



HAL
open science

Activity Report: PhD 1st Year

Yann Argotti

► **To cite this version:**

Yann Argotti. Activity Report: PhD 1st Year: Study of Qualimetry essentials applied to embedded software product and organization, with consideration to software entropy. LAAS / CNRS; Renault Software Labs. 2019. hal-02051994

HAL Id: hal-02051994

<https://laas.hal.science/hal-02051994>

Submitted on 1 Mar 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.

IC/ISI Group

Activity Report: PhD 1st Year

Study of Qualimetry essentials applied to embedded software product and organization, with consideration to software entropy

ARGOTTI Yann
2-21-2019

Table of Contents

1- Introduction	2
2- Scientific Context	2
3- Goals	4
4- Organization of research work.....	6
5- Current achievements.....	7
6- Conclusion.....	10
References	11
Annexes.....	13
A- First part of quality model classification and decision tool: get_synonyms.py	13
B- Accepted conference paper: IEEE System Conference 2019, 8 th of April	14
Figure 1 - Genealogical tree of quality models - scope is related to web services [29]	3
Figure 2 - An example of Darwing diagram or cladogram	5
Figure 3 - Current research and development flow.....	6
Figure 4 - The House of Qualimetry and its six pillars [1]	7
Figure 5 - The unified quality model conception, aligned with the three “Quality Model” pillars	8
Figure 6 - Our measurement process proposal articulated over three phases.....	9
Figure 7 - The measurement refinement process	10
Figure 8 - Our objects of interest: organization, process, project, people, supplier, service and software product	10

1- Introduction

The purpose of this report is to summarize the first-year of research activity related to Yann Argotti's PhD inside the IC/ISI group at LAAS-CNRS. This PhD candidate is registered at EDSYS school and INSA Toulouse in "Computer science and embedded systems" specialization track. Claude Baron, IC/ISI group responsible and Philippe Esteban, member of IC/ISI group, are co-joint thesis directors. Thesis subject is "Study of Qualimetry essentials applied to embedded software product and organization, with consideration"

During this first year, current topic was analyzed to understand the problematic. This analyze was completed by a state-of-the-art start that allowed to identify what are the current technical road blockers and to have the first contributions synthesized over the production of a research paper [1] that was accepted for the incoming IEEE System Conference 2019 (see Annex). In parallel, the overall thesis approach was refined and consolidated with the need for creating a tool to support the research activity.

Thus, in next parts of this report we are going to review the scientific context linked to our problematic, the goals we are targeting with this PhD work, how we are organizing our research work, then go through the current achievements and finally conclude.

2- Scientific Context

Today quality is a key project aspect that follows and drives software development since its beginning. Delay and coding workload reductions, technical back and forth optimization linked to qualification, customer acceptance and sustaining phases are essential stakes in project cost: solely, these activities represent around 65% of overall cost and bad testing processes can increase the risk of project delay or cancellation by 25 to 300% [2]. Moreover, like in thermodynamics, software entropy [3], or complexity, increases during software system lifetime, and then it is mandatory to measure those "disorders" to avoid to lose project control over time.

Qualimetry [4], [5] approach brings a set of good practices, fosters dysfunction detection, enhances control, increases efficiency and productivity not only to developers but also to overall organization. Therefore mastery of this scientific discipline is decisive in terms of efficiency improvement, through standardized quality model[6], metrics and controlled process of software development.

Indeed, this qualitative approach, or control, of the quality is realized via quality models which can follow either an adherence to process or capability level (e.g. CMM [7], ISO/IEC 15504 [8]) or assessment through a set of attributes and metrics (e.g. McCall [9], ISO/IEC 25010 (ie SQuaRE) [10]). In addition, since 1977 with McCall's model [8], many software quality models were regularly defined or derived from others, including some effort of normalization. We can quickly identify Boehm's model [11], [12] in 1978, ISO/IEC 9126 [6] in 1986, FURPS /FURPS+ model [13] in 1987 and then 2000, Evans & Marciniak's model [14] in 1987, Deutsch & Willis' model [15] in 1988, Capability Maturity Model (ie CMM) [7] and Ghezzi's model [16] in 1991, Dromey's model [17] in 1992 and 1995, IEEE model [18] in 1993, SEI model [19] in 1995, SATC Model [20] in 1996, Bansiya's Quality Model for Object Oriented Design [21] in 2002, Kazman's model [22] in 2003, Aspect - Oriented Software Quality Model [23] in 2006, Component based Software development Quality Model [24] in 2008, DEQUALITE model [25] in 2009, UML Conceptual Model [26] in 2010, Sehra S.K.'s model [27] and ISO/IEC 25010 (ie SQuaRE) [10] in 2011, ... and Optimized Quality model for Agile Development [28] in 2017. Aligned to what *Figure 1* - extracted from [29] - is also depicting, we can then state that there is no obvious quality model best choice, despite comparison studies such as [30], [31] or [32], and none of the existing quality models are bridging the assessment and the adherence sides.

Finally, it is interesting to notice that software entropy, or "disorder", over time measurements, could be characterized by software aging¹ [33], [34] and can be measured thanks to reliability quality characteristics [10]. However, by definition the related sub-characteristic metrics are performed in the context of continuous software execution and don't include software maintenance or update impacts during the software product life: slowly, the software product is being degraded by itself. We can conclude that strengthening a quality model may require also further consideration to product life vs product realization with some metric refinements.

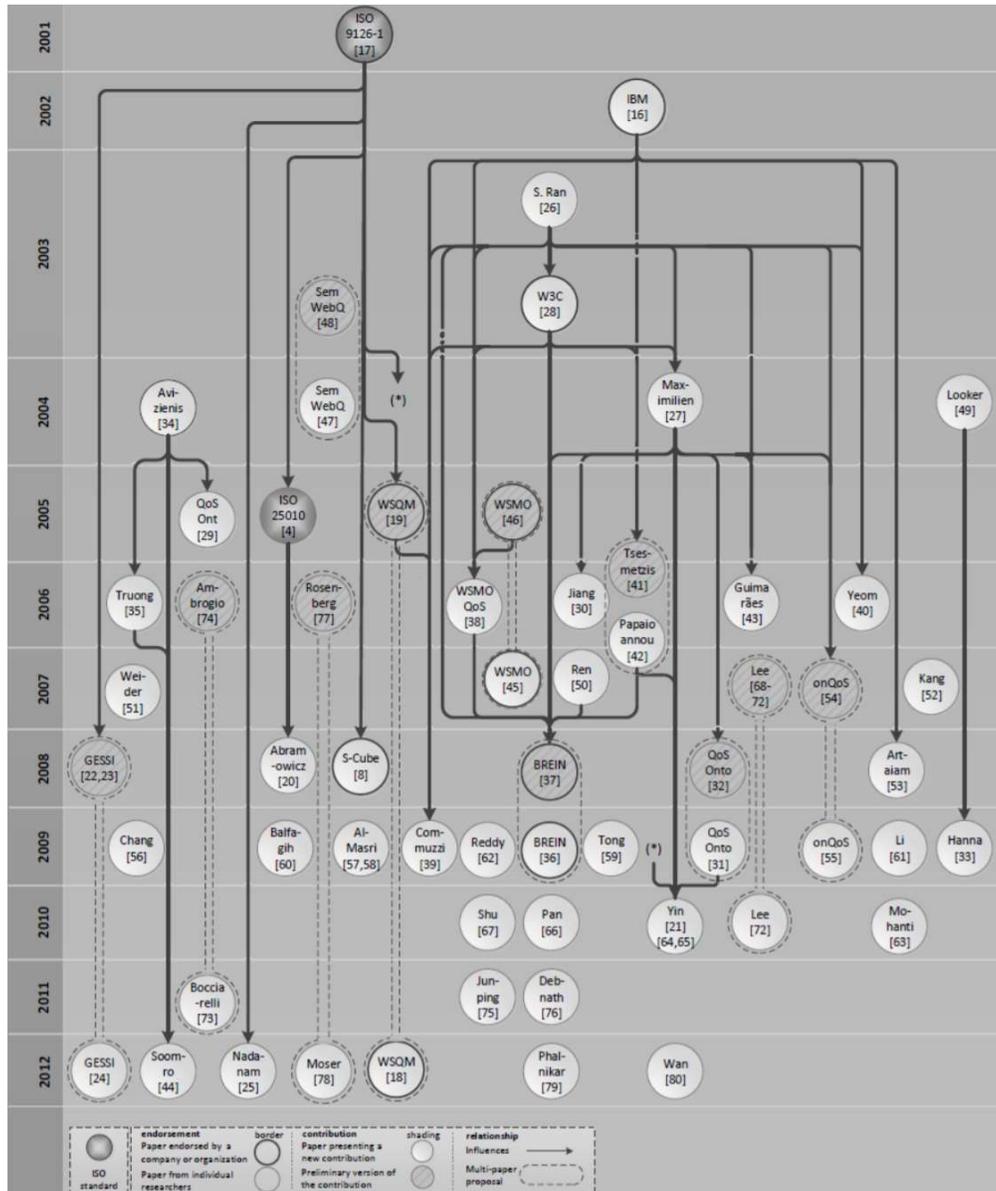


Figure 1 - Genealogical tree of quality models - scope is related to web services [29]

¹ Our assumption here is that we consider software from the very beginning of its implementation (i.e. at its early age phase) and then, over time, "disorder" is growing due to integration of patches to either bring new functionality or issue fix, and also due to its execution.

