Building a Catalogue of ISO/IEC 25010 Quality Measures Applied in an Industrial Context

Mariana Falco1* and Gabriela Robiolo2
1LIDTUA/CONICET, Engineering School, Austral University. Mariano Acosta 1611, Pilar, Buenos Aires, Argentina
2LIDTUA, Engineering School, Austral University. Mariano Acosta 1611, Pilar, Buenos Aires, Argentina
*Email: mfalco@austral.edu.ar

Abstract. Measuring quality is extremely important while developing a software product, but there is a lack of knowledge of which are the ISO/IEC 25010 quality measures that are currently being used in the industry. Through a literature analysis of 27 studies, the present article introduces a catalogue of 269 quality measures applied in an industrial setting, which is formed by a subset of 81 metrics defined by GQM approach, a subset of 81 specific metrics for quality characteristics, and a subset of 86 metrics defined through ISO/IEC 25023 and 21 through IT-CISQ. In conclusion, it can be said that GQM is the most widely used method, regardless of standards. Likewise, Maintainability, Performance Efficiency and Usability are the quality characteristics that have shown the highest degree of interest because they are more receptive to the end user.

1. Introduction
Several changes have occurred in the past years that had a huge impact on all aspects of software development, delivery, and consumption, mostly the cloud computing era [1]. Also, sprints are on a regular basis within the software development lifecycle, and nowadays, companies deploy new code into production weekly or even daily [2]. With these fast and more frequent release times, it is extremely important to properly measure software quality due to the impact of the speed of releases. Measuring quality is a practice that takes time, and quality measures can vary from one software to the other, from one domain to the other, and it can exist those from academia and those applied to an industrial setting.

Within the Systems and Software Quality Requirements and Evaluation (SQuaRE) family of standards, it is possible to find ISO/IEC 25010:2011 [3] categorizes the quality model of a software product into a set of quality characteristics and sub-characteristics. In this context, it is necessary to be able to identify the metrics applied to the industry because those are the ones that can be easily reproducible, as well as tested through the application. Our main goal is to understand to what extent quality measures are being applied in companies, while our research question is the following: Which are the quality measures defined in ISO/IEC 25010 that are currently being used in the industry?

In this context, we have developed a Product Quality Evaluation Method (PQEM) [4] is a five-step semi-automated method, whose main goal is to analyze, study, measure and assess the quality level of the different iterations within a software product. And also, we are currently developing the QualityTracker Tool as an automated software tool to help businesses and industries in the process of measuring software quality, based on the PQEM method.
In order to answer the research question, a set of scientific databases (Science Direct, ACM, Springer, and IEEE) were queried with a mix of search strings (quality attributes requirements, quality attributes, Goal Question Metric, GQM, software quality and each quality characteristic within the ISO/IEC 25010 [3]). From the search, were included studies that were related to software measurement and definition of quality measures applied to the industry. The latter allowed us to build a catalogue of quality measures applied on an industrial setting, which its introduction remains as the main goal of the present study. Even though there are some studies that describe a measures catalogue within software products, we haven’t found a catalogue of quality measures that have been applied to the industry. The article is structured as follows: Section 2 describes a brief related work, while Section 3 describes PQEM and the Quality Tracker Tool. Section 4 introduces the catalogue and its main characteristics, and finally, Section 5 presents the conclusions and future work.

2. Related Work
A recent survey conducted with industrial practitioners and experts revealed some concerns about code quality and the need for instruments to support them [5]. Regarding existing catalogues, we have found some relevant ones such as a catalogue for thresholds for object-oriented software metrics [6], another one that contains metrics to identify infrastructure-as-code properties [7] as well as another for mobile elicitation techniques [8]. Considering that quality needs to be evaluated from different levels of detail [9], it is important to discover the existing quality measures applied to industry. We didn’t find a catalogue of software quality measures specifically applied to an industrial setting, which its existence will represent a way that developers and quality managers can develop, maintain, manage and evolve a software product.

3. PQEM and Quality Tracker tool
A previous effort by the authors was the definition of PQEM [4], which allows the practitioners as well as managers to perform the elicitation, measurement, and synthesis of the quality level from a software product. Also, we begin developing an automated software tool called QualityTracker to help business and industries in the process of measuring software quality. While different automation tools exist, not all of them address a comprehensive analysis of all quality characteristics or associated measures, nor do they calculate quality level in a multidimensional aggregated number. Within this tool, it was thought to define and incorporate a catalogue with a set of ready-to-apply quality measures which have been applied in the industry.

