Journal of Software Engineering and Applications, 2016, 9, 128-146
Published Online April 2016 in SciRes. http://www.scirp.org/journal/jsea
http://dx.doi.org/10.4236/jsea.2016.94011

Evaluation and Improvement of an Organizational Resource Applying Strategy Patterns

Maria Fernanda Papa, Belen Rivera, Pablo Becker, Luis Olsina

GIDIS Web, School of Engineering, National University of La Pampa, General Pico, Argentina
Email: pmfer@ing.unlpam.edu.ar, riveramb@ing.unlpam.edu.ar, beckerp@ing.unlpam.edu.ar, olsinal@ing.unlpam.edu.ar

Received 19 January 2016; accepted 19 April 2016; published 22 April 2016

Abstract

For any software company that frequently performs quality assurance activities devoted to measurement, evaluation (ME) and change/improvement (MEC) projects, ME and MEC strategies can be valuable organizational assets. In this paper, we analyze the improvement of a ME strategy, which can be considered an organizational resource to be applied to quality assurance activities. This resource is called the GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*) strategy. AME/MEC strategy embraces the next three integrated capabilities: 1) the *ME/MEC domain conceptual base and framework*; 2) *the process perspective specifications*; and, 3) the *method specifications*. The improvement of GOCAME was performed instantiating two strategy patterns. A strategy pattern is a reusable solution to recurrent problems in ME/MEC projects. For an improvement goal, the selected MEC strategy pattern allows instantiating in a project a set of tailored activities and methods for measurement, evaluation, analysis and change. Particularly, we instantiate the GoME_1QV (*Goal-oriented Measurement and Evaluation for One Quality View*) strategy pattern to understand the GOCAME current quality state and compare it with the so-called GQM$^+$ Strategies. First, this evaluation and analysis allows us to know the GOCAME strengths and weaknesses with regard to the quality of the three capabilities. Second, we instantiate the GoMEC_1QV (*Goal-oriented Measurement, Evaluation and Change for One Quality View*) strategy pattern to improve the GOCAME current state, producing as result a new version of the GOCAME strategy.

Keywords

Strategy Pattern, Quality View, Evaluation, Change, Improvement

1. Introduction

Nowadays, software organizations are immersed in very competitive markets. This situation challenges organi-

How to cite this paper: Papa, M.F., Rivera, B., Becker, P. and Olsina, L. (2016) Evaluation and Improvement of an Organizational Resource Applying Strategy Patterns. *Journal of Software Engineering and Applications*, **9**, 128-146.
http://dx.doi.org/10.4236/jsea.2016.94011

zations for paying special attention to the quality of products and services offered to consumers. Moreover, organizations should take into account the selection of the best resources aimed at impacting positively into their processes, knowing beforehand that resources and processes influence the quality of their products and services.

In this direction, the ISO 25010 standard [1] states that: "*the software lifecycle processes* (*such as the quality requirements process*, *design process and testing process*) *influence the quality of the software product and the system*. *The quality of resources*, *such as human resources*, *software tools and techniques used for the process*, *influence the process quality*, *and consequently*, *influence the product quality*. *Software product quality*, *as well as the quality of other components of a system*, *influences the quality of the system*. *The system quality has various influences* (*effects*) *depending on the contexts of use*. *The context of use can be defined by a set of a user*, *a task*, *and the environment*". **Figure 1** illustrates these relationships as per ISO.

Consequently, software organizations that perform quality assurance activities should have a well-established quality evaluation and improvement approach in order to fulfill measurement, evaluation, analysis and change project goals. Project goals, particularly, information need goals may have specific purposes such as to understand the current quality state of an entity, to compare different entities for knowing their strengths and weaknesses, to improve the quality of an entity, to select the most suitable entity among a set of alternatives, etc.

Regarding the above, this article discusses the evaluation and improvement of an entity, particularly, a resource, using a *holistic quality evaluation and improvement approach* [2]. The architecture of this approach is based on two pillars, namely: 1) *a quality multi-view modeling framework*; and 2) *ME*/*MEC integrated strategies*. The first pillar specifies an ontology of quality views including terms such as entity category and quality focus and their i*nfluences* and *is determined by* relationships as shown in **Figure 1**. This ontology allow us to represent quality views, for instance, the Resource Quality View, the Process Quality View, etc. in a more formal way than in [1].

The second pillar provides a set of ME/MEC strategies, which helps software quality assurance (SQA) leaders to get data and information for analysis and decision-making in addition to achieve the information need goals for projects. Strategy is a frequently used and broad term, so for our purposes, we have defined it as: "*principles*, *patterns*, *and particular domain concepts and framework that may be specified by a set of concrete processes*, *in addition to a set of appropriate methods and tools as core resources for helping to achieve a project goal*" [3]. Furthermore, we have conceived that a strategy [3] [4] can be represented with three capabilities, namely: (1) the ME/MEC domain conceptual base and framework; (2) the process perspective specifications; and, (3) the method specifications. These three capabilities support the principle of being integrated [5] since, for instance, the same ME terms are consistently used for activities and methods.

Additionally, in the last decade we have earned experience in developing a couple of specific ME/MEC strategies. For instance, we have developed the GOCAME (*Goal-Oriented Context-Aware Measurement and Evaluation*) and SIQ in U (*Strategy for Improving Quality in Use*) strategies, in 2008 and 2010 respectively. These strategies were applied in several concrete evaluation and improvement projects such as documented in [4] [6]-[8]. For these ME/MEC projects, one or two quality views were considered. Also, both strategies have the three above-mentioned capabilities, which are supported in an integrated way.

On the other hand, we have envisioned the idea of packaging the earned experience in using ME/MEC strategies into strategy patterns. It is recognized that patterns have had and continue to have a significant impact in software and web engineering [9]-[11]. In a nutshell, the pattern's main aim is to provide a general and reusable

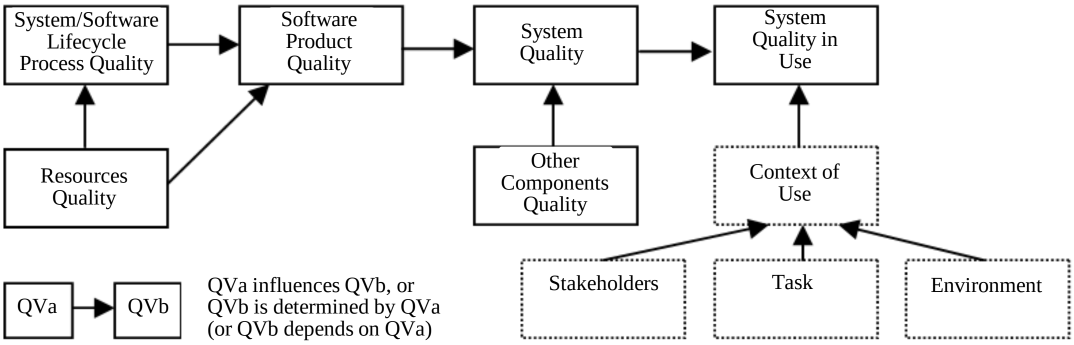


Figure 1. Target entities and their relationships for evaluating quality (adapted from [1]).

solution to a recurrent problem. We have observed that strategy patterns can be applied to recurrent ME or MEC problems/goals of any project. As a result, we specify a set of strategy patterns in a catalog that offers flexible and tailorable solutions for evaluating and improving the quality focuses for different entities in ME/MEC projects.