3- Goals

Since Qualimetry is by definition the quantification of Quality for any object, including process, and considering the technological barriers seen in previous section, the inceptive thesis main goal is to:

Main Goal

Define and evaluate an optimized Quality Model in order to bridge and quantify quality not only for software development (ie requirements, models, source code) but also for software organization, with conformance² to ASPICE MAN.6 [35], ISO/IEC 25010 [10], ISO 26262 [36] and ISO/TS 16949:2009 [36].

Secondary Goals

Secondary initial thesis goals are to exercise this quality model and achieve:

- Software Maturity within Continuous Integration process measurement
 - Traceability / Quality by design / Change impact
 - Test efficiency
 - Project Landing zone (Time to market vs risk & complexity)
- Software Aging measurement
 - Software Model Aging (e.g. Impact from Maintenance, FOTA and/or Complex system vs impact to safety & reliability)
 - Product Aging Landing zone

However, during this first year, our further study and analysis performed on Qualimetry and quality models required us to refactor and consolidate these originals goals. Indeed, we were able not only to confirm our early statement that there is no obvious solution for quality model for software development and for software organization, but also that there is a misunderstanding about Qualimetry³ [1] and we identified gaps on quality model classification and on decision for the quality model solution to apply/use. Comparison studies we can found in [30], [31] or [32] illustrate clearly that fact. Only some work done by Oriol *et al.* [29] including ontology consideration open the perspective of a solution. So, to be able to address our main initial goal, we can decompose it into:

Main Goal: sub-goals

1- Taxonomy for quality model

- a. Build a taxonomy for quality model,
 - Complete state of the art on quality model classification, ontology and classification in general,
 - Build taxonomy based on
 1. Ontology with synonym and word constellation⁴,
 2. Our unified quality model conception,
 3. Definition of quality model pedigree,
 4. Our polymorphism degree measure,
 5. Any additional relevant approaches seen in state-of-the-art part,
- b. Apply to our case study that taxonomy,
 - Classify SW product quality models
 - Identify SW product quality model main genes
 - Deliver a kind of “*Charles Darwin Diagram*” or cladogram, *Figure 2*, of SW product quality models

2- Define an oracle for decision to get optimum quality model solution,

- a. Build a decision oracle for quality model,
 - Complete state of the art on decision related to quality model and on oracle in computer science in general,

² The company, welcoming student here, is in automotive field which therefore drives standard choices here.

³ In general, qualimetry is understood as “applied qualimetry” and not really as qualimetry as a science.

⁴ See the <http://www.atlas-semantiques.eu> site for more details

- Identify inputs, context, constraints, heuristics and inference rules ...,
 - Build oracle: Decision means “deciding which quality model fits the practical needs” => context
- b. Apply to our case study that oracle,
- Proceed on case study: sw product quality model
 - Evaluate result

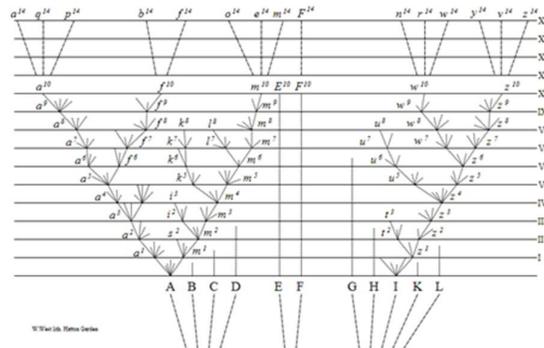


Figure 2 - An example of Darwin diagram or cladogram

To help and support these two tasks, we are adding a third sub-goal that must be achieved in parallel to these two ones: this the creation a unique tool and quality model database. Moreover, that tool will demonstrate that our proposal is viable and usable.

Main Goal : sub-goals

3- Define and implement a tool to support quality model taxonomy & oracle

- Compute automatically distance between two models,
- Describe Quality model seen in state of the art (use Yaml for quality model description, pedigree, setup bridge/dictionary of “synonym” =><http://www.atlas-semantiques.eu/?l=EN>),
- Apply the computation over the multiple quality model combination,
- Darwin’s diagram / Cladogram generation,
- Quality model decision helper (oracle?),

Thanks to this goal refinement, we can rewrite the first initial secondary goal relying on the taxonomy and oracle studied and developed in our main goal. In addition, we understand that the amount of work required to achieve it may bring risk to not be able to complete on time last initial secondary goal (ie software aging). Therefore, the rewritten secondary goals are defined below:

Rewritten Secondary Goals

- **Quality Models and Metrics linked to CI**
 - Identify characteristics linked to CI: iteration, change patch, main characteristics/sub-characteristics...
 - Identify Weight factors and applicable code metrics
 - Entropy metric within quality model and CI
 - Use taxonomy and oracle against CI context
 - Compare and impact result analyze: w/ & w/o weight factors, w/ & w/o a quality model, entropy evolution
- **Software Aging measurement**
 - Summarize problematic and difference against regular (ie often associated to reliability) vs our definition of Software Model Aging
 - Takes examples: Impact from Maintenance, FOTA and/or Complex system vs impact to safety & reliability, product aging landing zone

4- Organization of research work

Research work will be organized into several incremental steps, organized around the goals defined and refined in previous section.

Indeed, the first step is to proceed on study and analyze the current concepts and problematics behind the thesis subject. This study shall be based on a rigorous state of the art, started from the listed references, on qualimetry, quality quantification, quality models and our fields of interested which are embedded software product and its organization. As written in previous section, our analysis concluded on the need to build a taxonomy for quality model and then followed by a need to construct a methodology on decision / oracle to generate optimum quality model solution. This will be completed with the related metrics which must include independent (e.g. Function Point, COSMIC [38]) or not measures of technical or quality characteristics, considering also applicable norms and standard for embedded software development, process (agile [28] and in V-cycle) and software organization.

This global approach is acting on the theory field. However, we will need to exercise regularly our findings against the practical field, applying on some specific case studies, such as software product quality models and continuous integration process. These experiments phases will allow us to loop back to correct, consolidate, optimize our taxonomy and decision methodology. We are planning multiple back and forth loops between theory and practice field to be able to converge on a suitable and practicable solution. To support also these experiments, we are planning to implement a tool, with a first aim on computing distance against two quality models.

The research work organization is summarized in *Figure 3*. We see the various task sequence, the theory vs practice aspects, and their loopback and the tool supporting our overall approach. We can notice that we have completed first task and are currently focusing on the quality model taxonomy (or classification methodology).

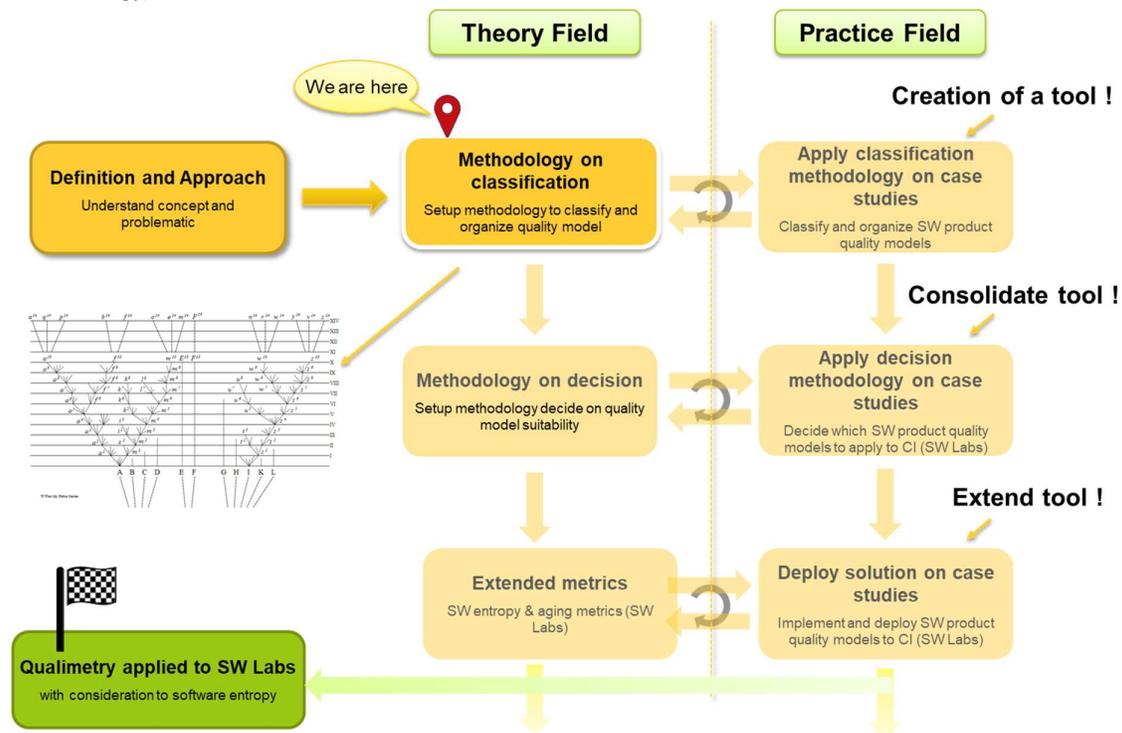


Figure 3 - Current research and development flow

5- Current achievements

In the section 4-, we saw how we organized our research work to answer to our goals and that our initial task, “*Definition and Approach*”, is done now. During this task, where we proceeded on state-of-the-art studies, we noticed that Qualimetry science was often misinterpreted, considering it more from its applied side and therefore we noted some common mistakes such as missing weight factors, or derivation rules for example. Thus, to leverage knowledge about this young science⁵ and reconcile both theoretical and apply aspect, we built a synthetic view of what Qualimetry is via what we named: the “*House of Qualimetry and its six pillars*” (cf Figure 4).

