

Quality Quantification Applied to Automotive Embedded Systems and Software Advances with qualimetry science

Yann Argotti, Claude Baron, Philippe Esteban, Denis Chaton

▶ **To cite this version:**

Yann Argotti, Claude Baron, Philippe Esteban, Denis Chaton. Quality Quantification Applied to Automotive Embedded Systems and Software Advances with qualimetry science. Embedded Real Time Systems (ERTS) 2020, Jan 2020, Toulouse, France. hal-02382316

HAL Id: hal-02382316
https://laas.hal.science/hal-02382316

Submitted on 27 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Quality Quantification Applied to Automotive Embedded Systems and Software

Advances with qualimetry science

Yann Argotti[123], Claude Baron[23], Philippe Esteban[43] and Denis Chaton[1]

Summary — Quality quantification is an unavoidable topic in today daily company life. In this paper, the authors review why quality quantification is critical, what are the main difficulties with the current approaches and highlight the qualimetry approach as the solution. After a state of the art on qualimetry and on quality model concept strengthened with polymorphism, the first steps of their applications to automotive embedded systems and software in Renault are showcased. The results are not only the benefits in quality quantification for complex systems, such as homogeneity, consistency and compatibility, but also the highlighted risks with the changes in versions of quality models in *Automotive SPICE* and how to define a derivable quality model over electronic control units and vehicle.

Keywords — Qualimetry, quality model, polymorphism, metrics, measure, automotive, standards

I. Context and research objectives

A. The need to evaluate and quantify quality

Nowadays Renault is producing automotive systems at a high cadence. These automotive systems are very complex and embed many sub-systems. Evaluating and quantifying the level of quality of a system and of each sub-system is important, for different reasons exposed below.

First, a company such as Renault has to comply with many standards and regulation. This is obvious when we consider transportations systems such as cars or airplanes where we have to follow functional safety standards such as ISO26262 [1], ARP4754A[5] [2] and DO-178C [3]. Therefore, properly quantifying quality will tell us if we fulfill or not those standards.

Moreover, "quality quantification" covers both quality aspects (supporting the identification of the systems main characteristics) and quality models (supporting the organization of these characteristics). Quantification helps optimizing and controlling the large flow of metrics and measurements, and extracting the subset that makes most sense to Renault (or which is the most useful for Renault).

We can certainly find many other good reasons why quality quantification is important. However, missing some steps in quality quantification may sometimes turn into catastrophic scenarios. We can quickly cite a few well-known examples: the issue of software update with Therac-25 causing irradiation and death of 6 patients during 1985-1987 [4], Ariane 5 explosion on its first launch on the 4th of June 1996 [5] due to the reuse of the previous navigation system that was not aligned with the new rocket version velocity and then resulting on the loss of $370 million, on the 26th of June 2017 Takata's bankrupt happened due to an unaddressed known bug in their airbag [6] and on 2018, Toyota recalled 2.4 million hybrid cars because of a failure in the "failsafe" driving mode linked to an uncaught software issue [7]. Through these four examples, we have four different systems with four different quality quantification contexts, and an obvious demonstration that their consequences, measured in term of people loss and / or budget, were catastrophic, thus highlighting the need to have not only a reliable and accurate quality quantification approach, but also adapted to system usage context.

The quality addressed in this paper is the quality of product during its entire life cycle, including development (requirement analysis, design, implementation), maintenance and operation.

---

[1] Renault SW Labs

[2] INSA Toulouse

[3] CNRS - LAAS

[4] Université de Toulouse III, Paul Sabatier

[5] The authors would like to point out that Aerospace Recommend Practice (ARP) is a guideline coming from SAE International, and originally SAE International was initially established as the Society of Automotive Engineers on 1905.