3.1. Product Quality Evaluation Method (PQEM)
Product Quality Evaluation Method (PQEM) [4] is a five-step method, whose main goal is to analyze, study, measure and assess the quality level of the different iterations within a software product, and that produces a single value between 0 and 1 as the final outcome that represents the product quality level. PQEM is structured by the standard ISO/IEC 25010 [3], the standard ISO/IEC 25023 [10], the Goal-Question-Metric (GQM) approach [11], and the extension of testing coverage to calculate the coverage of each quality characteristic.

The five steps of PQEM are the following: 1) Product setup, 2) Elicitation of Quality Attributes Requirements (QARs), 2.a) Select quality characteristics and sub-characteristics, 2.b) Specify quality attributes requirements, 2.c) Define metrics of each quality attribute requirement, 2.d) Define acceptance criteria of each quality attribute requirement, 3) Measure and test each quality attribute requirements, 4) Collect and synthesize results, 5) Assessment of the product quality level, and 6) Collect measurement. It is worth mentioning that within step 2), each quality requirement needs at least one measure to be evaluated. It is desirable to have a list of measures at the time of selecting a metric for a question posed. The quality analyst will ask what has been used previously.

3.2. QualityTracker Tool
In this context, our main goal is to develop an automated software tool to help business and industries in the process of measuring software quality. While different automation tools exist, not all of them address a comprehensive analysis of all quality characteristics or associated measures (whether they are defined by ISO/IEC 25023 [10] as well as new ones without the standard), nor do they calculate quality level in a multidimensional aggregated number [4]. Therefore, this automated tool will serve as a means of interface with other existing tools such as SonarQube, in order to achieve a complete tool that will allow decisions to be made in each iteration of a software product in different industries and businesses.

4. Presenting the Catalogue

Aligned to the goal embedded with the QualityTracker Tool, another goal is the construction of the catalogue is to define a set of quality measures which have been applied to an industrial setting, and therefore share it with the research community as well as project leaders, project managers, quality leaders, and developers in order to show and understand what have been applied and what is missing. Table 1 describes the amount of metrics per each quality characteristic, divided between those that were obtained through the application of the GQM approach [11] extracted from 19 papers, those obtained from a paper that approaches the IT-CISQ standard [12], those metrics defined to achieve the quality characteristic itself but without any specific approach and extracted from 4 papers, and finally those from the ISO/IEC 25023 [10].

Regarding Table 1, the 27 analyzed studies let us to obtained 269 metrics applied in an industrial setting, from which: a) 81 metrics defined by the GQM approach [11] for Usability, Reliability, Security, Maintainability, Safety and Evolvability; b) 21 metrics defined through IT-CISQ [12]; c) a set of 81 metrics defined for Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Maintainability and Portability, and d) 86 metrics defined while following ISO/IEC 25023 [10]. As shown, the quality characteristics with the higher amount of defined metrics are Maintainability, Performance Efficiency, and Usability [13].

4.1. Quality Measures Captured with GQM

Within the literature review carried out, we have found the following contributions which are built on the foundations of the Goal-Question-Metric approach [11], a decision support framework for metrics selection (DSFMS) [14], SYMBIOSIS provides security-specific templates and methodological elements to contextualise security metrics and align them with business objectives [15], to measure and improve testability on the basis of the Goal Question Metric Approach [16], to define metrics for Evolvability defects [17], to assure the quality of technical documentation [18], to measure the nature and types of hazards associated with systems of systems [19], to define metrics for usability within the mobile industry [20], to define low level measurements [21], for automated metrics based evaluation of the software architecture [22], a hybrid framework to evaluate the quality of cloud computing based e-business [23], for industrial measurement programs [24], for software process management [25], V-GQM: a feedback approach to validation of a GQM study [26], for bottom-up design of robot systems [27], for product-focused process improvement [28], an extension of GQM [29], within the Fixed Switching of Nokia [30], for a process framework for customising software quality models [31], and for assessing and evaluating domain candidates for product line engineering [32].