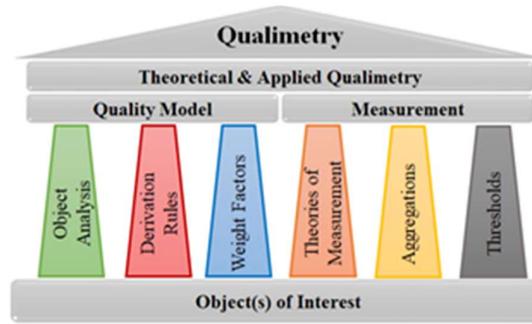


Figure 4 - The House of Qualimetry and its six pillars [1]

As a result of this analysis, we investigated both sides: quality model and measurements. On the quality model we identified three main streams of work. The first one, done by Wagner [40], synthetize research on quality control for software products based on quality model and measurements. The second one corresponds to the qualimetry point of view, relying on Azgaldov et al. [4], [39]. The last stream we identified is the one composed of all the common research and standard approaches about quality models. We summarized our analysis in a Table I, showing some commonalities but also some differences and gaps.

Stream of approach	Wagner <i>et al.</i> [18], [47]	Quality models such as ISO/IEC 25000 [9], [23]–[27], [29]	Azgaldov <i>et al.</i> [12], [48]	
Quality model scope	Project and Software product	System and Software product and in use	Any area	
Attributes	1 Evaluation context & plan	none	Evaluation plan	Evaluation context
	2 Purposes	• Definition • Assessment • Prediction • Multi-purpose	• Definition • Assessment (evaluation part)	• Definition • Assessment
	3 QEM: method to assess quality	Not specified but assumes approximate method	Not specified but assumes approximate method	Rigorous method • Short-cut method • Approximate method
	4 QEM: source of information about values in QEM	Not specified but assumes expert method	Not specified but assumes expert method	• Expert method • Non-expert method (i.e. analytical method) • Hybrid method
	5 Data organizational types	• Hierarchical • Meta-model • Statistical and implicit	• Hierarchical • Meta-model	• Hierarchical
	6 Rules to derives trees	none	none	~30 rules
	7 Weight factors	Per property / characteristic	Per property / characteristic	Per property / characteristic
	8 Polymorphism	none	none	none

Table I - Comparison of the main three distinct streams of work supporting quality model specification

The rows of our table are summarizing the distinct attributes that we can use to characterize or design quality models without limiting ourselves to a specific domain such as software product, for instance. Then

⁵ Qualimetry appeared as the genesis as a new science in former USSR since 1968 [39].

for each row, or attribute, to generate its consolidated definition, we take the union of the results from each stream. We note that there is one exception in our table. The “polymorphism” attribute is missing from all these three streams of approach.

Indeed, this is our third contribution. We introduced the polymorphism concept - concept frequently used in programming-oriented object- for quality model to reflect the facts that

- 1- For same type of objects
 - a. We may have common quality model “interface” (ie *Ad hoc polymorphism*) or characteristics,
 - b. We may have some heritage between quality models with more or less important variation of characteristics / sub-characteristics between “sister” quality models (e.g. quality model for a generic chair can be derived as one for long chair and one for rocking chair), (ie *subclassing or inheritance*)
- 2- Over a project or product life cycle, for example, quality model can change (e.g in design phase we have different focus than in maintenance one),

Interestingly, genetic offers us a way to measure variety or polymorphism. This is the nucleotide diversity formula (1) introduced by Nei and Li in 1979 [41]. With this fourth contribution to our research work; we introduced a mathematical way to define and compute distance between quality models.

$$\pi = \sum_{ij} x_i x_j \pi_{ij} \tag{1}$$

As final step to benefit to this study related to quality model, we organized these height quality model attributes in a way where, if we rely on their sequential use, we have a unified conception process for quality model. *Figure 5* shows this process. The colored square boundaries indicate to which quality model pillar the attributes are belonging.

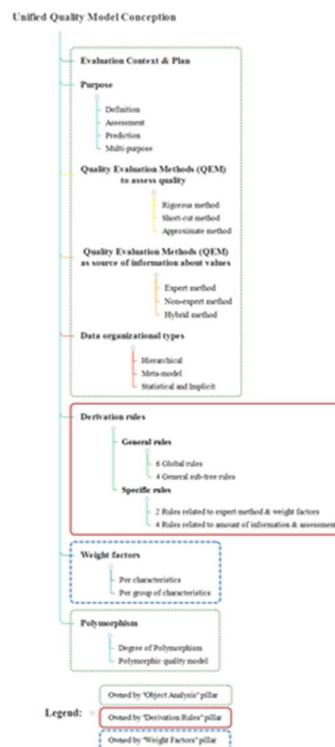


Figure 5 - The unified quality model conception, aligned with the three “Quality Model” pillars

The second architrave of our “House of Qualimetry” is the measurement. During our analysis around this topic, we noted that usually measurement processes are sequential, invent if they include sometimes some loopback to consolidate the process and data. However, we have proposed a refined process (cf Figure 6), aligned with Qualimetry, and decomposed into three main phases: initial, planning and execution.

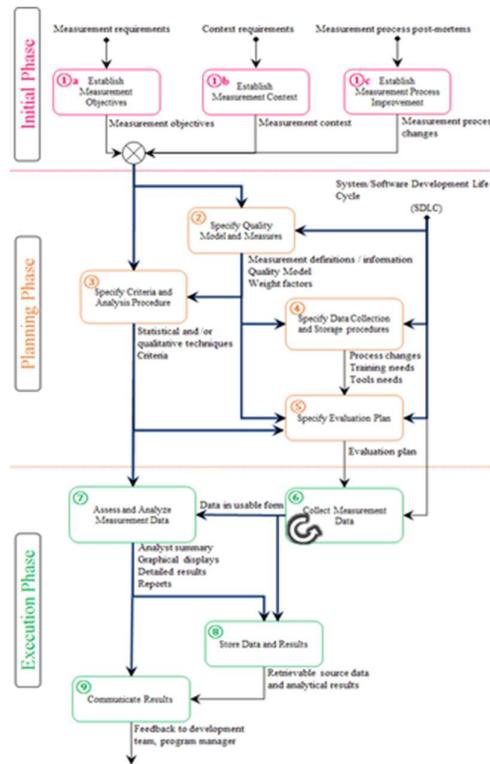


Figure 6 - Our measurement process proposal articulated over three phases

In parallel to the elaboration of that process, we noticed in the literature that a measurement can be affected by different types of transformations - scaling, rating, aggregation, threshold, validity & reliability - which can be rearranged following a natural order. Indeed, if we take a raw measurement, the first step is to assign to a specific scale, identifying which mathematical and statistical tools, for example, can be used for that raw measurement. Then we are in a position where we can state if the measurement is valid and reliable, get a rating of this measurement giving an early indication / interpretation of that measurement The Figure 7 is depicting this measurement transformation or refinement pipeline. This is our seventh contribution.

The last study we performed for this first year, was about our objects of interest within the thesis scope (ie embedded software and its organization). Therefore, we identified height main distinct objects belonging to six object categories. The Figure 8 shows not only the multiple objects of interest, but also point out that there are relationships between these objects and we have three distinct types of product: product in development/conception, product in use and product as service.

To conclude on this section, we would like to highlight that we detailed and promoted our height contributions with the achievement of three documents. First, we submitted a research paper with a scope of system engineering. This paper covers our first seventh contributions and was accepted for the incoming IEEE System Conference (SysCon) 2019 (see Annexe B). We completed another research paper at 95%. It is targeting a journal with a scope of software engineering. Our third and last document is a procedure document that we wrote to explain and describe measurement process from Figure 6 put in Automotive SPICE [35] context. This document was reviewed internally to the company welcoming the student.

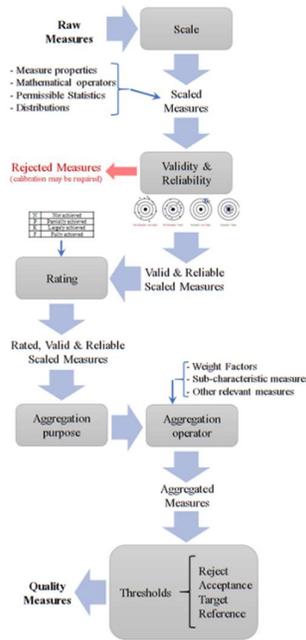


Figure 7 - The measurement refinement process

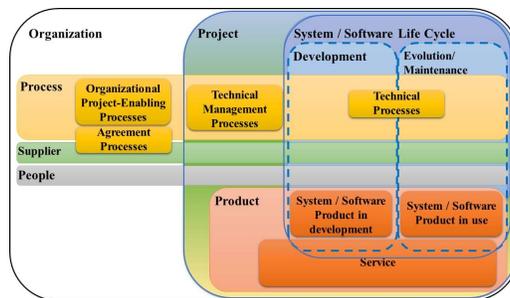


Figure 8 - Our objects of interest: organization, process, project, people, supplier, service and software product

6- Conclusion

In conclusion, this first year was quite productive in term of contributions and achievements: height new contributions to current state of the art, in which seven of them are already acknowledged by peers- and two main document achievements – IEEE conference paper and a company internal measurement procedure. In addition, we showed that our research work scope covers not only software engineering but also system engineering.