There are different categories of patterns documented since early 90's, for example, object-oriented design patterns [10], usability and architectural patterns [12], analysis patterns [11] [13], among others. However, strategy patterns have recently been documented in [14]. Specifically, a ME/MEC strategy pattern includes in its structure a set of generic ME/MEC process and method specifications to be instantiated for a particular purpose and context. The patterns' catalog and a potential recommender tool may support to the SQA leader in the selection process. This means that for a particular ME/MEC project goal (which embeds in its statement the quality focus and the quality views), the selection of the most appropriate strategy pattern to be instantiated can be eased.

This paper is motivated by the fact of showing the evaluation and improvement process made on the GOCAME strategy throughout the time by using strategy patterns. Thus, GOCAME was considered a resource from the entity category standpoint, with an evaluation focus on the quality of its three capabilities. Particularly, we will illustrate the employment of two strategy patterns, namely: GoME_1QV (*Goal-oriented Measurement and Evaluation for One Quality View*) and GoMEC_1QV (*Goal-oriented Measurement*, *Evaluation and Change for One Quality View*) in two periods of the timeline as depicted in **Figure 2**.

As above mentioned, we have developed the GOCAME strategy in 2008. So later, at early 2009, we have decided to understand the quality of its capabilities compared with other existing integrated strategies. To this aim, we first instantiated the GoME_1QV strategy pattern. Using this pattern allowed us to understand and analyze the strengths and weaknesses that GOCAME had in comparison with other strategies. In that case study [5], the purpose of the ME information need was to "*understand and compare*" the "*Capability Quality*" from the "*SQA leader*" user viewpoint. We have defined the Capability Quality characteristic as the "*degree to which a resource is suitable and appropriate for supporting and performing the actions when used under specified conditions*". The entity category is an "*Integrated ME Strategy*" whose super-category is "*Resource*". Also, *GOCAME* and *GQM*$^+$ *Strategies* [15] [16] were the two concrete entities to be assessed for the "*Resource Quality*" focus. This focus embraces the three required capabilities of an integrated strategy, which are represented by the "*Process Capability Quality*", "*Conceptual-Framework Capability Quality*", and "*Methodology Capability Quality*" sub-characteristics. The yielded results allowed us to understand their strengths and weaknesses. So in 2010, we saw the opportunity to plan actions for further improvements after analyzing the GOCAME weaknesses and some strengths of GQM$^+$ Strategies that could be taken.

Regarding this, we have performed two cycles of improvements and re-evaluations, from middle of 2010 to early 2014, by instantiating the GoMEC_1QV strategy pattern. Improvements were achieved by performing evaluation-driven changes in the resource, *i.e.*, in the GOCAME strategy. Therefore, the purpose of the ME information need was "*Improve*" from the same user viewpoint as before. Regarding the first improvement cycle attributes of the "*Process Capability Quality*" characteristic, which had benchmarked with low performance indicators, were changed and then re-evaluated to gauge the improvement gain. The second cycle was made as shown in **Figure 2**, where the major improvement was focused on attributes of the "*Conceptual-Framework Capability Quality*" characteristic, as we will discuss later on.

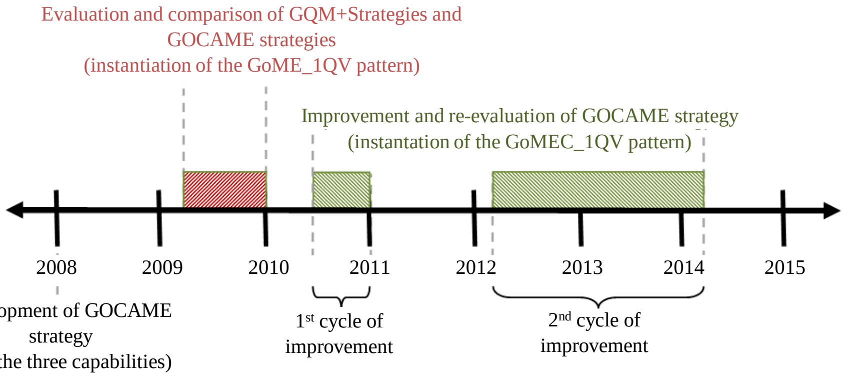

Evaluation and comparison of GQM+Strategies and GOCAME strategies (instantiation of the GoME_1QV pattern)
Improvement and re-evaluation of GOCAME strategy (instantation of the GoMEC_1QV pattern)
2008
2009
2010
2011
2012
2013
2014
2015
Development of GOCAME strategy (with the three capabilities)
1st cycle of improvement
2nd cycle of improvement

Figure 2. Timeline for the development and improvement of the GOCAME strategy.

Hence, the main contribution documented in this paper is the instantiation of two strategy patterns for the evaluation and improvement of GOCAME with regard to the quality of its three capabilities.

Following this Introduction, Section 2 describes related work addressing research that deals with holistic evaluation and improvement approaches, and strategy patterns as well. Section 3 gives a background for understanding our holistic evaluation approach. Also, it shows the relations between Resource Quality View with others quality views. Section 4 discusses ME/MEC strategy patterns in general addressing the specification of GoMEC_1QV in particular. Then, Section 5 illustrates the GOCAME case study instantiating both GoME_1QV and GoMEC_1QV strategy patterns. Finally, Section 6 draws our main conclusions and outlines future work.

2. Related Work

In this paper, we outline and exemplify a holistic quality evaluation and improvement approach whose architecture is based on two pillars, as commented above: 1) a quality multi-view modeling framework; and 2) ME/MEC integrated strategies. Additionally, a set of strategy patterns for this approach are introduced and two are instantiated with the aim of improving an organizational resource. So, we analyze below the state-of-the-art research literature with these three concerns in mind, *i.e.*, the two integrated pillars and strategy patterns.

Regarding the first pillar of the approach, there exists research that deals with quality views and quality models. But as far as we know there is no work defining and specifying an ontology of quality views, nor an explicit glossary of terms. One of the most relevant documents previously cited is the ISO 25010 standard, in which different quality views and their "influences" and "depends on" (or "is determined by") relationships are informally represented (see **Figure 1**). Also, the explicit meaning of the quality view concept is missing. Moreover, there is no clear association between a quality focus and an entity category, nor explicit definitions of the different entity categories as we do in **Table 1**. Rather, ISO 25010 outlines views in the context of a system quality lifecycle model, where some views can be evaluated by means of the quality model that the standard proposes.

Another initiative related to quality views is analyzed in [17] in which just the "influences" relationship between External Quality and Quality-in-Use (QinU) characteristics is determined by means of Bayesians networks, taking as reference the ISO 9126-1 [18] standard (this standard was superseded by [1]). However, it does not discuss a holistic quality evaluation approach that links quality views with ME/MEC strategies, as we are proposing. Finally, in [7] the 2Q2U (*internal/external Quality*, *Quality in use*, *actual Usability*, *and User experience*) quality framework is proposed. This framework extends the quality models defined in [1] adding new

Table 1. Ontology of quality views: Term definitions.