4.2. Quality Measures from ISO/IEC 25023

ISO/IEC 25023 [10] belongs to the family of the Systems and Software Quality Requirements and Evaluation (SQuaRE) family of standards, and it defines quality measures for quantitatively evaluating systems and software product quality in terms of characteristics and sub-characteristics defined in ISO/IEC 25010 [3]. As stand on the standard description, the main users of this standard are people carrying out quality requirements specification as part of development, quality management, supply, acquisition, and maintenance. Regarding the literature review, Schramme and Macías [1] performed the analysis and measurement of internal usability metrics through code annotations using this standard, as
well as Tsuda et al. [17] who have defined a comprehensive software quality evaluation framework and benchmark based on SQuaRE.

**Table 1. Quality characteristics and amount of metrics obtained.**

| Quality characteristics | Sub characteristics | Metrics with GQM | Metrics with IT-CISQ | Metrics from ISO/IEC 25023 |
|-------------------------|---------------------|------------------|----------------------|---------------------------|
| Functional Suitability  |                     | 0                | 0                    | 1                         | 4                         |
| Performance Efficiency  |                     |                  | 0                    | 0                         | 37                        | 12                        |
| Compatibility           | Co-existence, Interoperability | 0 | 0 | 5 | 4 |
| Usability               | Appropriateness recognisability, Learnability, Operability, User error protection, User interface aesthetics, Accessibility | 14 | 0 | 12 | 22 |
| Reliability             | Maturity, Availability, Fault-tolerance, Recoverability | 17 | 0 | 9 | 11 |
|                         | Safety              | 6                | 0                    | 0                         | 0                         |
| Security                | Confidentiality, Integrity, Non-repudiation, Accountability, Authenticity | 12 | 0 | 0 | 11 |
|                         | Usable security     | 0                | 0                    | 4                         | 0                         |
| Maintainability         | Modularity, Reusability, Analyzability, Modifiability, Testability | 30 | 21 | 9 | 13 |
|                         | Evolvability        | 9                | 0                    | 0                         | 0                         |
| Portability             | Adaptability, Installability, Replaceability | 0 | 0 | 4 | 9 |
| **Total**               |                     | 81               | 21                   | 81                        | 86                        |

4.3. **Metrics with IT-CISQ**
The Consortium for Information & Software Quality (CISQ) is a not-for-profit IT leadership group that develops standards for automating software measurement from source code, and this includes measures of software size, structural quality, technical debt, and related metrics [12]. CISQ was co-founded by the Object Management Group, and Software Engineering Institute at Carnegie Mellon University. It contains a set of standards to automate software measurement including software sizing (automated functions points, automated enhancement points), code quality (security, reliability, performance efficiency, maintainability), and technical debt.

Within the literature analysis, we have found an article which was oriented to Maintainability (as shown in Table 1) [33], and these metrics are included within the catalogue apart from the fact of its applicability in an industrial setting. In this context, the standard defined a set of 29 common software weaknesses (CWE) to mitigate in the source code. But based on the particular conditions defined by
Plösch and others [33] they only included 21 of them. It is worth mentioning that within the standard, each CWE possesses the description available online.

4.4. Metrics
We have found some studies [34-37] which described or defined a set of quality measures in order to achieve specific quality characteristics, following Table 1: Functional Suitability (1), Performance Efficiency (37), Compatibility (5), Usability (12), Reliability (9), Maintainability (9), Portability (4), and Usable Security (4).

5. Discussion
Although a quality measurement model is determined by the objectives of each business, our objective was to collect the metrics already applied in the industry to help focus QA analysts on aspects that the community considers relevant. Along these lines, we were surprised by the small number of articles that show the application of quality metrics in the industry. But the focus on Maintainability, Performance Efficiency and Usability could be observed, and if we ask ourselves it could be justified that Maintainability is necessary to monitor the systems during their production, which is when they have a longer and more expensive life; Performance Efficiency because it is a highly sensitive measure for the end user, and finally, Usability because it is extremely important to make user interfaces friendly.

We also observed that the integration proposed in standard ISO/IEC 25010 of the functional and non-functional aspects has not been promoted, an aspect that we have taken into account when proposing the PQEM [4], since it was possible to synthesize in a number the measurement of all the characteristics. Also, the lesser use of Compatibility and Portability was evidenced, which although they are key aspects in the context of current technological changes, we understand that they have less sensitivity for the end user. GQM [11] as the dominant methodology, outside the scope of the standards, since 81 metrics of them were obtained through this approach. There is a benefit of aligning with the business objectives, clarifying in its definition why and what I am measuring, giving greater semantics to the measurement results.