Regarding our research approach, it is a joint effort between theory and practice field using experiment loopback. This is key for us because it allows to consolidate our theory and, at the same time, ensures that our theory can be applied against practicable problems. Moreover, to foster the applicability aspect of our approach, we initiated our tool for quality model classification and decision. The first implemented functionality is to retrieve synonyms and word constellations from an online semantics atlas (cf. Annexe A).

In our next steps we are going to continue work on quality model taxonomy, relying on our contributions and extending them with ontology consideration, weight factor inclusion and introduction of quality model pedigree concept. Once completed, we will be ready to start our journey on the oracle part where context, timeline and classification achievements will help us to bring an answer.

References

- [1] Y. Argotti, C. Baron, and P. Esteban, "Quality quantification in Systems Engineering from the Qualimetry Eye," presented at the IEEE International Systems Conference (SysCon), Orlando, USA, 2019.
- [2] R. Black, *Critical Testing Processes*. Addison-Wesley, 2003.
- [3] I. Jacobson, M. Chisterson, P. Jonsson, and G. Övergaard, "Object-Oriented Software Engineering: A Use Case Driven Approach," pp. 69–70, 1992.
- [4] G. G. Azgaldov, *The Theory and Practice of Product Quality Assessment. Essentials of Qualimetry*, Moscow: Ekonomika (in Russian). 1982.
- [5] G. G. Azgaldov, "Development of the theoretical basis of qualimetry," doctoral dissertation, Kuibyshev Military Engineering Academy, Moscow (in Russian), 1981.
- [6] "ISO/IEC 9126-1:2001 - Software engineering - Product quality - Part1: Quality Model," *International Organization for Standardization*, 2001.
- [7] M. C. Paulk, B. Curtis, and M. B. Chrissis, "Capability maturity model, version 1.1," *IEEE Software*, vol. 10, no. 4, Jul. 1993.
- [8] "ISO/IEC 15504-1:2004 - Information technology – Process assessment – Part 1: Concepts and vocabulary," *International Organization for Standardization*, 2004.
- [9] 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.
- [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] B. W. Boehm, "Characteristics of Software Quality," *TRW Series of Software Technology*, Jan. 1978.
- [12] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, and M. J. Merrit, "Characteristics of Software Quality," *North Holland Publishing Co*, Jan. 1978.
- [13] R. Grady and D. Caswell, *Software Metrics: Establishing a Company-wide Program*, Prentice Hall. 1987.
- [14] M. Evans and J. Marciniak, "Software Quality Assurance and Management," *New York: John Wiley*, 1987.
- [15] M. Deutsch and R. Willis, "Software Quality Engineering: A Total Technical & Management Approach," *Prentice-Hall*, 1988.
- [16] Ghezzi, C. M. Jazayeri, and D. Mandrioli, "Fundamental of software Engineering," *Prentice-Hall, NJ, USA*, 1991.
- [17] G. Dromey, "A Model for Software Product Quality," *IEEE Transactions on Software Engineering*, vol. 146, no. 21, 1995.
- [18] IEEE Std 1061-1998, "textitIEEE Standard for a Software Quality Metrics Methodology, R2009, Revision of IEEE Std 1061." 1992.
- [19] M. Barbacci, M. H. Klein, T. A. Longstaff, and C. B. Weinstock, "Quality Attributes," Carnegie Mellon University, Technical Report (CMU/SEI-95-TR-021), ESC-TR-95-021, 1995.
- [20] L. E. Hyatt and L. H. Rosenberg, "Software Metrics Program for Risk Assessment," *Elsevier Acta Astronautica*, no. 40, pp. 223–233, 1997.
- [21] J. Bansiya and C. G. Davis, "A Hierarchical Model for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 28, no. 1, 2002.
- [22] R. Kazman, R. L. Nord, and M. Klein, "A Life-Cycle View of Architecture Analysis and Design Methods," Carnegie Mellon University, (CMU/SEI-2003-TN-026), 2003.
- [23] P. Kumar, "Aspect-oriented Software Quality Model: The AOSQ Model," *International Journal*, vol. 3, no. 2, Mar. 2012.
- [24] A. Sharma, R. Kumar, and P. S. Grover, "Estimation of quality for software components: an empirical approach," *ACM SIGSOFT Software Engineering Notes*, vol. 33, no. 6, pp. 1–10, Nov. 2008.
- [25] F. Khomh and Y. G. Guéhéneuc, "DEQUALITE: Building Design based Software quality Model," presented at the Conference on pattern Languages of Programs, 2008.
- [26] C. F. J. Lange and M. R. V. Chaudron, "Managing model quality in UML based Software Development," in *International Conference*, 2003, vol. 2.
- [27] S. K. Sehra, Y. S. Brar, and N. Kaur, "Sehra, S.K., Brar, Y.S. and Kaur, N., Soft Computing Techniques for Software Project Effort Estimation," *Int'l J. Adv. Comp. Math. Sci*, vol. 2, no. 3, pp. 160–167, 2011.
- [28] A. Tabassum, S. Nazir Bhatti Bahria, A. Rida Asghar Bahria, I. Manzoor, and A. Imtiaz, "Optimized Quality Model for Agile Development: Extreme Programming (XP) as a Case Scenario," *IJACSA International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, 2017.

- [29] M. Oriol, J. Marco, and X. Franch, "Quality models for web services: A systematic mapping," *Journal Information and Software Technology*, vol. 56, no. 10, Oct. 2014.
- [30] Y. Boukouchi, A. Marzak, H. Benlahmer, and H. Moutachaouik, "Comparative Study of Software Quality Models," *International Journal of Computer Science Issues (IJCSI)*, vol. 10 (6), no. 1, Nov. 2013.
- [31] S. Manoj Wadhwa, "A Comparative Study of Software Quality Models," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 5 (4), pp. 5634–5638, 2014.
- [32] M. Moronge Abiud and P. Mbugua, "An analytical comparative analysis of the software quality models for software quality engineering," *Comprehensive Research Journal of Management and Business Studies (CRJMBS)*, vol. 1 (2), pp. 15–24, Oct. 2016.
- [33] P. Zheng, Q. Xu, and Y. Qi, "An Advanced Methodology for Measuring and Characterizing Software Aging," in *Software Reliability IEEE 23rd International Symposium on Software Reliability Engineering Workshops*, 2012.
- [34] I. M. Umesh, G. N. Srinivasan, and M. Torquato, "Software Aging Forecasting Using Time Series Model," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 8, no. 3, pp. 589–596, Dec. 2017.
- [35] VDA QMC Working Group 13 / Automotive SIG, "Automotive SPICE Process Assessment / Reference Model., version 3.1 - revision 656." 01-Nov-2017.
- [36] "ISO 26262-6:2011 - Road vehicles - Functional safety - Part 6: Product development at the software level," *International Organization for Standardization*, 2011.
- [37] "ISO/TS 16949:2009 - Quality management systems - Particular requirements for the application of ISO 9001:2008 for automotive production and relevant service part organizations," *International Organization for Standardization*, 2009.
- [38] "GSO ISO/IEC 19761:2017, Software engineering - COSMIC: a functional size measurement method," *GCC Standardization Organization*, 2017.
- [39] G. G. Azgaldov *et al.*, "Qualimetry: the Science of Product Quality Assessment," *Standart y i kachest vo*, no. 1, 1968.
- [40] S. Wagner, *Software Product Quality Control*, Springer-Verlag Berlin Heidelberg. 2013.
- [41] M. Nei and W.-H. Li, "Mathematical model for studying genetic variation in terms of restriction endonucleases," in *Proceeding of the National Academy of Science of the USA*, 1979, vol. 76, pp. 5269–5273.