B. Many possibilities but difficulties to find the optimum approach

We understand from above that quality quantification is critical; but depending on which quality quantification approach is used, we may face different types of challenge. The first case we can face is when the solution we are considering to quantify quality is too general and then requires a certain level of refactoring or tailoring without any guarantee to get to the right solution. This is often the case with references or standards like CMMI [8], ISO/IEC9126 [9] or ISO/IEC25010 [10] which have the ambition to cover as many types of systems/domains of application as possible. A study conducted by Wagner *et al.* [11] showed that 79% of companies that use standards customize them. At the opposite, the solution can be too specific and then reuse against another more or less close systems can be hard; this is the case for instance with Factor / Criteria / Metric from McCall *et al.*[12] or with Basili *et al.* [13] and the Goal / Question / Method approaches. A third possibility is that the solution set we have is too large, and by consequence there is not an obvious right solution; for instance, it is the case of software products, for which more than 40 distinct quality models can easily be identified in the literature. Finally, the last case is when we have both theoretical and applied aspects for quality quantification, like Wagner [14] on software product quality control or Azgaldov *et al.*[15], [16] on general quality assessment; these approaches may be a little bit heavier to use but they offer a large potential: they are part of *Qualimetry*.

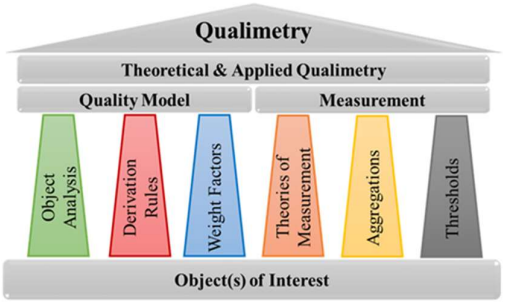


Fig. 1 -The house of qualimetry and its pillars [17]

In the next sections, we complete an overview of the state of the art about qualimetry and quality model concept, strengthening this last one with the innovative introduction of polymorphism. Then we apply these concepts to the automotive domain, encompassing embedded systems and embedded software before concluding on the results, the benefits and the next steps.

II. State of the art on qualimetry and on quality model concept

A. Qualimetry

Qualimetry is the science of quality quantification [16] and consequently covers both "*intellectual and practical activity encompassing the systematic study of [...]*"[18] quality quantification. These two activities are reflected by the theoretical and applied qualimetry. In addition, even if this concept is not recent, this science is relatively young because it was born in former USSR in 1968 [15], following the creation of a working group composed of scientists coming from various areas (e.g. economists, architects, civil engineers, car makers) who were sharing the same goal [16]: generalizing quality quantification approach. In order to foster its understanding and support its knowledge leveraging within quality quantification scope, we have proposed a synthetic representation, the *House of Qualimetry* (see Fig. 1).

By definition, qualimetry is a science, so it addresses both theory and application dimension. These two dimensions themselves rely on two domains: quality model -focusing on the quality characteristics- and measurement -addressing the quantification part. Quality model sits on three pillars. The first one is related to the analysis for identifying quality characteristics while the second one support their organization and add some rules to control analysis depth. The last pillar here is covering characteristic importance among other characteristics. On the measurement side, we retrieve the same topology. The first measurement pillar is key because it is about the different theories of measurement and therefore brings all the mathematical and statistical tools. Second pillar is reflecting the measurement combination or data aggregation together depending on their purpose. Third and final pillar supports the use of measurements during assessment, control and prediction: this is the threshold pillar. Completing the description, the basement of the house is the object(s) of interest that is to say the object(s) candidate for quality quantification. The two ISO/IEC25010:2011 quality models (*i.e.* "*systems/software product quality*" and "*quality in use*") [10] illustrate the result that can be achieved by applying the two quality model first pillars against automotive embedded software product[6].

[6] The reason why we are linking to ISO/IEC 25010 quality model is due to the fact that Automotive SPICE [19] -the process assessment and reference model in automotive field- is referring to that standard for software product quality model.

B. Quality Models

To understand the quality model concept more precisely, we can rely on ISO/IEC IS 9126-1 [9] where a quality model is defined as "*the set of characteristics, and the relationships between them that provides the basis for specifying quality requirements and evaluation*". This definition can be completed by ISO/IEC 25010 [10] which highlights that a quality model is "*convenient breakdown of product quality*", "*serve as a framework to ensure that all characteristics of quality are considered*" and "*provide a set of quality characteristics relevant to a wide range of stakeholders, such as: software developers, system integrators, acquirers, owners, maintainers, contractors, quality assurance and control professionals, and users.*".