Term	Definition
Calculable Concept (synonym: Characteristic, Feature) (from ME ontology)	A characteristic that represents a combination of measurable attributes. Note 1: A *calculable concept* can be evaluated but cannot be measured as an attribute-at least in a non very trivial way such as "good" or "bad". Note 2: A characteristic can have sub-characteristics.
Calculable-Concept Focus	It is a *calculable concept* which represents the root of a *calculable-concept model*. Note 1: A *calculable-concept focus* is associated to one *entity super-category* to be evaluated.
Calculable-Concept Model (from ME ontology)	The set of *calculable concepts* and the relationships between them, which provide the basis for specifying the non-functional requirements and their further evaluation. Note 1: A possible instance of a *Calculable-Concept Model* is the ISO 25010 Quality-in-use Model.
Calculable-Concept View	Abstract relationship between one *calculable-concept focus* and one *entity super-category*. Note 1: Names of *calculable-concept views* are Quality View, Cost View, among others.
Entity Category (synonym: Object Category) (from ME ontology)	Object category that is to be characterized by measuring its attributes.
Entity Super-Category	Highest abstraction level of an entity category of value to be characterized and assessed in Software Engineering organizations. Note 1: Names of entity super-categories are *Resource*, *Process*, *Software Product*, *System*, *System in use*, among others.
Quality Focus	It is a *calculable-concept focus* for quality.
Quality View	It is a *calculable-concept view* for quality.
Resource	It is the *entity super-category* which embraces assets that can be assigned to processes, activities and tasks. Note 1: Examples of assets are Tool, Strategy, Software team, etc.
Resource Quality	It is the *quality focus* associated to the *resource* entity super-category to be evaluated.
Resource Quality View	It is the *quality view* that relates the *resource quality* focus with the *resource* entity super-category.

sub-characteristics for EQ and QinU, and considers the "influences" and "depends on" relationships for three quality views, namely: Software Product, System and System-in-Use Quality Views. But there is no explicit quality view component specified, as documented in [2] and in the next Section.

Regarding the second pillar, *i.e.*, ME and MEC integrated strategies, there exists a couple of related work. For example, *Goal Question Metric* (GQM) [19], *Continuous Quality Assessment Methodology* (CQA-Meth) [20], *Practical Software Measurement* (PSM) [21], and *Quality Improvement Paradigm* (QIP) [22]. Particularly, [15] [16] presents GQM$^+$ Strategies, which is built on top of the so-called GQM strategy. Both strategies include the principle of the three integrated capabilities of a strategy [5]. But none consider the quality views' concepts and the "influences" and "depends on" relationships, nor the ME/MEC strategy pattern idea.

Summarizing the approaches, CQA-Meth is a flexible methodology that allows the quality assessment of any software model. This methodology and its tool are part of the CQA integrated environment that can be used by companies to perform quality assessments of their own or third-party products. CQA-Meth defines the processes necessary to carry out the evaluation of UML models, and facilitate communication between the client (sponsor of the evaluation) and the evaluation team. CQA-Meth lacks an explicit conceptual framework from a terminological base standpoint. While CQA-Meth comes from the same research group who developed the FMESP (*Framework for the Modeling and Evaluation of Software Processes*) approach [23] which does have a conceptual framework with an ontological base as indicated by authors in [20], the relationship among the three capabilities is not explicit at all.

Other measurement approach widely accepted in the industry that helps manage software development projects is PSM. It is an information-oriented approach that describes a software measurement process, also being part of a comprehensive management program and software development project management. PSM describes how to define and integrate measurement requirements, collect and analyze measurement data and implement the entire measurement process in an organization. PSM was one of the sources for the development of the ISO/IEC 15939standard [24] (as indicated in http://www.psmsc.com/iso.asp), and after the formal appearance of this document, PSM was updated according to this standard as well. Another related work is the QIP paradigm, whose premise is that improvement is a continuous process. This approach helps organizations to implement a continuous improvement process taking into consideration past experiences. Therefore, QIP is beneficial in mature organizations that are aware of their learning processes and experiences. QIP uses GQM for defining goals and appropriate metrics that will guide the implementation of the process. While this paradigm is consolidated as a continuous improvement approach, it does not offer an integrated guide for measurement, evaluation and change projects. Moreover, the three capabilities of most of these strategies are not well specified in an integrated way.

Lastly, regarding the third concern, a lot of research deals with patterns. There is plenty literature about design patterns [10], analysis patterns [11] [13], architectural and usability patterns [9] [12], language patterns [25], among other categories and issues. But this literature is, to the best of our knowledge, not intended to measurement, evaluation and improvement processes and stages in which quality views and ME/MEC strategy patterns could be used accordingly. For example, authors in [9] [12] define a framework that expresses relationships between Software Architecture and Usability (SAU). This proposal consists of an integrated set of design solutions that had been identified in various industry cases. However, in our opinion, a clear separation of concerns among quality views, quality models, ME/MEC integrated strategies and strategy patterns is missing.

3. Outlining the Holistic Quality Evaluation and Improvement Approach

As indicated above, the architecture of our holistic quality evaluation and improvement approach is built on two pillars. Sub-section 3.1 discusses the first pillar, *i.e.*, the quality multi-view modeling framework, which specifies the proposed ontology of quality views and the grouping of its concepts into the quality_view component. Sub-section 3.2 analyzes, as part of the second pillar, what an integrated strategy for the purpose of evaluation and improvement is.

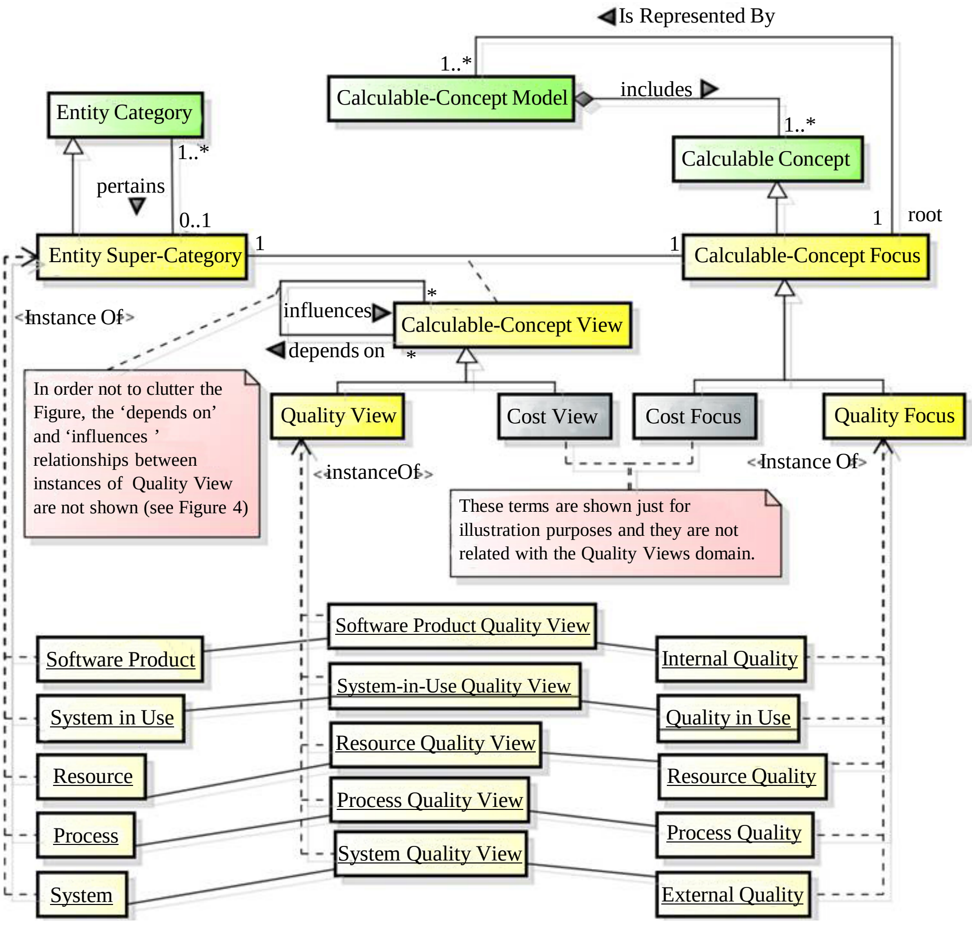
3.1. Quality Multi-View Modeling Framework

On one hand, a ME/MEC project can involve one or more entity super-categories such as Resource, Process, Software Product, System and System in Use. Each entity super-category is evaluated considering its corresponding quality focus such as Resource Quality, Process Quality, Internal Quality, External Quality and QinU.