Annexes

A- First part of quality model classification and decision tool: `get_synonyms.py`

This short python program is the first piece of our tool: it makes a web request to www.atlas-semantiques.eu site in order to get a json based result of synonym and their corresponding constellations. The first text box below is an example of screen display of that python script: it looks for “functional” synonym. The second text box is the python script itself.

```
$/get_synonyms.py functional
Request word => functional

Constellations:

Lvl: 0 =>
in working order, operable, operational, operative, practical, running, serviceable,
usable, useable, useful, utilitarian, working

Lvl: 1 =>
official
```

```
#!/usr/bin/env python
import sys
import urllib, json

## active or not some debug trace
debug = 0

try:
    url = "http://www.atlas-semantiques.eu/view/synjson.php?r=" + sys.argv[1] + "&d=EN"
    ## here is an exemple of URL
    ## url = "http://www.atlas-semantiques.eu/view/synjson.php?r=functional&d=EN"
    response = urllib.urlopen(url)
    data = json.loads(response.read())
    if debug:
        print data

    ## Request: our root word
    print "Request word => " + data["request"] + "\n"

    ## Word list: Assume that we are asking only EN => we have only one item
    tab = data["words"][0]
    word_list = tab["word"]
    if debug:
        print word_list
    ## Fermeture : lists of words associated => constellations
    print "Constellations:" + "\n"
    lvl = 0
    tab = data["fermetures"]["fermeture"]
    if debug:
        print tab
    ## for each constellation, from the closest to the farrest
    for i in range(len(tab)):
        elt = tab[i]
        print "Lvl:", lvl, " => "
        words = elt["wordRef"]
        ## get all words of that constellation (concatenate for nicer display)
        res = ""
        for j in range(len(words)):
            ## we are getting only wordref, so we need to get the real value
            res += word_list[words[j]]["text"]
            if (j < (len(words)-1)):
                res += ", "

        print res + "\n"
        lvl+=1
except:
    print "Bad argument: expecting word to look-for as first parameter"
```



Quality quantification
in System Engineering

This paper is available on HAL at: <https://hal.laas.fr/hal-02010891>. Snapshots are below:

Quality quantification in Systems Engineering from the Qualimetry Eye

Yann ARGOTTI
LAAS-CNRS, Université de Toulouse
CNRS, INSA
QAPM department
Rensault Software Labs
Toulouse, France
yann.argotti@laas.fr

Claude BARON
LAAS-CNRS, Université de Toulouse
CNRS, INSA
Toulouse, France
claudbaron@laas.fr

Philippe ESTEBAN
LAAS-CNRS, Université de Toulouse
CNRS, UPS
Toulouse, France
philippe.esteban@laas.fr

Abstract—Nowadays, quality definition, assessment, control and prediction cannot easily be missed in systems engineering. One common factor among these activities is quality quantification. Therefore, throughout this paper, the authors focus on the problems relating to quality quantification in systems engineering. They first identify the main drawbacks of the current approaches adopted in this domain. They demonstrate how current solutions are not easily repeatable and adaptable across systems and how in most cases, the related standards such as ISO/IEC 25010 or Automotive-SPICE to cite just a few, are not used as they are within companies today. Fortunately, qualimetry, a young science with the purpose of quality quantification, provides the tools to resolve these gaps. To be able to use these tools, the authors propose a synthetic representation of qualimetry and its six pillars, named the “House of Qualimetry” and explain the fundamental aspects of qualimetry. They identify a set of 8 attributes to characterize the design quality model and based on these attributes, propose a new process to design or adapt the quality model. Among these attributes, a new one is introduced to capture and measure the quality model evolution and adaptation aspect: the polymorphism and the polymorphism degree. Finally, the authors consolidate the measurement part thanks to a new measurement process before returning to the benefits of these contributions to systems engineering.

Keywords—systems engineering, qualimetry, quality model, measure, polymorphism

I. INTRODUCTION

Quality quantification activity and its usage in decision making is often underestimated and failures on these activities result to non-quality which costs companies 5% total revenue [1]. Moreover, sometimes it happens that the impact related to these failures is even worse with dramatic consequences. We can refer to some well-known examples. On the 15th of April 1912, RMS Titanic sank during its maiden voyage resulting in the loss of 1,523 people [2]. This number could have been greatly reduced if the correct decision was taken during the design phase regarding the waterproof quality of compartments. Over the 1985-1987 period, Therac-25 caused massive radiation overdoses to six patients [3]. The failures were the result of issues in the design and development process. On the 28th of January 1986, the Challenger spacecraft exploded 73 seconds after its ignition, killing all seven crew members [4], with a root cause mainly associated with NASA’s company culture and its decision making processes. On the 4th of June 1996, because of an integer overflow linked to the reuse of the same navigation software than Ariane 4, Ariane 5 was self-destructed less than 40 seconds after ignition as a result [5]. Following an unaddressed major defect in airbag, the

bankrupt of Takata, an automotive equipment manufacturer, occurred [6] on the 26th of June 2017. In each all of these tragic events, root causes were either uncaught or unaddressed issue(s) in the design, architecture, product, change, decision or development process. A proper quality quantification could have identified these issues leading to corrective action and resolution before it was too late.

Quality quantification, an implicit activity associated with verification and validation processes, is particularly exercised during the quality control part of these two processes [7]. It governs not only the set of relevant quality characteristics, but also how we measure and assess them to ensure that the system that is designed and produced meets its requirements on time. In addition, it gives us the tools to evaluate how well these requirements are met from a quality perspective.

The current techniques used to quantify quality in systems engineering are usually specialized to a specific domain, adapting standards (e.g. CMMI [8] or ISO/IEC 25010 [9]) or latest research achievements. However, that approach is too centric on the object currently under design, development or production and therefore prevents to generalize and benefit immediately from advances on other areas or systems.

In 1968, a new science finally emerged that could generalize the quantification of quality: Qualimetry. This science covers both the theoretical and applied aspects of quality quantification for any domain whether it is technical or non-technical. Unfortunately, this relatively young science, which has a large scope, is not widely used even in systems engineering where we encounter only specific applied qualimetry case studies which are mostly decoupled from theoretical qualimetry. Thus, we are proposing to bring this science into systems engineering, showing the field of perspective offered by qualimetry.

In the following sections of this paper, we first review the current context and problems linked to quality quantification in systems engineering and see how qualimetry enables their resolutions. We then propose a synthesized view on qualimetry, represented by what we call the “House of Qualimetry”, that fosters its understanding, depicting quality models and measurement concepts. Next, we consolidate these two concepts of model and measurement by proposing a unified quality model conception and a new measurement process. Finally, we review the interests, with respect to systems engineering, of a qualimetry approach reinforced with our contributions versus the traditional way of quantifying quality.

XXXX-X-XXXX-XXXX-XXXX/00 ©20XX IEEE

II. CONTEXT AND PROBLEMS OF QUALITY QUANTIFICATION IN SYSTEMS ENGINEERING

We have seen from the examples in the introduction how essential it is to properly evaluate and assess the quality of a system. However, that task is more complex than simply expressing the above sentence. Indeed, it requires that we have a clear definition of: what is beneath quality, the system we aim to evaluate, the system dependencies, the way we are characterizing the system quality, how and when we are measuring these characteristics and controlling then the quality level during each step of the system life cycle, described in ISO/IEC/IEEE 15288:2015[9]. Hopefully, we have many years of work and literature upon which we can rely.

Starting with quality, while its definition evolved from the 5th century B.C. Greek philosopher's thoughts in their quest to know "what is knowledge?" [11]–[13], quality of something or someone represents the properties or characteristics of that thing or being. Nowadays, the definition is more nuanced in that quality is perceived as positive by default. However, to avoid any further debate about the meaning of quality, we are taking the definition from IEEE Standard glossary [14], which is also the one used in the International Software Testing Qualification Board glossary [15]:

"The degree to which a system, component, or process meets

- 1- Specified requirements,
- 2- Customer or user needs or expectations"

Regarding system and system life cycle, INCOSE Handbook [16] gives us the right core knowledge and foundation. Nevertheless, one important thing we have to take into consideration is that a system is a combination of three dimensions: physical, computational (or logical) and human [17]. Therefore, quantifying and then controlling the quality of a system consists of being able to characterize, measure and assess each of these dimensions and their respective combinations.

When we speak about system quality characterization, we refer to the general approach which identifies and organizes over a quality model [12], [18] the relevant characteristics of the system that we have to measure and assess in order to be able to draw a conclusion about its quality level. In 2011, ISO/IEC 25010 standard [10], an evolution and extension to systems engineering field of the previous standard ISO/IEC 9126 [19], was published and is the current reference. Moreover, ISO/IEC 25010 is itself part of the ISO/IEC 250xx standard series [9], [20]–[29] called System and Software Quality Requirements and Evaluations *aka* SQuARE. Interestingly, we are noticing that SQuARE scope does not cover System, Software and Hardware (i.e. physical dimension of a system) like ISO 26262 [30], the "Road vehicles – Functional safety" standard does.

However, despite the fact that this standard set gives us three quality models¹ with a focus on system and software, it is a weak standard that requires complements and clarifications. As B. Boehm stated in a recent *Systems Engineering Research Center* talk [31], this standard is too generic in that it attempts to fit each and every case into one

and does not consider the evolution of stakeholders needs depending on time, environment and the type of stakeholder. This statement is illustrated in the survey carried out by Wagner et al. [32] who performed a survey focused on practitioners and companies located in German-speaking countries. The result was that quality related standards² were used in less than 28% of the cases and out of this 28%, almost 79% were more or less deep adaptations of the quality model specified by these standards. So, our quality quantification dilemma becomes a question of whether it is better to create a new quality model or tailor the existing model.

