lay the groundwork for future investigation.

6. Concluding Remarks and Future Work

With the widespread adoption of APIs in modern software ecosystems, the need for robust API management practices has become increasingly relevant. Therefore, a comprehensive understanding of software product quality is essential for addressing API Management activities, capabilities, threats, and challenges, ultimately enhancing the success and trustworthiness of API-based systems and services.

Although software quality has become paramount in API Management, the field remains under-explored in scientific research. The primary objective of this paper was to present the ongoing work of the doctoral thesis, which aims to define a framework for characterizing, measuring, evaluating, and improving the software product quality of API Gateways. This framework is based on three key pillars: the ISO/IEC 25000 standards as a formal reference model, the GQM approach as a guiding framework, and API Management capabilities as quality requirements.

This paper also outlined the research objectives, methodology, and planned activities of the research. Notable activities include published work such as the systematic mapping of the literature, the identification and weighting of quality characteristics from the perspective of industry professionals, the definition of a metamodel as a flexible and conceptually coherent framework, and work under review, specifically the development of a preliminary catalog of metrics derived from a comprehensive analysis of API Gateway products offered in the industry.

Finally, it is anticipated that by the end of this research, a set of artifacts will be available to support the processes of characterizing, measuring, evaluating, and improving the quality of API gateways. These artifacts will include a quality model based on the API-MQM metamodel, evaluation methods, tools, and case studies focused on both experimental environments and real-world applications.

Referências

Ayala, C., Franch, X., Conradi, R., Li, J., and Cruzes, D. (2013). Developing software with open source software components. *Finding Source Code on the Web for Remix and Reuse*, pages 167–186.

Ayala, C., Hauge, Ø., Conradi, R., Franch, X., and Li, J. (2011). Selection of third party software in off-the-shelf-based software development—an interview study with industrial practitioners. *Journal of Systems and Software*, 84(4):620–637.

Basili, V. R., Caldiera, G., and Rombach, H. D. (1994). The goal question metric approach. In *Encyclopedia of software engineering*.

Bermbach, D. (2017). Quality of cloud services: Expect the unexpected. *IEEE Internet Computing*, 21(1):68–72.

Beyer, B., Jones, C., Petoff, J., and Murphy, N. (2016). *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly, Sebastopol, CA.

Cachero, C., Calero, C., and Poels, G. (2007). Metamodeling the quality of the web development process' intermediate artifacts. In *Web Engineering: 7th International Conference, ICWE 2007 Como, Italy, July 16-20, 2007 Proceedings 7*, pages 74–89. Springer.

De, B. (2023). *API Management: An Architect's Guide to Developing and Managing APIs for Your Organization*. Apress Berkeley, CA.

dos Santos, E. and Casas, S. (2023). API management and SQuaRE: A comprehensive overview from the practitioners' standpoint. In *Argentine Congress of Computer Science*, pages 137–150. Springer.

dos Santos, E. and Casas, S. (2024). Unveiling quality in API management: A systematic mapping study. In *2024 L Latin American Computer Conference (CLEI)*, pages 1–10, Buenos Aires, Argentina. IEEE.

dos Santos, E. and Casas, S. (2025). AN API MANAGEMENT SOFTWARE QUALITY METAMODEL BASED ON SQUARE AND GQM. In *Engenharias e soluções: Ciência e tecnologia para o desenvolvimento humano*, pages 249–266. Atena Editora, 1 edition.

Duan, Y., Fu, G., Zhou, N., Sun, X., Narendra, N. C., and Hu, B. (2015). Everything as a service (XaaS) on the cloud: Origins, current and future trends. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 621–628, New York City, NY, USA. IEEE.

Espinha, T., Zaidman, A., and Gross, H.-G. (2015). Web API growing pains: Loosely coupled yet strongly tied. *Journal of Systems and Software*, 100:27–43.

Gamez-Diaz, A., Fernandez, P., and Ruiz-Cortes, A. (2017). An analysis of RESTful APIs offerings in the industry. In Maximilien, M., Vallecillo, A., Wang, J., and Oriol, M., editors, *Service-Oriented Computing*, volume 10601, pages 589–604. Springer International Publishing. Series Title: Lecture Notes in Computer Science.

Gregg, B. (2013). Thinking methodically about performance. *Commun. ACM*, 56(2):45–51.

Iyer, B. and Subramaniam, M. (2015). The strategic value of APIs. *Harvard Business Review*, 1(7):2015.

Kitchenham, B., Pickard, L., and Pfleeger, S. L. (1995). Case studies for method and tool evaluation. *IEEE software*, 12(4):52–62.

Nistala, P., Nori, K. V., and Reddy, R. (2019). Software quality models: A systematic mapping study. In *2019 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, pages 125–134. IEEE.

Ofoeda, J., Boateng, R., and Effah, J. (2019). Application Programming Interface (API) Research: A Review of the Past to Inform the Future. *International Journal of Enterprise Information Systems*, 15(3):76–95.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77.

Petersen, K., Vakkalanka, S., and Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18.

Pfleeger, S. L. (1995). Experimental design and analysis in software engineering. *Annals of Software Engineering*, 1(1):219–253.

Preibisch, S. (2018). *API development: a practical guide for business implementation success*. Springer Science+Business Media, New York, NY.

Ralph, P. (2018). Toward methodological guidelines for process theories and taxonomies in software engineering. *IEEE Transactions on Software Engineering*, 45(7):712–735.

Sarrab, M. and Rehman, O. M. H. (2014). Empirical study of open source software selection for adoption, based on software quality characteristics. *Advances in Engineering Software*, 69:1–11.

Shaw, M. (2003). Writing good software engineering research papers. In *25th International Conference on Software Engineering, 2003. Proceedings.*, pages 726–736, Portland, OR, USA. IEEE.

Stol, K.-J. and Fitzgerald, B. (2018). The abc of software engineering research. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 27(3):1–51.

Tan, W., Fan, Y., Ghoneim, A., Hossain, M. A., and Dustdar, S. (2016). From the service-oriented architecture to the web API economy. *IEEE Internet Computing*, 20(4):64–68.

Yamamoto, R., Ohashi, K., Fukuyori, M., Kimura, K., Sekiguchi, A., Umekawa, R., Uehara, T., and Aoyama, M. (2018). A quality model and its quantitative evaluation method for web APIs. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, pages 598–607.