The relationship between an entity super-category and its quality focus is called *Quality View*. For example, the Resource entity super-category and the Resource Quality focus conform the Resource Quality View. On the other hand, for each quality view an appropriate quality model must be instantiated, as part of the definition and evaluation of non-functional requirements for a ME/MEC project. A quality model has a quality focus (the root characteristic) in addition to characteristics and sub-characteristics to be evaluated which combine measurable attributes. So the quality multi-view modeling framework embraces concepts such as quality view, quality model, relationships between quality views, among other issues.

Next, we describe the *quality multi-view modeling framework* pillar considering the proposed domain ontology for quality views and the linking of the new quality_view component with the previously developed C-INCAMI (*Contextual-Information Need*, *Concept model*, *Attribute*, *Metric and Indicator*) conceptual framework [4]. C-INCAMI explicitly and formally specifies ME concepts, properties, relationships and constraints, in addition to their grouping into components. This domain ontology for ME was also enriched with terms of a process generic ontology [3].

An ontology is a way of structuring a conceptual base by specifying its terms, properties, relationships and axioms or constraints. A well-known definition of ontology says "*an ontology is an explicit specification of a conceptualization*" [26]. On the other hand, van Heijst *et al.* [27] distinguish different types of ontologies regarding the subject of the conceptualization, e.g., domain ontologies, which express conceptualizations that are intended for particular domains; and generic ontologies, which include concepts that are considered to be generic across many domains. Regarding this classification, the *quality views ontology* can be considered as a domain ontology since its terms, properties and relationships are specific to the quality area. However, some terms like entity super-category can be considered generic. **Figure 3** depicts the quality views ontology using the UML class diagram [28] for representation and communication purposes.



In order not to clutter the Figure, the 'depends on' and 'influences ' relationships between instances of Quality View are not shown (see Figure 4)

These terms are shown just for illustration purposes and they are not related with the Quality Views domain.

Figure 3. Terms and some instances of the quality views ontology.

One core term in this ontology is *Calculable-Concept View*. This term relates the *Entity Super-Category* term with the *Calculable-Concept Focus* term. An Entity Super-Category is the highest abstraction level of an *Entity Category* to be characterized for measurement and evaluation purposes. On the other hand, a Calculable-Concept Focus is a *Calculable Concept* that represents the *root* of a *Calculable-Concept Model*. In **Table 1**, some core terms of the quality views ontology are defined.

The relationship between an Entity Super-Category and its associated Quality Focus is the *Quality View* key concept in our ontology. A Quality View is a Calculable-Concept View just for quality. Instances of the Quality View term are *Software Product Quality View*, *System Quality View*, *System-in-Use Quality View*, *Resource Quality View* and *Process Quality View* terms, as shown in **Figure 3**. It is worth mentioning that in the figure not all instances of quality views are shown, as for example the Service Quality View. Additionally, its relationships are defined in **Table 2**. (Note that these two tables have updated definitions compared with [2]).

Also, **Figure 3** shows that instances of Entity Super-Category are *Software Product*, *System in Use* and *Resource*, amongst others. On the other hand, a Calculable-Concept Focus can be for example a *Quality Focus* or a *Cost Focus*. Note that Cost Focus and *Cost View* are not directly related with the quality domain, so they are gray-colored terms. Some instances of Quality Focus are *Resource Quality*, *Process Quality*, etc. **Table 1** defines Resource Quality as "*the quality focus associated to the resource entity super-category to be evaluated*". This quality focus and its instantiated quality model will be illustrated in Section 5.

Figure 4 shows the *influences* and *depends on* relationships between instances of quality views which are commonly present in development, evaluation and maintenance projects. Thus, the Resource Quality View influences the Process Quality View. For example, if a development team uses a new strategy or method—both considered as entities of the Resource Entity Super-Category-this fact impacts directly in the quality of the development process they are carrying out. Likewise, the Process Quality View influences the Software Product Quality View, and so on. Conversely, the depends on relationship has the opposite semantic.

Lastly, note that the quality views ontology shares some terms with the ME ontology presented in [4]. Particularly, an *Entity Super-Category* is an *Entity Category*, which is a term from the non-functional requirements component in **Figure 5**. Entity Category is defined in **Table 1** as "*the object category that is to be characterized by measuring its attributes*". Also, a *Calculable-Concept Focus* is a *Calculable Concept* and represents the root of a *Calculable-Concept Model*. In **Table 1** a Calculable-Concept Model is defined as "*the set of calculable concepts and the relationships between them*, *which provide the basis for specifying the root calculable-concept requirements and their further evaluation*". As a result, in **Figure 5**, the new terms are grouped into the quality_view component which are linked with the former C-INCAMI non-functional requirements component. Note also that many C-INCAMI components in **Figure 5** are drawn without terms for better visualization. In **Figure 6**, the measurement and evaluation components are expanded.

Table 2. Ontology of quality views: Relationship definitions.

Relationship	Definition
depends On	A *calculable-concept view* depends on other *calculable-concept view*.
describes	A *ME information need* describes a *calculable-concept focus*.
influences	A *calculable-concept view* influences other *calculable-concept view*.
Is Represented By	A *calculable-concept focus* can be represented by one or several *calculable-concept models*.
pertains	An *entity category* can be classified into an *entity super-category*.

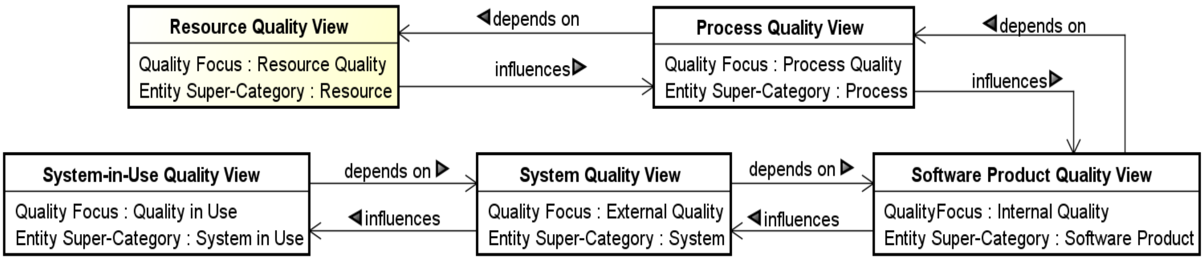

Resource Quality View
Quality Focus : Resource Quality
Entity Super-Category : Resource
◀depends on
influences▶
Process Quality View
Quality Focus : Process Quality
Entity Super-Category : Process
◀depends on
influences▶
System-in-Use Quality View
Quality Focus : Quality in Use
Entity Super-Category : System in Use
depends on ▶
◀influences
System Quality View
Quality Focus : External Quality
Entity Super-Category : System
depends on ▶
◀influences
Software Product Quality View
QualityFocus : Internal Quality
Entity Super-Category : Software Product

Figure 4. An instantiation of typical quality views in software development projects.