Most common and early methods to design quality model are the Factor / Criteria / Metric (FCM) by McCall et al. in 1977 [33] and its generalization by Basili et al. in 1994 [34] into the Goal / Question / Metric (GQM) methods. These methods consist of quality model construction, with corresponding metrics, by answering the questions such as "what are the system quality factors and their respective criteria?" or "what are the system quality goals and their corresponding questions?". Many of the quality models have been designed with these methods, including ISO/IEC 25010 which is based on GQM. Unfortunately, these methods are missing some important aspects in the design and adaptation methodology. We can cite for instance, the integration of quality model purpose with the Definition - Assessment - Prediction (DAP) classification from Deissenboeck et al. [35], see Fig. 1. This classification depicts the incremental relationship between these three purposes, starting from definition models. Other examples can be that factors or quality characteristics can have different impact, or weight, in the overall system quality, or also, the question of "among the large number of existing quality models and factors, or quality characteristics, how to select and adapt to them to our system?".

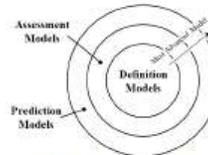


Fig 1 - The DAP classification introduced by Deissenboeck et al. [35]

Iqbal and Babar [36] were giving an approach using fuzzy logic to identify which of the ISO/IEC 25010 product quality model characteristics has to be used in their decision support system applied to "Internet Banking" case study, and then relied on Likert scale to help on the quantification aspect. On their side, Gitto et al. [37] proposed a methodology based on FCM to design complex system quality model. Unfortunately, in both cases, the focus is restricted to some specific subset of system and these authors missed qualitymetry, the science of quantification.

Qualitymetry, from the Latin *qualis* "of what kind" and the Greek *μετρον* "to measure", is the science of quality quantification. It is relatively young: its birth as a new scientific discipline occurred in 1968 [12], [38]. Its origin

¹ ISO/IEC 25010 describes product quality model, quality in use model and data quality model.

² Standards such as ISO/IEC 9126, ISO/IEC 25000, ISO 9001, CMMI

came from the need to have a generalization of quality quantification over any domains and type of object or being.

Naturally, as a science, it is composed of both theoretical and applied disciplines, but due to its youth, qualimetry requires some additions to proceed on systems engineering quality quantification. Firstly, a synthesized view is necessary to foster an understanding and capture its "pillars". We also remarked that the current general methodology approach and algorithm [12] to design quality model can be completed and unified based on the work of Wagner *et al.* [18] and achievement done on current quality model likes ISO/IEC 25010, for example.

A proper quality model is one side of the quality quantification problem. Certainly, the other side concerns the measurement of the quality characteristics and especially all measurement process activities. Additionally, and to be complete here, a qualimetry approach must integrate some missing aspects such as an evaluation plan, measurement record, analyze and reports. ISO/IEC 25040 standard [29] defines a linear evaluation process (see Fig. 2) with the same issue that we have with the rest of ISO/IEC 2500m standards: it is not precise enough and therefore requires interpretation and strong complement.

Hopefully we may rely on practical work carried out for software related decision makers by McGarry *et al.* [39] which introduces a process that includes evaluation planning, analysis techniques and measurement information models. We can also find the measurement process introduction carried out by Miller *et al.* [40], which has a scope of systems engineering and the process published by Delkiers *et al.* [41], a US-CERT team on secure software development. All these works must be merged together in order to have a consolidated measurement process. That consolidation also depends on the unified quality model conception highlighted in above paragraph and covered in a later section of this paper.

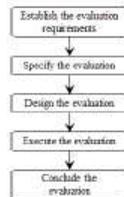


Fig. 2 - Software product quality evaluation process defined by ISO/IEC 25040 [29]

Consequently, in the following sections of this paper, we synthesize clearly the different concepts beneath qualimetry to make it more practicable. We then return to the foundation of the quality model design with a focus on qualimetry and propose the conception of a unified quality model which can be applied to any system, even if we integrated some work done within a narrow scope such as software product. We then propose an upgraded measurement process, considering unified conception dependencies and missing parts. Finally, we make a final review and draw a conclusion on the interest of our approach and next steps.

III. HOUSE OF QUALIMETRY

In order to leverage this science to a large range of audience, foster its accurate understanding and ensure that no major concepts beneath it are eluded or forgotten, we are proposing a synthesized view of the "House of Qualimetry" and its 6 pillars, depicted by Fig. 3.

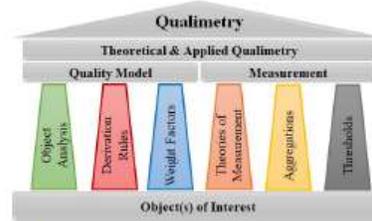


Fig. 3 - The "House of Qualimetry" and its 6 pillars.

As a science³, qualimetry relies naturally on two interlaced and complementary disciplines: theoretical [42] and applied qualimetry [43]. These two disciplines are combined into an entablature which relies on two architraves: "quality model" and "measurement". "Quality model" covers the identification, organization and representation of the relevant quality characteristics while "measure" covers the evaluation, manipulation and control of them.

Furthermore, each of these two architraves is relying on a set of three pillars, described in sub-section 1) and 2), settled on a basement reflecting the object(s) of interest (i.e. the one(s) that is (are) aimed to be quality quantified).

1) "Quality Model" pillars

While the first pillar (i.e. object analysis) is the major one, the other two are also mandatory in order to achieve the right quality model.

a) "Object analysis" pillar: This pillar gathers the necessary knowledge and activities to understand, identify and organize the relevant quality characteristic linked to the analysis of our object of interest (ie the one that its aims to have its quality quantified). Thus, we first define the purpose of our analysis, aligned with the DAP classification (see Fig. 1); we then analyze our object of interest in order to identify the quality characteristics, sub-characteristics and sub-sub-characteristics... that are relevant to us; finally we decide how we are going to organize all this data. We can note that quite often the data organization is achieved via a hierarchical structure (ie tree structure).

b) "Derivation rules" pillar: Here, the focus is with regards to global and specific qualimetry rules [12] to help optimize the design of the organizational data structure. For example, maximum tree height, division by equal characteristic, branch a tree until only simple or quasi-simple characteristics remain at its top tier.

³ We invite the reader to refer to Arzgalov *et al.* [12], for a demonstration about qualimetry as a science.

c) *"Weight factors" pillar*: Often forgotten, even in standards such as ISO/IEC 25010, the weighting factors are critical because they reflect the importance of quality characteristics among the same level of quality characteristics.

2) *"Measurement" pillars*

As was the case for the previous set of pillars, these three pillars are all mandatory in order to proceed accurately on measurement taking, even if the *"theories of measurement"* pillar represents the main one.

a) *"Theories of measurement" pillar*: This pillar is composed of three main streams of measurement theories. [44], [45]: operational measurement (i.e. how to operate / use the measure), representational measurement (i.e. how to represent the measure) and "various minor" theories. In a sense this is a fundamental pillar as it is bringing together all mathematical and statistical tools for our measurements.

b) *"Aggregations" pillar*: The aim is to deal with the way of combining (i.e. mean, median, variance and more [46]) together either all or a subset of the measurements depending on their purpose [18]. The aggregated measurements can either be weighted or un-weighted.

c) *"Thresholds" pillar*: This pillar is associated with the measure of the ability to assess, control⁴ and therefore make the correct decision. In general, man is using two types of thresholds: acceptance and target. Acceptance is often confused with the reject threshold even though they are not the same: the acceptance threshold is the worst case threshold level that may be accepted, it lies just above the best case reject level. In fact, four types of threshold exist as follows [12]: reject, accept, target and reference. Target corresponds to the threshold we are actually aiming for whereas reference corresponds to the reference value used in the industry or in the community at the time when the measurement is taken.

IV. UNIFIED QUALITY MODEL CONCEPTION

As we have seen in section III, one archetype in the *"House of Quality"* is the quality model. A quality model is an organized and multi-level representation of relevant quality characteristics for an object of interest. The multi-level aspect can be defined as the sub-sequent refinement of characteristics. For example, in ISO/IEC 25010, we have the functional suitability characteristic which is composed of three sub-characteristics: functional completeness, functional correctness and functional appropriateness.

To create such a quality model, we have identified three main streams of approaches, Azgaldov, Wagner and ISO/IEC 250nm. Azgaldov et al. [12] is representing the general quality approach while Wagner [18] is describing modeling as an iterative approach, within the software product scope, developed in the Quamoco research project [47]. Finally, ISO/IEC 250nm provides a good illustration of the work that has been done on creating other existing quality models that can be found in literature in general. We may note that there are other works that are very similar to

⁴ We can use "criteria" instead of "threshold" particularly for assessment and control, but the concept is identical and "threshold" terminology is linked to measurement.

quality model such as McGarry et al. [39] but the three above streams are a good synthesis of current distinct approaches. TABLE I provides a summarized comparison between each approach based on their scope and a list of quality model attributes. This attribute list has been elaborated by collecting for each of the three distinct approaches, all the attributes considered by their authors when designing or characterizing the quality model. Indeed, we have noted that even if most of these attributes are in common, they are not equally detailed and used. In addition, and to be complete, we propose an important new attribute, polymorphism. This attribute will be described further in this section.