To go one step further on the quality model knowledge, we have conducted a study [17] to be able to isolate a pattern related to quality models relative to their design, conception or adaptation. We have identified a set of eight attributes (cf. TABLE 1): six shared between all approaches (*evaluation context & plan*, *purposes*, *Quality Evaluation Methods (EQM) to assess quality*, *QEM as source of information about values*, *data organizational types* and *weight factors*), one unique, the *derivation rules* coming from qualimetry field and one new, absent from any previous related streams of work: the *polymorphism*. Moreover, the notable fact with these attributes is that, if we handle or consider them sequentially, we land with a unified conception process to create or adapt quality model, starting from "*evaluation context and plan*" up to "*polymorphism*".

TABLE 1 – THE 8 QUALITY MODEL ATTRIBUTES EXERCISED AGAINST SAME STANDARD EVOLUTION: ISO/IEC 9126:2001 & ISO/IEC 25010:2011

	Attribute	ISO/IEC 9126:2001	ISO/IEC 25010:2011
#1	**Evaluation context & plan**	Information Technology Software product quality & quality in use	System (computer oriented) & Software product quality, quality in use & data quality
#2	**Purposes**	Definition & Assessment (evaluation)	Definition & Assessment (evaluation)
#3	**QEM to assess quality**	Short-cut method	Short-cut method
#4	**QEM as source of information about values**	Hybrid method	Hybrid method
#5	**Data organizational types**	Hierarchical	Hierarchical (& meta-model)
#6	**Derivation rules**	Respect of global rules with exception of rule #5 (reliability)	Respect of global rules with exception of rule #5 (reliability)
#7	**Weight factors**	Not weighted	Not weighted
#8	**Degree of polymorphism**	**0.6792** (0 = identical; 1 = 100% disjoined) [53 leaf characteristics, 32 unique, 8 similar]	

C. Polymorphism

With regards to this last attribute, we consider two aspects for polymorphism concept applied to quality model: *ad-hoc* and temporal. To explain what is behind these aspects, we can make an analogy with biology: let us compare a quality model to a butterfly. For the first aspect, we start with a generic butterfly which has a set of characteristics (two wings, a trunk, three pairs of thoracic legs, two antennas …). In the real world, we have many variants of this generic butterfly that can be more or less close to each other (wing color, pattern and shape, size, lifestyle …). Each of this variant inherits from the generic butterfly characteristics. Thereby, we can have variants of quality models inheriting from a generic quality model. The second aspect is linked to a temporal consideration. Like the butterfly, starting as an egg, then becomes a caterpillar, chrysalis, then a new born butterfly and comes up with a flying butterfly, a quality model can change, evolve depending of the systems or software product life cycle.

Continuing one step further with biology, and more particularly with genetic, we borrow a formula (1) introduced by Nei and Li in 1979 [20], used to compute the degree of polymorphism between DNA sequences and we apply it to quality model. Thanks to this mechanism we are able to compute distances between quality models from a polymorphism or variety point of view.

$$\pi = \sum_{ij} x_i x_j \pi_{ij} \qquad (1)$$

The π_{ij} coefficients are calculated by considering the probability to have a specific quality characteristic / sub- characteristic. This calculus is based on a pool of quality models. For instance, if a characteristic recurrently appears in those specific quality models, its probability is 1. If half of the time the characteristic is present and the other half is another close (*i.e.* not disjoint) characteristic, then their respective probability is 0.5. When applying this approach to ISO/IEC 9126 and ISO/IEC 25010, we identified: 53 leaf characteristics, 32 unique, 8 similar (*i.e.* close but not identical: for instance, "*Modifiability*" versus "*Changeability*"), and 13 identical. This lands us to find with (1) 67.92% of differences.

On the measurement side, we had to enhance current measurement process to include consideration to the pillars of quality model and measurements. Also, like a processor, we are cadencing the measurement process with the systems or software development life cycle, to integrate the temporal aspect of polymorphism we indicated previously.