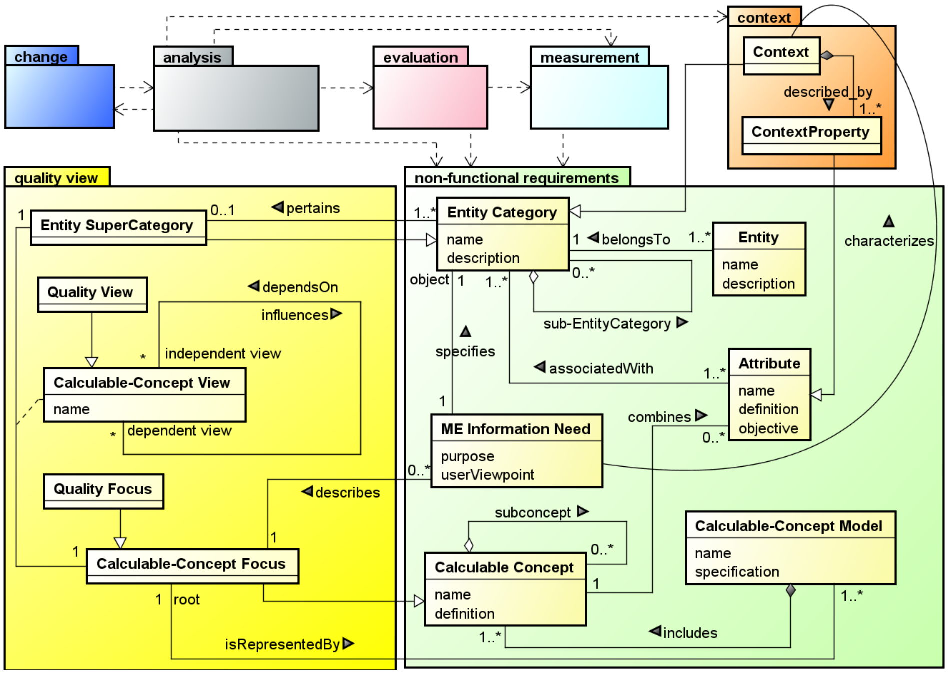


Figure 5. The quality_view component which extends the C-INCAMI conceptual framework.

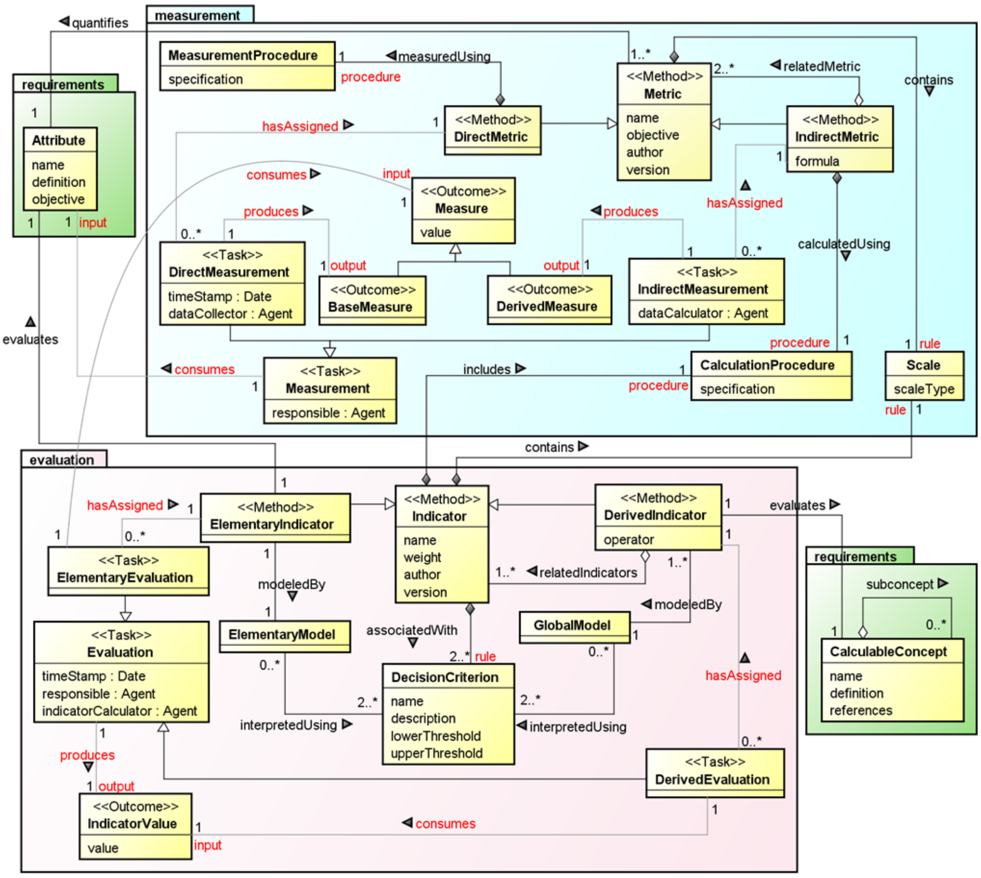


Figure 6. Measurement and evaluation components of the C-INCAMI conceptual framework enriched with process terms.

3.2. Integrated Strategies for Measurement, Evaluation and Improvement

Integrated ME/MEC strategies are the second pillar of our *holistic quality evaluation and improvement approach*. The fact of modeling quality views and their relationships is crucial for the aim of this pillar, since strategies are chosen considering quality views to be evaluated according to ME/MEC project goals.

In our approach, an integrated strategy simultaneously supports three capabilities [5]: 1) *the domain conceptual base and framework*; 2) *the process perspective specifications*; and 3) *the method specifications*. To the first capability, C-INCAMI explicitly specifies the ME/MEC terms, properties, relationships and constraints, in addition to their grouping into components. The second capability, the process specifications, usually describes a set of activities, tasks, inputs and outputs, artifacts, roles, and so forth. Besides, process specifications can consider different process perspectives such as functional, behavioral, informational and organizational [29]. Usually, process specifications primarily state what to do rather than indicate the particular methods and tools (resources) used by specific activity descriptions. The third capability provides the ability to specify methods, which ultimately represent the particular ways to perform the ME/MEC tasks.

These three capabilities support the strategy's principle of being integrated [5]. Since, for instance, the same ME terms are consistently used for activities (what) and methods (how), as we highlight in the next Section.

4. Specifying Strategy Patterns

In Software Engineering, a well-known definition for pattern is "*each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice*" [10]. That is, a pattern provides a documented and tested solution for recurring problems in similar contexts. Thus, we have observed that strategy patterns can be applied for ME/MEC projects' recurrent problems. Strategy patterns offer flexible and tailorable solutions for evaluating and improving the quality focuses for different entities in ME/MEC projects. The selection of a suitable pattern is made taking into account the goal and context of the project in conjunction with the intervening view or views, as seen in **Figure 4**.