TABLE I - COMPARISON OF THE MAIN THREE DISTINCT STREAMS OF WORK SUPPORTING QUALITY MODEL SPECIFICATION

Stream of approach	Wagner et al. [18], [47]	Quality models such as ISO/IEC 250nm [9], [23]-[27], [29]	Azgaldov et al. [12], [48]
Quality model scope	Project and Software product	System and Software product and in use	Any area
1 Evaluation context & plan	none	Evaluation plan	Evaluation context
2 Purposes	• Definition • Assessment • Prediction • Multi-purpose	• Definition • Assessment (evaluation part)	• Definition • Assessment
3 QEM: method to assess quality	Not specified but assumes approximate method	Not specified but assumes approximate method	• Rigorous method • Short-cut method • Approximate method
4 QEM: source of information about values in QEM	Not specified but assumes expert method	Not specified but assumes expert method	• Expert method • Not-expert method (i.e. analytical method) • Hybrid method
5 Data organizational types	• Hierarchical • Meta-model • Statistical and implicit	• Hierarchical • (* Meta-model)	• Hierarchical
6 Rules to derive trees	none	none	~30 rules
7 Weight factors	Per property / characteristic	Per property / characteristic	Per property / characteristic
8 Polymorphism	none	none	none

So, based on these raw comparison results, we first merged them into a consolidated list of attributes to consider when designing or characterizing the quality model. This merge includes: the most complete occurrence of each of these attributes. Then, in a second time, we ordered them to generate a unified quality model conception as depicted in Fig. 4: thus, to design or describe a quality model, the user must consider and use sequentially each of these attributes one by one.

a) *Evaluation Context & Plan*: Before performing any further analysis or design of a quality model, the first step is to understand what we want to achieve. For example what is the scope and what are the boundaries of this quality quantification? What are the intentions, limitations, dependencies? What audience are we targeting? What are the responsibilities, timeframe etc...? The answers to these questions provide us with our evaluation context and plan.

Without knowing them, we won't be able to design the right quality model.

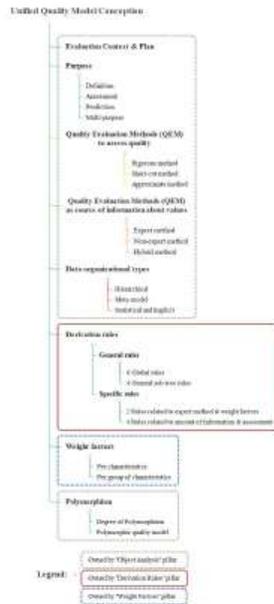


Fig. 4 - The unified quality model conception, aligned with the three "Quality Model" pillars

b) Purpose: Once we have defined the context, we are in a position to determine the intended usage we are targeting with the quality model being designed. There are three main purposes, the ones described in the DAP classification (see Fig. 1). The Definition purpose corresponds to the description using all the quality characteristics that are relevant and meaningful during quality quantification. The Assessment purpose extends the Definition purpose with corresponding metrics. The Prediction purpose is dedicated to predicting quality. In addition to these 3 categories, we can add a fourth multi-purpose category. Here the quality model is used, not only for definition but also for assessment and prediction.

c) Quality Evaluation Methods (QEM): QEM are not antinomic to FCM and GQM methods. Indeed, while these two methods do help to provide hints on how to find certain quality characteristics⁵, QEM are describing two methods linked to how the quality model is going to be designed and subsequently evaluated. The first method characterizes how

⁵ We would raise that both FCM and QCM are not method for quality model design but rather support on how to analyze our object(s) of interest.

exhaustive the analysis and quality characterization of our object of interest will be: rigorous method conducts to very detailed quality model while short-cut considers the most essential quality characteristics and therefore leads to a lighter quality model. The second method qualifies the source of information that is being used. It can be based on the findings of domain experts, non-experts or a combination of both.

d) Data organizational types: Now that we know context and plan, the purpose of quality model and the QEM to identify our quality characteristics, we have to decide how we are going to organize these data. There are three main types: hierarchical (e.g. tree), meta-model, statistical and implicit. Most of the quality models are taken from the hierarchical type.

e) Derivation rules: As we have seen in section III, these rules are guidelines that must be respected during the organization of data. They are composed of global and specific rules mainly dedicated to hierarchical type.

f) Weight Factors: This is the same concept as the one described in section III. It is fundamental and must be handled once quality characteristics are identified and organized.

g) Polymorphism: We are introducing a new and final attribute to our unified quality model conception: polymorphism. This is the same concept than we have in object-oriented programming. It reflects the capacity of a quality model to describe different types of objects as well as to link with other quality models. To complete this concept we use the nucleotide diversity formula (1) introduced by Nei and Li in 1979 [49] to measure the degree of polymorphism, or diversity, against other quality models and objects of interest.

$$n = \sum_{ij} x_i x_j n_{ij} \quad (1)$$

V. MEASUREMENT PROCESS USING QUALITY MODEL

Now that we have set a unified quality model conception to join and extend current quality model design and characterization, we can consider the quality measurement aspect and more particularly the measurement process. Indeed, the aim of a measurement process is not only to proceed on, or collect, measure but also to record and analyze the results, control quality, help on decision making, including doing some predictions and communicating the results to the right stakeholders. If we refer to the current process from ISO/IEC 25040 [29] shown in Fig. 2, we have a coarse and linear definition of the tasks that must be achieved for measurement.

So, as we indicated into section II, we are detailing and completing this process including some practical and complementary work in this field mainly carried out by McGarry *et al.* [39], Miller *et al.* [40], Dekkers *et al.* [41] and Automotive-SPICE⁶ [50]. We articulate our proposal of measurement process (cf. Fig. 5) into three sequential phases: Initial, Planning and Execution.

⁶ For this standard, we are considering MAN 6 management process linked to measurement which gives a set of guidelines that allows us to assess and exercise our proposed measurement process.

during quality quantification. This is what we call the pillars of quality model and measurement.

Moreover, we have proposed a unified design conception for quality model, applicable to any field. It shows that if we use one of the existing quality models, there are quality model attributes that are often forgotten: weight factor aspect is one such attribute, respect of derivation rules is another. Additionally, we have extended the current definition of quality model by introducing the polymorphism concept which captures the fact that quality model can cover multiple types of object and a quality model can be an instantiation of another quality model. This is important because it provides consistency over quality models.

Regarding the measure aspect of qualimetry, we addressed the gaps in current existing solutions and standards when we attempted to apply them to an internal automotive project relying on Automotive-SPICE [50] and more particularly with the MAN.6 measurement management process. A measurement process is obviously not a linear process composed of a few tasks because of its dependencies and the variety of tasks that must be performed. Our proposed process ensures not only that we are identifying each of the measurement requirements and context but also that we are integrating the use of the quality model, record and consumption (i.e. assessment, analysis, prediction, production of reports and dashboards) of measurement data as well as communication to help decision makers.

In a more concrete way and to see how applicable our approach is to real systems, we may consider the automotive field². Indeed, there exists a wide variety of car platforms (e.g. mini-compact, crossover, supercar, convertible, commercial, sport, van ...) that can be considered as variants of a vehicle. Moreover, each type of car platform is a complex system, itself composed of more than 40 systems that are distributed over more than 60 Electronic Control Units (ECU). An ECU is a compound of hardware and software; it is characterized by a set of common characteristics shared with other ECUs (e.g. diagnostic, connection interface, power), a set of specific characteristics (e.g. HMI, communication, safety) and a context (e.g. door control, engine control, telematic control, seat control). Thus, for such complex systems, including all sub-systems, the interest of our approach is that it brings homogeneity, consistency and compatibility to quality quantification. In addition, for the entire complex system - including its different systems -, our approach helps specify a joint "vocabulary", defining a derivable quality model (e.g. ECU or car platform one) and likewise allows smooth incremental change management which is key in agile development methodology.

VII. CONCLUSION

As we observed in the introduction, quantifying quality is key in order to properly assess and control system quality, as well as to provide useful support and data to decision makers. Consequently, this paper has focused on strengthening quality quantification for systems engineering, starting with the main gaps identified.

² In the same way, we could also take concrete examples from the aeronautical field.

We have seen that current quality quantification in systems engineering can be consolidated into a specific applied qualimetry case study, which is limited and often prevents replication or generalization in other systems. Moreover, and as Wagner *et al.* survey [32] highlighted, approximately 94% of companies are designing their own quality models diverging more or less from existing standards, such as ISO/IEC 25010 or A-SPICE [50] for instance. The main reason for that divergence is that those models are not sufficiently precise to fit company's needs.

Therefore, we proposed to step back to the foundation and use qualimetry, which is the science of quality quantification, to support us in filling these gaps. Thereby, our first contribution aims to clarify, leverage and foster knowledge related to qualimetry by proposing the synthetic view of the "House of Qualimetry" and its six pillars. Then, to support this synthetic view, we have elaborated its two architraves: quality model and measure.

Our second and third contributions were the identification of the height required attributes to characterize and design quality model, and the unified quality model conception (cf. Fig. 4) respectively. This unified conception is a sequential process to design, adapt or replicate quality model.

Moreover, one of these height attributes constitutes our fourth contribution. This is the polymorphism concept applied to our quality model. It captures quality model evolution, adaptation and replication aspects. We completed it with the polymorphism degree which gives us a formula to evaluate intrinsic distance between quality models.

Finally, our last contribution, our measurement process proposal, consolidated the "House of Qualimetry", not only by exploiting the "measures" architrave, but also establishing a clear link with the "quality model" architrave.

In conclusion, this paper opens a new perspective with regards to quality quantification in systems engineering thanks to qualimetry science which gives us the hindsight to fill in the identified gaps using practical solutions.