With respect to the threats to validity, one topic is the range of years chosen to carry out the search (1994-2020) but considering that only those applied to the industry were selected, it can be considered as a decrease in threat. On the other hand, the scientific bases chosen represent an interesting and complete corpus of studies, but perhaps the non-inclusion of another base such as PubMed leads to the loss of relevant articles in specific domains. Likewise, if we might perform the search in December or in the first months of 2021, we might be able to add new studies that are currently being reviewed or even in press.

6. Conclusions
The present article focused on understanding which metrics are being applied in the industry, and so a literature review was carried out from 27 articles. The analysis of each article allowed us to build a catalogue of quality measures that have been applied to an industrial setting. We have identified 269 quality measures in total, and this have been defined to achieve a set of quality characteristics: Functional Suitability, Performance Efficiency, Compatibility, Usability, Reliability, Security, Maintainability, Portability, Safety, Evolvability, and Usable Security. Also, we have identified the lack of full publications that include the analysis of the different measures monitored by existing tools such as SonarQube, which are currently used by over 100000 organizations. This contribution would be necessary to understand the uses and benefits to the quality measurement process. As a future work, we will analyze this last issue as well as conduct a case study in order to validate the QualityTracker tool.

Acknowledgments
This work is supported by a research grant from Universidad Austral.

References
[1] Schramme M and Macías J A 2019 Analysis and measurement of internal usability metrics through code annotations defined in ISO/IEC 25010. A catalogue of thresholds for object-oriented software metrics. *Journal of Systems and Software* 170: 110726

[2] Falco M and Robiolo G 2019 A Unique Value that Synthesizes the Quality Level of a Product A framework for software usability and user experience measurement in mobile industry. In 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME) IEEE pp. 580-589

[3] Filó T G, Bigonha M and Ferreira K 2015 A catalog of thresholds for object-oriented software metrics. *Proc. of the 1st SOFTENG* pp. 29-35

[4] Palomba F, Ghafari M, and Nierstrasz O 2019 Towards a catalogue of Mobile Elicitation Techniques. In *International Conference on Product-Focused Software Process Improvement* (Springer, Cham) pp. 281-288

[5] Mordal K, Anquetil N, and Guerriero M, ISO/IEC 25010, (s.f) https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

[6] Iqbal M I 2013 A decision support framework for metrics selection in goal-based measurement programs: GQM-DSFMS. *Journal of Systems and Software*, 86(12), 3091-3108

[7] Tangan J, Rönnkö K and Gencel C 2013 A framework for software usability and user experience measurement in mobile industry. In 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (IEEE) pp. 156-164

[8] Beer A, and Felderer M 2018. Measuring and improving testability of system requirements in an industrial context by applying the goal question metric approach. In *Proceedings of the 5th International Workshop on Requirements Engineering and Testing* pp. 25-32

[9] Ducasse S 2017 A catalog of thresholds for object-oriented software metrics. *Proc. of the 1st SOFTENG* pp. 29-35

[10] Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE), ISO/IEC 25023:2016, https://www.iso.org/standard/35747.html

[11] Caldiera V R B G and Rombach H D 1994 The goal question metric approach defined in ISO/IEC 25010. A catalog of thresholds for object-oriented software metrics. *Proc. of the 1st SOFTENG* pp. 29-35

[12] Consortium for Information and Software Quality (CISQ), Measuring code quality, https://www.it-cisq.org/standards/code-quality-standards/

[13] Tan J, Rönnkö K and Gencel C 2013 A framework for software usability and user experience measurement in mobile industry. In 2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement (IEEE) pp. 156-164

[14] Gencel C, Petersen K, Mughal A A and Iqbal M I 2013. A decision support framework for metrics selection in goal-based measurement programs: GQM-DSFMS. *Journal of Systems and Software*, 86(12), 3091-3108

[15] Philippou E, Frey S and Rashid A 2020. Contextualising and aligning security metrics and business objectives: A GQM-based methodology. *Computers & Security*, 88: 101634

[16] Beer A, and Felderer M 2018. Measuring and improving testability of system requirements in an industrial context by applying the goal question metric approach. In *Proceedings of the 5th International Workshop on Requirements Engineering and Testing* pp. 25-32

[17] Tsuda N, Washizaki H, Fukazawa Y, Yasuda Y, and Sugimura S 2018 Machine Learning to Evaluate Evolvability Defects: Code Metrics Thresholds for a Given Context. In *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)* (IEEE) pp 83-94