We have specified a set of strategy patterns, following to a some extent the pattern specification template used in [10]. Our template includes the following items: 1) *name*: A descriptive and unique name, usually expressed in English; 2) *alias*: Acronym or other names for the pattern; 3) *intent*: Main objective for the pattern; 4) *motivation* (*problem*): Project problem/goal solved by the strategy pattern; 5) *applicability*: Situations in which the pattern can be applied; 6) *structure* (*solution*): Generic structure and instantiable solution that the strategy pattern offers; 7) *known uses*: References of real usage; 8) *scenario of use*: Concrete example and illustration for the instantiated pattern. In the sequel, a couple of strategy patterns are commented, describing only their intention and known uses.

GoME_1QV is a strategy pattern used to provide a solution in the instantiation of a ME strategy aimed at supporting just an understanding goal when one quality view is considered (e.g., Resource Quality View, Software Product Quality View, System Quality View, etc.). This strategy pattern must be selected when the project goal is just to understand the current situation of an entity with regard to the corresponding quality focus. The generic process of GoME_1QV consists of six activities, which are the A1-A6 gray-colored activities in **Figure 7**. This is the simplest pattern to be instantiated and the mostly used in ME projects we have run, e.g., in the evaluation of a mash-up application [7] and a shopping cart [4]. This pattern was also used to evaluate and compare GOCAME and GQM$^+$ Strategies in 2010 (recall **Figure 2**). Also, its application will be commented in sub-section 5.1.

The GoMEC_1QV strategy pattern is applied when the project goal states that it is necessary not only to understand the current situation of the entity at hand but also to perform changes on it, re-evaluate it, and gauge the improvement gain achieved. It embraces eight generic activities, *i.e.*, the A1-A8 activities in **Figure 7**. This pattern is instantiated for a MEC project goal considering just one quality view. We have illustrated GoMEC_1QV in a case study for improving the Usability of the Facebook app [14], where the Usability characteristic was linked to the External Quality focus. This pattern is thoroughly specified next, and then instantiated in sub-section 5.2 for the Resource Quality focus.

GoMEC_2QV is another strategy pattern included in the catalog, which gives a solution for an improvement project goal which involves two quality views and their relations. Recall that between two quality views, the 'influences' and 'depends on' relationships can be used. This implies that one quality view plays the role of

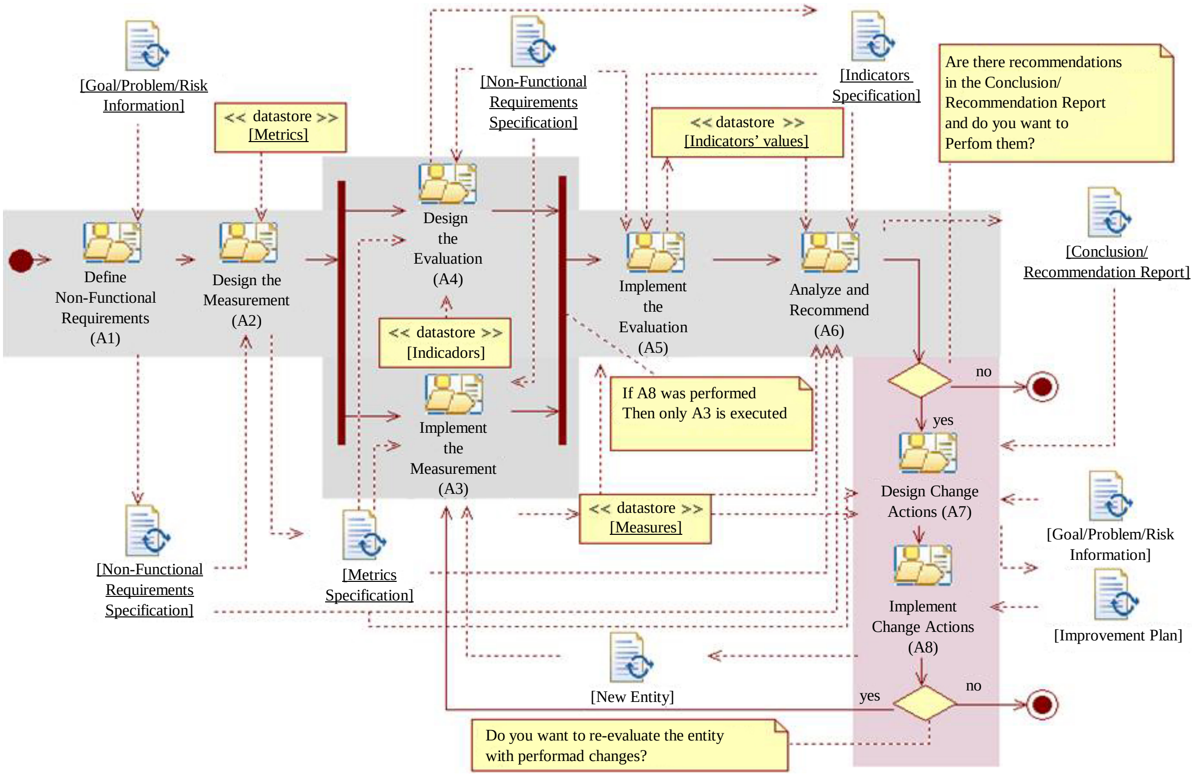

[Goal/Problem/Risk Information]
<< datastore >> [Metrics]
[Non-Functional Requirements Specification]
[Indicators Specification]
Are there recommendations in the Conclusion/ Recommendation Report and do you want to Perfom them?
<< datastore >> [Indicators' values]
Define Non-Functional Requirements (A1)
Design the Measurement (A2)
Design the Evaluation (A4)
Implement the Evaluation (A5)
Analyze and Recommend (A6)
[Conclusion/ Recommendation Report]
<< datastore >> [Indicadors]
Implement the Measurement (A3)
If A8 was performed Then only A3 is executed
no
yes
Design Change Actions (A7)
<< datastore >> [Measures]
[Goal/Problem/Risk Information]
[Non-Functional Requirements Specification]
[Metrics Specification]
Implement Change Actions (A8)
[Improvement Plan]
[New Entity]
yes
no
Do you want to re-evaluate the entity with performad changes?

Figure 7. Generic process from the functional and behavioral perspectives for the GoMEC_1QV pattern.

dependent view, while the other plays the role of *independent view* (see these roles in **Figure 5**). For example, if we consider the System Quality View and the System-in-Use Quality View, these relations embrace the hypothesis that evaluating and improving the EQ focus of a system is one means for improving the QinU focus of a system in use [1]. In turn, understanding QinU problems may provide feedback for deriving External Quality attributes that if improved could impact positively in the system quality. Furthermore, we can envision valid and interesting relationships for instance between Resource Quality View and Product Quality View. That is, by evaluating and improving the resource quality can be one means for improving the Internal Quality focus of a product. For example, changes in the development team can impact positively in the architectural design. SIQinU strategy [6] is an instance of this pattern. This strategy was used as case study in a real software testing enterprise with headquarters in Beijing, which examined JIRA (www.atlassian.com/software/jira/), a commercial software defect tracking system.

Below, the GoMEC_1QV pattern is specified considering a template with the items listed above.

GoMEC_1QV Specification

Name: Goal-oriented Measurement, Evaluation and Change for One Quality View

Alias: GoMEC_1QV or GOCAMEC_1QV in [14]

Intent: To provide a solution in the instantiation of a measurement, evaluation, analysis and change strategy aimed at supporting a specific improvement goal of a project when one quality view is considered.