To summarize, polymorphism is an help in system engineering where we have a context of systems and/or sub-systems that define a system. Polymorphism brings consistency and support to adaptation due to the context and stakeholder variety.

D. Quality Model distance impacts

The use of polymorphism variety formula is a great tool that help us estimating and explaining what the impacts and consequences are to change, update or adapt current quality model or to apply one quality model instead of another one.

Indeed, the consequences are directly linked to what we aim to do with quality model. For instance, let say that a company was currently using ISO/IEC 9126 and want to be compliant with latest available standard, which is ISO/IEC 25010, then this distance will help to understand and estimate:

- what the risks are linked to such change (low distance = low risk, high distance = high risk),
- what are the areas that are the most impacted (where we have more change, declining the distance for each quality characteristics),
- how much work it will cost,
- where quality quantification, assessment and control are changing,
- how much validation path is changed finding, allowing to capture different types of bugs possibly never found before and discarding other areas and path,

Changes of quality models can occur due to change or evolution of targeted product or stage in its life cycle. Consequently, this may lead in different results and the distance can predict that we may get different results.

In addition, this formula can be used to support decision and to control change or update quality models, including the case of polymorphism: when distance is low, change can be ignored while a high distance tells us that we need to apply this change. Finally, the distance formula can help to split quality model changes into reasonable, from a workload and risk point of view, change increments.

III. Application to automotive

A. With regards to embedded systems

Thanks to this overview of the quality quantification from the qualimetry perspective, we are in a position where we can apply those concepts to the automotive field, thus answering Renault's needs which are: to have a robust, efficient, homogeneous, compatible and consistent quality quantification as well as specify a joint "vocabulary" over the entire complex system that a vehicle is.

Indeed, a car is an instantiation of a vehicle platform which is then addressing a set of car variants such as mini-compact, convertible, super car, cross over, commercial, van…. Therewith, a car is a complex system, composed of more than 40 systems themselves distributed over more than 60 Electronic Control Units (ECU), depending on whether this is a low-end, medium or premium car.

Moreover, besides the fact that each ECU is itself composed of a hardware and an embedded software, an ECU has some common characteristics shared with other ECUs (e.g. diagnostic, connection interface, power), a set of specific characteristics (e.g. HMI, communication, safety) and its own context (e.g. door control, engine control, telematic control, seat control). As a matter of fact, each such subsystem has a vocabulary more or less specific and quality quantification which vary more or less from the other sub-systems. This system complexity description depicts and corresponds to the complexity we have in Renault.

B. With regards to embedded software

Concerning quality model for automotive embedded software, Automotive SPICE[7] [19] advises to rely on ISO/IEC 25010:2011 for embedded software product quality model in its measurement process, called

[7] Renault is relying on Automotive SPICE with regard to its software development activities

MAN.6. However, we remark that in previous versions of Automotive SPICE, such as v2.5 [21], the standard that was referred to for embedded software quality model was ISO/IEC 9126:2001. Moreover, ISO/IEC 25010:2011 standard is the extension, or evolution [10], of ISO/IEC 9126:2001standard. Thus, to see how close or diverse the quality models from these two standards are together, we extract the corresponding pattern instantiations from the height quality model attributes we saw in previous paragraph. TABLE 1 summarizes the comparison results. We notice that most of the attributes are equal except the first one which deals with "*evaluation context and plan*", but this is something that we could expect since ISO/IEC 25010:2011 is an evolution of the other one with a wider scope.

Now if we compute the degree of polymorphism between these two quality models (as seen in II.C), we obtain a result indicating a diversity of almost 68% which means that finally those quality models are quite different. Therefore, this is a drastic evolution, or change despite the fact that in ISO/IEC 25010:2011 document it is claimed this is an extension. Also, upgrading quality model from previous standard to new one can bring risk, particularly if your current quality model works well. Fig. 2 highlights some quality model differences between these two standards.