[18] Wingkvist A, Ericsson M, Lincke R and Löwe W 2010. A metrics-based approach to technical documentation quality. In *2010 Seventh International Conference on the Quality of Information and Communications Technology* (IEEE) pp 476-481

[19] Michael J B, Shing M T, Cruickshank K J and Redmond P J 2010. Hazard analysis and validation metrics framework for system of systems software safety. *IEEE Systems Journal* 4(2) 186-197

[20] Tan J, Rönnkö K and Gencel C 2013 A framework for software usability and user experience measurement in mobile industry. In *2013 Joint Conference of the 23rd International Workshop on Software Measurement and the 8th International Conference on Software Process and Product Measurement* (2013, October) (IEEE) pp. 156-164
[21] Schrettner L, Fülöp L J, Beszedes A, Kiss Á and Gyimothy T 2012. Software quality model and framework with applications in industrial context. In 2012 16th European Conference on Software Maintenance and Reengineering IEEE pp 453-456

[22] Venkitachalam H, Richenhagen J, Schlosser A and Tasky T 2015, May. Metrics for verification and validation of architecture in powertrain software development. In Proceedings of the First International Workshop on Automotive Software Architecture (2015, May) pp 27-33

[23] Zhao F, Nian G, Jin H, Yang L T and Zhu Y 2015 A hybrid ebusiness software metrics framework for decision making in cloud computing environment IEEE Systems Journal 11(2) 1049-1059

[24] Brockers A, Differding C and Threin G 1996. The role of software process modeling in planning industrial measurement programs. In Proceedings of the 3rd International Software Metrics Symposium (IEEE.) pp 31-40

[25] Debou C, Kuntzmann-Comelles A and Rowe A 1994. A quantitative approach to software process management. In Proceedings of 1994 IEEE 2nd International Software Metrics Symposium (IEEE) pp. 26-34

[26] Olsson T and Runeson P 2001 V-GQM: A feed-back approach to validation of a GQM study. In Proceedings Seventh International Software Metrics Symposium (IEEE) pp 236-245

[27] McGinn C, Bourke E, Kelly T O and Cullinan M F 2018. Adapting the Goals/Questions/Metrics (GQM) Method for Applications in Robot Design. In 2018 IEEE International Conference on Robotics and Automation (ICRA) (IEEE) pp 4746-4751

[28] Birk A, Derks P, Hamann D, Hirvensalo J, Oivo M, Rodenbach E and Taramaa J 1998. Applications of measurement in product-focused process improvement: A comparative industrial case study. In Proceedings Fifth International Software Metrics Symposium. Metrics (Cat. No. 98TB100262) (IEEE) pp 105-108

[29] Tahir T, and Gencel C 2010 A structured goal based measurement framework enabling traceability and prioritization. In 2010 6th International Conference on Emerging Technologies (ICET) (IEEE) pp 282-286

[30] Oinas A 2000 Defining goal-driven fault management metrics in a real world environment: a case-study from Nokia. In Proceedings of the Fourth European Conference on Software Maintenance and Reengineering (IEEE) pp 101-107

[31] Sibisi M and Van Waveren C C 2007 A process framework for customising software quality models. In AFRICON 2007 (IEEE) pp 1-8

[32] Geppert B and Weiss D M 2004. Goal-oriented assessment of product-line domains. In Proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry (IEEE Cat. No. 03EX717) (IEEE) pp 180-188

[33] Plöesch R, Schuerz S and Späthe S 2015. On the validity of the IT-CISQ quality model for automatic measurement of maintainability. In 2015 IEEE 39th Annual Computer Software and Applications Conference (IEEE) 2, 326-334

[34] Apel S, Hertlampf F and Späthe S 2019 Towards a Metrics-Based Software Quality Rating for a Microservice Architecture. In International Conference on Innovations for Community Services (Springer, Cham) pp 205-220

[35] Gordietiev O, Kharchenko V and Leontiev K 2018. Usability, Security and Safety Interaction: Profile and Metrics Based Analysis. In International Conference on Dependability and Complex Systems (Springer, Cham) pp 238-247

[36] Deb Nath N, Salgado C, Peralta M, Riesco D, Roqué L, Montejano G and Mazzi M 2019 A Software Testing Strategy Based on a Software Product Quality Model. In International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning (Springer, Cham) pp 248-259

[37] Schramme M and Macías J A 2019. Analysis and measurement of internal usability metrics through code annotations Software Quality Journal 27(4) 1505-1530