Motivation (*Problem*): The purpose is to understand the current situation of a concrete entity in a specific context for a set of characteristics and attributes related to a given quality focus and then change the entity and re-evaluate it in order to gauge the improvement gain, through the systematic use of measurement, evaluation, analysis and change activities and methods.

Applicability: This pattern is applicable in MEC projects where the purpose is to understand and improve the quality focus of the evaluated entity for one quality view, such as Resource, System, System-in-Use Quality Views, among others.

Structure (*Solution*): The pattern structure is based on the three capabilities of an integrated strategy viz, the specification of the conceptual framework for the MEC domain, the specification of MEC process perspectives, and the specification of MEC methods. GoMEC_1QV provides a generic course of action that indicates which activities should be instantiated during project planning. It also provides method specifications for indicating how the activities should be performed. Specific methods can be instantiated during scheduling and execution phases of the project. Below, we describe the main structural aspects of the three strategy capabilities.

1) The concepts in the non-functional requirements, context, measurement, evaluation, change, and analysis components (**Figure 5** and **Figure 6**) are defined as sub-ontologies. The included terms, attributes and relationships belong to the MEC area. In **Figure 6** we show just the main ME terms. Note that ME terms in **Figure 6** are also enriched with terms from a generic process ontology [3] by means of stereotypes. These concepts are used consistently in the activities, artifacts, outcomes and methods of any ME/MEC strategy.

2) The process specification is made up from different perspectives, *i.e.*, *functional* which includes activities, inputs, outputs, etc.; *behavioral*, which includes parallelisms, iterations, etc.; *organizational*, which deals with agents, roles and responsibilities; and *informational*, which includes the structure and interrelationships of artifacts produced or consumed by activities. Considering the functional and behavioral perspective, **Figure 7** depicts the generic process for this pattern. The names of the eight (A1-A8) MEC activities must be customized taking into account the concrete quality focus to be evaluated.

3) The method specification indicates how the descriptions of MEC activities must be performed. **Table 3** and **Table 4** exemplify three method specification templates: the first for a direct metric used as method specification for direct measurement tasks; the second for an indirect metric, used in indirect measurements; and the third for an elementary indicator, used in elementary evaluations. Note that terms in method specification templates come from the ME conceptual base. Many other method specifications can be envisioned such as task usage log files, questionnaires, aggregation methods for derived evaluation, amongst others. For change activities traditional methods such as refactoring, re-structuring, re-parameterization, document updating, among others can be specified as well.

Known uses: GoMEC_1QV was used in a MEC project devoted to improve Usability and Information Quality attributes of a shopping cart, *i.e*, from the System Quality View through refactoring as change method [8]. Besides, this pattern was instantiated in a MEC project for the Resource Quality View [5].

5. Illustrating Two Strategy Patterns for the GOCAME Case Study

As indicated in the Introduction Section, this paper is motivated by the fact of describing the evaluation and

Table 3. Method specification templates for the measurement task: a) direct metric; b) indirect metric.

<table>
<tr><th>a) Direct Metric</th><th>b) Indirect Metric</th></tr>
<tr><td><b>Quantified Attribute name:</b></td><td><b>Quantified Attribute name:</b></td></tr>
<tr><td><b>Metric name:</b></td><td><b>Metric name:</b></td></tr>
<tr><td><b>Objective: Author: Version:</b></td><td><b>Objective: Author: Version:</b></td></tr>
<tr><td><b>Measurement Procedure:</b></td><td><b>Calculation Procedure:</b></td></tr>
<tr><td>Type: Specification:</td><td>Procedure specification: Formula:</td></tr>
<tr><td><b>Scale:</b>[Numerical|Categorical]</td><td><b>Scale:</b>[Numerical|Categorical]</td></tr>
<tr><td>Scale Type name: Value Type: Representation:</td><td>Scale Type name: Value Type: Representation:</td></tr>
<tr><td><b>Unit:</b></td><td><b>Unit:</b></td></tr>
<tr><td>Name: Description: Acronym:</td><td>Name: Description: Acronym:</td></tr>
<tr><td><b>Tool:</b> (Note: Info about the used tool if any)</td><td><b>Tool:</b> (Note: Info about the used tool if any)</td></tr>
<tr><td></td><td><b>Related Direct Metrics:</b></td></tr>
</table>

Table 4. Method specification template for the elementary evaluation: elementary indicator.

<table>
<tr><th>Elementary Indicator</th></tr>
<tr><td><b>Interpreted Attribute name: Indicator name: Author: Version:</b></td></tr>
<tr><td><b>Elementary Model:</b></td></tr>
<tr><td>Elementary Model specification:</td></tr>
<tr><td>Decision Criteria [Acceptability Levels]</td></tr>
<tr><td>Name: Range: Description:</td></tr>
<tr><td><b>Scale:</b>[Numerical|Categorical]</td></tr>
<tr><td>Scale Type name: Value Type: Representation:</td></tr>
<tr><td><b>Unit:</b> Name: Description: Acronym:</td></tr>
</table>

improvement process made on the GOCAME strategy throughout the time. To this aim, we will illustrate the employment of two strategy patterns following the timeline shown in **Figure 2**.

5.1. Applying GoME_1QV

At early 2009, we decided to understand the quality of the GOCAME strategy. To this aim, we first instantiated the GoME_1QV strategy pattern. Using this pattern allowed us to understand and analyze the strengths and weaknesses that GOCAME had in comparison with other strategies. The evaluation was focused on the *Capability Quality*. This characteristic embraces the idea of to which extent a resource is suitable and appropriate for supporting and performing the expected actions. Capability Quality includes three quality sub-characteristics, namely: *Process Capability*, *Conceptual-Framework Capability*, and *Methodology Capability*. These sub-characteristics represent the principle of integratedness of a strategy.

The selection criteria used for the choice of the entities to be evaluated were: 1) ME strategies should be documented in the literature of public domain in English language, *i.e.*, in digital libraries with recognized visibility such as IEEE Xplore, Springer Link, ACM digital library and Scopus; 2) ME strategies should have recognized impact in the academia or industry; and, 3) ME strategies should simultaneously support the three quoted capabilities. We performed a systematic literature review. At that time, there were many proposals published in the ME area having some strategies the three capabilities, but most were not in an integrated way. As a result, we included two concrete entities to be assessed: GQM$^+$ Strategies and GOCAME. In this selection process, we disregarded FMESP, CQA-Meth, among others, as discussed in [5] [30].

Since the GoME_1QV strategy pattern was applied, we instantiated for the Resource Quality View the A1-A6 generic activities that the pattern structure provides. Next we describe each instantiated activity.

(A1) Define Nonfunctional Requirements for Resource Quality is the first instantiated activity. The yielded work product is the Nonfunctional Requirements Specification for Capability Quality, which includes the Information Need Specification, the Requirements Tree Specification, and the Context Specification.

The purpose of the information need is to "*understand and compare*" the "*Capability Quality*" from the "*SQA leader*" user viewpoint. The entity category is an "*Integrated ME Strategy*" whose super-category is "*Resource*". *GQM$^+$ Strategies* and *GOCAME* are the concrete entities to be assessed. The focus of the evaluation is "*Resource Quality*".