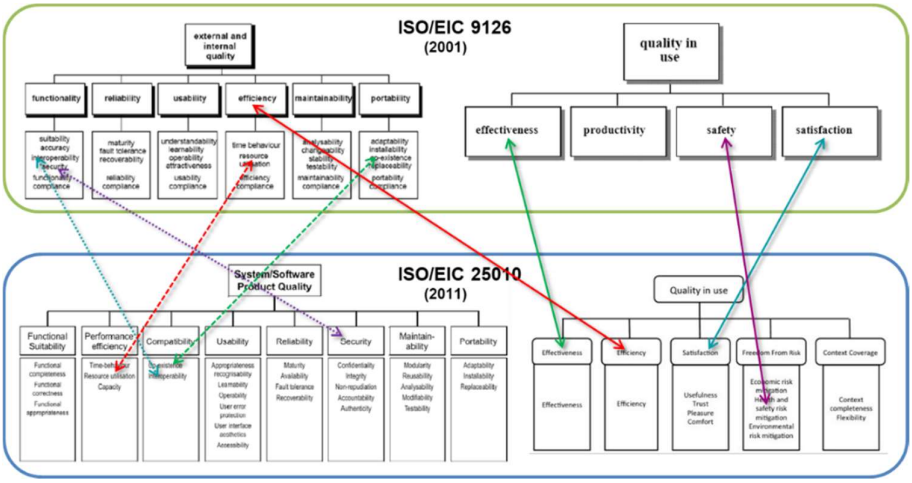


Fig. 2 - Example of some differences between ISO/IEC 9126:2001 & ISO/IEC 25010:2011

C. With regards to polymorphism

In the above sub-sections A and B, we have captured enough knowledge to begin applying polymorphism to automotive embedded software, for instance. In the following paragraphs, we limit our polymorphism application to one level of quality characteristic enumeration[8].

First, we initiate the elaboration of a common ECU quality model using A-SPICE 3.1 guidelines [19]. We note that these guidelines refer also to a subset of ISO/IEC 25010 [10]. The result is a quality model composed of 6 quality characteristics: functional suitability, reliability, usability, performance efficiency, maintainability and portability. All those quality characteristics are aligned with a scope of embedded software. Then, in order to apply polymorphism to this common ECU quality model, we consider two distinct variants, among many, of this common ECU in our study case here: **I**n **V**ehicle **I**nfotainment (*i.e.* IVI) ECU and **B**ody **C**ontrol **M**odule (*i.e.* BCM) ECU. The IVI ECU is responsible for infotainment and is the main human user Interface to control different options of the vehicle. In that sense, the performance efficiency is not as important as quality in use aspect which must include efficiency, effectiveness and satisfaction. Indeed, since human-machine interface is key for this ECU, the right performance criteria must be relying on the user perception of the performance rather than pure processing time for instance. Regarding BCM, this ECU can be seen as the main vehicle gateway, dealing with various communication protocols but with no direct interaction with the user (*i.e.* there is. human-machine interface). Then security and safety aspect, included into freedom from risk characteristics, are major quality characteristics to cover for this ECU.

Fig. 3 below illustrates this example of how to apply polymorphism with a subset of quality characteristics from generic ECU quality models to IVI ECU quality model and BCM ECU quality model,

[8] we don't include any sub-characteristics in our example, but we encourage the reader to consolidate current quality models with further sub-characteristics and metrics.

using ISO/IEC 25010 as complement guidelines. In black characteristics defined in Common ECU quality models, in red characteristics that may be discarded/not used, in green complementary characteristics.