ACKNOWLEDGMENT

We would like to thank our colleagues at Renault and Renault Software Labs with regards to some fruitful discussions related to quality models and processes linked to measurement. These exchanges were carried out under the scope of an internal Renault-Nissan-Mitsubishi alliance project related to Automotive-SPICE [50].

REFERENCES

- [1] "Enquête Nationale : les Coûts de la Non-Qualité dans l'Industrie." Afnor Group (in French), Oct-2017.
- [2] K. Gordon, "Titanic Anniversary: The Myth of the Unsinkable Ship," 02-Apr-2012. [Online]. Available: <http://www.bbc.com/future/story/20120402-the-myth-of-the-unsinkable-ship>
- [3] N. G. Leveson and C. S. Turner, "An Investigation of the Thrac-25 Accidents," *Computer*, vol. 26, no. 7, pp. 18-41, 1993.
- [4] E. Ben, "Missed Warnings: The Fatal Flaws Which Doomed Challenger," *Space Safety Magazine*, 28-Jan-2014. [Online]. Available: <http://www.spacesafetymagazine.com/spacedisasters/challenger-disaster/missed-warnings-fatal-flaws-doomed-challenger/>
- [5] R. L. Baber, "The Ariane 5 explosion: a software engineer's view," *Risk*, 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] Project Management Institute, *A Guide to the Project Management Body of Knowledge - 6th edition*, Newtown Square, Pennsylvania: Project Management Institute, Inc., 2017.
- [8] M. C. Phulk, B. Curtis, and M. B. Christis, "Capability maturity model, version 1.1," *IEEE Software*, vol. 10, no. 4, Jul. 1993.
- [9] "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.
- [10] "ISO/IEC/IEEE 15288:2015 - Systems and software engineering -- System life cycle processes," *International Organization for Standardization*, 2015.
- [11] P. Artman, "From Aristotle to Descartes: A Brief history of quality." [Online]. Available: <https://blog.smartbear.com/software-quality-from-aristotle-to-descartes-a-brief-history-of-quality/>.
- [12] G. Arzgalov, A. Kostin, and A. Padilla Omiste, *The ABC of Qualimetry, toolkit for measuring the immeasurable*, Ridero, 2015.
- [13] G. P. Stavropoulos, *The Complete Aristotle*, Free GPS Library, 2013.
- [14] "IEEE Standard Glossary of Software Engineering Terminology, IEEE Std 610.12-1990," Institute of Electrical and Electronic Engineers, Inc., New York, NY, 10-Dec-1990.
- [15] "ISTQB glossary 3.1," <https://www.istqb.org/downloads/category/20-istqb-glossary.html>.
- [16] D. D. Walden, G. J. Roedler, K. J. Forsberg, D. R. Hamelin, and T. M. Shorell, *Systems Engineering Handbook: a Guide for System Life Cycle Processes and Activities*, fourth, Wiley, 2013.
- [17] A. Pryter and R. Adcock, "Report of the Workshop on the relationship between Systems Engineering and Software Engineering," Stevens Institute of Technology, Cranfield University, SERC and INCOSE, Hoboken, New Jersey, Jun. 2014.
- [18] S. Wagner, *Software Product Quality Control*, Springer-Verlag Berlin Heidelberg, 2013.
- [19] "ISO/IEC 9126-1:2001 - Software engineering - Product quality - Part 1: Quality Model," *International Organization for Standardization*, 2001.
- [20] "ISO/CEI 25001:2014 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Planning and management," *International Organization for Standardization*, 2014.
- [21] "ISO/IEC TS 25011:2017 - Information technology - Systems and software Quality Requirements and Evaluation (SQuaRE) - Service quality models," *International Organization for Standardization*, 2017.
- [22] "ISO/IEC 25012:2008 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model," *International Organization for Standardization*, 2008.
- [23] "ISO/IEC 25020:2007 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Measurement reference model and guide," *International Organization for Standardization*, 2007.
- [24] "ISO/IEC 25021:2012 - Systems and software engineering - System and software product Quality Requirements and Evaluation (SQuaRE) - Quality measure elements," *International Organization for Standardization*, 2012.
- [25] "ISO/IEC 25022:2016 - Systems and software engineering - Systems and software product Quality Requirements and Evaluation (SQuaRE) - Measurement of internal quality," *International Organization for Standardization*, 2016.
- [26] "ISO/IEC 25023:2016 - Systems and software engineering - System and software product Quality Requirements and Evaluation (SQuaRE) - Measurement of system and software product quality," *International Organization for Standardization*, 2016.
- [27] "ISO/IEC 25024:2015 - Systems and Software engineering - Systems and Software product Quality Requirements and Evaluation (SQuaRE) - Measurement of data quality," *International Organization for Standardization*, 2015.
- [28] "ISO/IEC 25030:2007 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality requirements," *International Organization for Standardization*, 2007.
- [29] "ISO/IEC 25040:2011 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Evaluation process," *International Organization for Standardization*, 2011.
- [30] "ISO 26262-6:2011 - Road vehicles - Functional safety - Part 6: Product development at the software level," *International Organization for Standardization*, 2011.
- [31] B. Boehm, "SERC TALKS: How to Query, Qualify and Quantify the Qualities Quagmire?," 08-Aug-2018.
- [32] S. Wagner, K. Lochmann, S. Winter, A. Goeb, M. Kläs, and S. Nannenmacher, "Software Quality Models in Practice: Survey Results," Technische Universität München Institut für Informatik, TUM-I9, 2012.
- [33] 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.
- [34] V. Basili, G. Caldiera, and H. D. Rombach, "Goal Question Metric Approach," *Encyclopedia of Software Engineering, John Wiley & Sons, Inc.*, pp. 528-532, 1994.
- [35] F. Deissenboeck, E. Jürgens, K. Lochmann, and S. Wagner, "Software quality models: Purposes, usage scenarios and requirements," in *Proceedings of the 7th International Workshop on Software Quality (WoSQ '09)*, 2009.
- [36] H. Iqbal and M. Babar, "An Approach for Analyzing ISO / IEC 25010 Product Quality Requirements based on Fuzzy Logic and Likert Scale for Decision Support Systems," *IJACSA International Journal of Advanced Computer Science and Applications*, vol. 7, no. 12, pp. 245-260, 2016.
- [37] J.-P. Gano, M. Bosch-Mauchand, A. P. Durupt, Z. Cherif, and I. Gauruch, "A Methodology for Complex System Quality Model Construction - First level," *ScienceDirect, IFAC-PapersOnLine*, vol. 49, no. 12, pp. 310-324, 2016.
- [38] G. G. Arzgalov et al., "Qualimetry: the Science of Product Quality Assessment," *Standartnyi iachestvo*, no. 1, 1968.
- [39] J. McGeary et al., *Practical Software Measurement: Objective Information for Decision Makers*, Addison-Wesley, 2001.
- [40] C. Miller, R. S. Carson, S. Fowler, D. J. Gutzler, and G. Roedler, "Systems Engineering Measurement Primer: A Basic Introduction to Measurement Concepts and Use for Systems Engineering," *INCOSE, San Diego, CA, Nov.*, 2010.
- [41] C. Deikens, D. Zubrow, and J. McCurley, "Measures and Measurement for Secure Software Development," 2013. [Online]. Available: <https://www.us-cert.gov/bsi/articles/best-practices/measurement/measures-and-measurement-secure-software-development>.
- [42] P. A. Florenskii, "Some Remarks on Product Quality Assessment," *Vestn. uor. ekperiment. elektrotehniki*, no. 11, 1928.
- [43] G. G. Arzgalov and A. V. Kostin, "Applied qualimetry: its origins, errors and misconceptions," *Benchmarking: An International Journal*, vol. 18, no. 3, pp. 428-444, 2011.
- [44] J. A. Diez, "A Hundred Years of Numbers: An Historical Introduction to Measurement Theory 1887-1990 Part I: The Formation Period, Two Lines of Research: Axiomatics and Real Morphisms, Scales and Invariance," *Studies in History and Philosophy of Science*, vol. 28, no. 1, pp. 167-185, 1997.
- [45] J. A. Diez, "A Hundred Years of Numbers: An Historical Introduction to Measurement Theory 1887-1990 Part II: Scales and the Mature Theory and Uniqueness Representation," *Studies in History and Philosophy of Science*, vol. 28, no. 2, pp. 237-265, 1997.
- [46] M. Detryniecki, "Fundamentals on Aggregation Operators," Computer Science Division University of California, Berkeley United States of America, 2001.
- [47] S. Wagner et al., "The Quamoco Product Quality Modelling and Assessment Approach," in *Proc. 34th International Conference on Software Engineering (ICSE'12)*, IEEE, 2012.
- [48] G. G. Arzgalov, "Development of the theoretical basis of qualimetry," doctoral dissertation, Kulobyshev Military Engineering Academy, Moscow (in Russian), 1981.
- [49] M. Nei and W.-H. Li, "Mathematical model for studying genetic variation in terms of restriction endonucleases," in *Proceeding of the National Academy of Science of the USA*, 1979, vol. 76, pp. 5269-5273.
- [50] VDA QMC Working Group 13 / Automotive SIG, "Automotive SPICE Process Assessment / Reference Model, version 3.1 - revision 656," 01-Nov-2017.