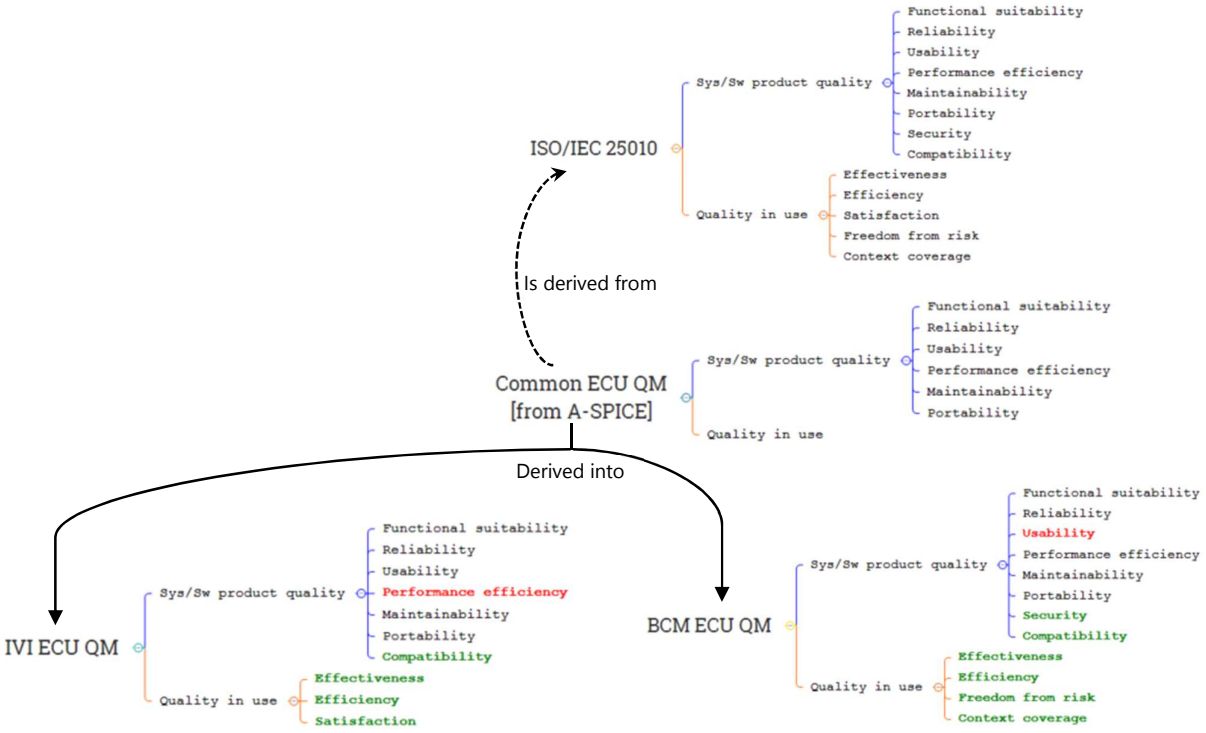


Fig. 3 - Quality model polymorphism example applied to two ECUs

The early results of these concept appliance in Renault are depicted through several dashboards: on code for a subset of non-functional characteristics from ISO/IEC 25010:2011 (see Fig. 4) and with regards to generic quality model for ECU linked applied to their related domain[9] (see Fig. 5).

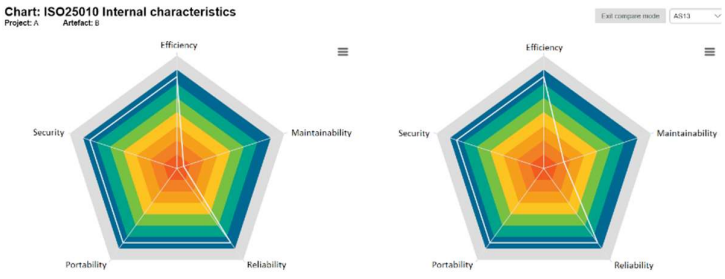


Fig. 4- Example of an evolution of AD ECU code metric results vs a subset of ISO/IEC 25010:2011 characteristics

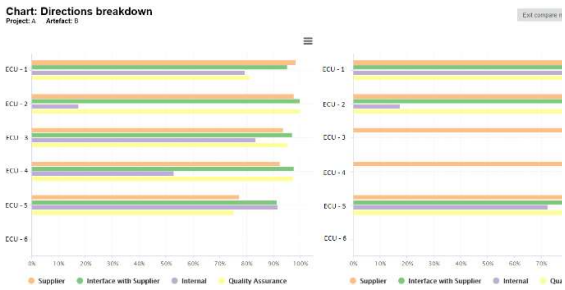


Fig. 5- Example of an evolution of generic quality model result per ECU domain

IV. Conclusion

In conclusion, we have seen what is qualimetry, consolidating some of its aspects, and how it can support generalization, adaptation and repetition in quality quantification activities. It can be as well applied and used for other activities in systems and software engineering, even at the early stage, supporting, for instance, conception, requirement elicitation and architecture design. In addition, our

[9] In Renault, a domain is covered by a specific department or "*direction*".

qualimetry based approach brings homogeneity, consistency and compatibility to quality quantification in the complex environment which is the automotive one. It helps specifying a joint vocabulary where each domain has its own. We also are in a position where we can define a derivable quality model for ECU and vehicle platform thanks to polymorphism. And finally, in a context of agile development methodology, our approach allows a smooth incremental change management.

Our next steps in Renault will focus on the consolidation and then deployment of our current generic quality model as well as the various specific quality models related to the ECU variants.

References

[1] "ISO 26262-6:2011 - Road vehicles - Functional safety - Part 6: Product development at the software level," *International Organization for Standardization*, 2011.
[2] "ARP4754A - Guidelines for Development of Civil Aircraft and Systems," *SAE International*, Dec. 2010.
[3] "DO-178C - Software Considerations in Airborne Systems and Equipment Certification," *Radio Technical Commission for Aeronautics*, Dec. 2011.
[4] N. G. Leveson and C. S. Turner, "An Investigation of the Therac-25 Accidents," *Computer*, vol. 26, no. 7, pp. 18–41, 1993.
[5] R. L. Baber, "The Ariane 5 explosion: a software engineer's view," *Risks*, vol. 18, no. 89, Mar. 1997.
[6] Reuters, "Takata's U.S. Unit Reaches Deal Paving Way for Sale.," *The New York Times*, 12-Feb-2018.
[7] S. McLain, "Toyota Recalls More Than 2 Million Vehicles Over Hybrid-System Fault," *The Wall Street Journal*, 05-Oct-2018.
[8] M. C. Paulk, B. Curtis, and M. B. Chrissis, "Capability maturity model, version 1.1," *IEEE Software*, vol. 10, no. 4, Jul. 1993.
[9] "ISO/IEC 9126-1:2001 - Software engineering - Product quality - Part1: Quality Model," *International Organization for Standardization*, 2001.
[10] "ISO/IEC 25010:2011 - Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models," *International Organization for Standardization*, 2011.
[11] S. Wagner, K. Lochmann, S. Winter, A. Goeb, M. Kläs, and S. Nunnenmacher, "Software Quality Models in Practice: Survey Results," Technische Universität München Insitut für Informatik, TUM-I19, 2012.
[12] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in Software Quality," *Griffiths Air Force Base, N.Y. Rome Air Development Center Air Force Systems Command*, 1977.
[13] V. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric Approach," *Encyclopedia of Software Engineering, John Wiley & Sons, Inc.*, pp. 528–532, 1994.
[14] S. Wagner, *Software Product Quality Control*, Springer-Verlag Berlin Heidelberg. 2013.
[15] G. G. Azgaldov *et al.*, "Qualimetry: the Science of Product Quality Assessment," *Standart y i kachest vo*, no. 1, 1968.
[16] G. Azgaldov, A. Kostin, and A. Padilla Omiste, *The ABC of Qualimetry, toolkit for measuring the immeasurable*, Ridero. 2015.
[17] Y. Argotti, C. Baron, and P. Esteban, "Quality quantification in Systems Engineering from the Qualimetry Eye," presented at the 13th Annual IEEE International Systems Conference (SysCon), Orlando, USA, 2019.
[18] "Online Oxford Dictionary - science definition," 2019. [Online]. Available: https://en.oxforddictionaries.com/definition/science.
[19] VDA QMC Working Group 13 / Automotive SIG, "Automotive SPICE Process Assessment / Reference Model., version 3.1 - revision 656." 01-Nov-2017.
[20] M. Nei and W.-H. Li, "Mathematical model for studying genetic variation in terms of restriction endonucleases," in *In Proceedings of the National Academy of Science of the USA*, 1979, vol. 76, pp. 5269–5273.
[21] Automotive SIG, VDA, "Automotive SPICE Process Assessment, version 2.5." 10-May-2010.