Since in the related literature there was no official or *de facto* standard that would have specified the Capability Quality model, we had to define our own quality model. **Table 5** shows, in the 1st column, the resulting requirements tree. The 1. Capability *Quality* characteristic includes the 1.1. *Process Capability Quality*, 1.2. *Conceptual-Framework Capability Quality*, and 1.3. *Methodology Capability Quality* sub-characteristics. **Table 6** shows their definitions. Additionally, definitions of all sub-characteristics and attributes are in [30], Appendix A.1.

The *Process Capability Quality* sub-characteristic is in turn composed of sub-concepts that represent how suitable are the activities, artifacts, process model and its perspectives, and process compliance as well. The suitability of activities and artifacts are measured by means of attributes that quantify the availability, completeness, granularity and formality aspects. On the other hand, process modeling suitability (see **Table 5**) consists of sub-characteristics that evaluate how adequate are the functional, behavioral, informational and organizational process perspectives, by means of attributes of availability, completeness and granularity. Finally, the process compliance sub-characteristic is related to the adherence of strategy processes with strategy conceptual base and standards.

The *Conceptual-Framework Capability Quality* sub-characteristic is in turn composed of sub-concepts that represent how suitable are the conceptual base and framework in addition to the terminological compliance. Finally, the *Methodology Capability Quality* sub-characteristic includes attributes for evaluating availability, completeness for allocating methods to activities, and the terminological compliance as well.

(A2) Design the Measurement for Resource Quality is the instantiated activity of the GoME_1QV strategy pattern, which produces the Metric Specification work product. This activity consists of selecting the meaningful metrics from the Metric repository (*datastore* stereotype in **Figure 7**) in order to quantify each attribute of the requirements tree. A metric represents the method to perform the specified steps to quantify an attribute.

For example, the indirect metric named "*Degree of Activities Description Availability*" (DADA) was selected for quantifying the "*Activities Description Availability*" attribute (coded 1.1.1.1 in **Table 5**). **Table 7** specifies

Table 5. Full requirements tree where attributes are in italic in the 1st column. In the rest of the columns, the indicator values [%] are for the first (2010) and last (2014) evaluation of GOCAME, and the thrown differences.

Requirements Tree	2010		2014		Difference	
1. **Capability Quality** (for an integrated ME strategy)	66.48	🟨	73.87	🟨	↑	7.39
1.1. **Process Capability Quality**	58.88	🟨	81.67	🟩	↑	22.79
1.1.1. Activities Suitability	46.67	🟥	92.51	🟩	↑	45.84
1.1.1.1. *Activities Description Availability*	31.91	🟥	100	🟩	↑	68.09
1.1.1.2. *Activities Description Completeness*	15.47	🟥	92.54	🟩	↑	77.07
1.1.1.3. *Process Breakdown Structure Granularity*	70	🟨	70	🟨	↔	
1.1.1.4. *Activities Description Formality*	100	🟩	100	🟩	↔	
1.1.1.5. *Role-to-Activity Allocation Availability*	0	🟥	100	🟩	↑	100
1.1.2. Artifacts Suitability	3	🟥	19.19	🟥	↑	16.19
1.1.2.1. *Artifacts Description Availability*	0	🟥	11.11	🟥	↑	11.11
1.1.2.2. *Artifacts Description Completeness*	0	🟥	9.09	🟥	↑	9.09
1.1.2.3. *Artifacts Breakdown Structure Granularity*	30	🟥	100	🟩	↑	70
1.1.3. Process Modeling Suitability	83.56	🟩	93.52	🟩	↑	9.96
1.1.3.1. Functional Perspective Suitability	88	🟩	88	🟩	↔	
1.1.3.1.1. *Functional Perspective Availability*	100	🟩	100	🟩	↔	
1.1.3.1.2. *Functional Perspective Completeness*	100	🟩	100	🟩	↔	
1.1.3.1.3. *Functional Perspective Granularity*	70	🟨	70	🟨	↔	
1.1.3.2. Informational Perspective Suitability	82.13	🟩	100	🟩	↑	17.87
1.1.3.2.1. *Informational Perspective Availability*	100	🟩	100	🟩	↔	
1.1.3.2.2. *Informational Perspective Completeness*	90.32	🟩	100	🟩	↑	9.68
1.1.3.2.3. *Informational Perspective Granularity*	30	🟥	100	🟩	↑	70
1.1.3.3. Behavioral Perspective Suitability	88	🟩	88	🟩	↔	
1.1.3.3.1. *Behavioral Perspective Availability*	100	🟩	100	🟩	↔	
1.1.3.3.2. *Behavioral Perspective Completeness*	100	🟩	100	🟩	↔	
1.1.3.3.3. *Behavioral Perspective Granularity*	70	🟨	70	🟨	↔	
1.1.3.4. Organizational Perspective Suitability	0	🟥	100	🟩	↑	100
1.1.3.4.1. *Organizational Perspective Availability*	0	🟥	100	🟩	↑	100
1.1.3.4.2. *Organizational Perspective Completeness*	0	🟥	100	🟩	↑	100
1.1.3.4.3. *Organizational Perspective Granularity*	0	🟥	100	🟩	↑	100
1.1.4. Process Compliance	85.79	🟩	88.05	🟩	↑	2.26
1.1.4.1. *Process-to-Concept-Base Terminological Compliance*	94.74	🟩	97.56	🟩	↑	2.82
1.1.4.2. *ME Process Standards Compliance*	50	🟨	50	🟨	↔	
1.2. **Conceptual-Framework Capability Quality**	75.09	🟩	77.60	🟩	↑	2.51
1.2.1. Conceptual Framework Suitability	75	🟩	75	🟩	↔	
1.2.1.1. *Conceptual Framework Modularity*	50	🟨	50	🟨	↔	
1.2.1.2. *Conceptual Conceptual FrameworkModeling Formality*	100	🟩	100	🟩	↔	
1.2.2. Conceptual Base Suitability	68.53	🟨	69.67	🟨	↑	1.14
1.2.2.1. *Conceptual Base Completeness*	21.33	🟥	24.18	🟥	↑	2.85
1.2.2.2. *Conceptual Base Structure Richness*	100	🟩	100	🟩	↔	
1.2.3. Conceptual Framework Compliance	84.31	🟩	91.54	🟩	↑	7.23
1.2.3.1. *Framework-to-C-Base Terminological Compliance*	84.31	🟩	91.54	🟩	↑	7.23
1.3. **Methodology Capability Quality**	77.43	🟩	74.37	🟨	↓	3.06
1.3.1. Methodology Suitability	83.19	🟩	83.85	🟩	↑	0.66
1.3.1.1. *Methodology Availability*	100	🟩	100	🟩	↔	
1.3.1.2. *Method-to-Activity Completeness*	82.98	🟩	84.62	🟩	↑	1.64
1.3.1.3. *Methodology Automated Support Availability*	50	🟨	50	🟨	↔	
1.3.2. Methodology Compliance	73.68	🟨	68.29	🟨	↓	5.39
1.3.2.1. *Methodology to Conceptual Base Terminological Compliance*	73.68	🟨	68.29	🟨	↓	5.39

Table 6. Definitions of the three capability quality sub-characteristics.

Calculable Concept	Definition
Process Capability Quality (1.1)	Degree to which a process is suitable and appropriate for supporting and performing the defined actions.
Conceptual-Framework Capability Quality (1.2)	Degree to which a conceptual framework is suitable and appropriate for supporting effectively the explicit and formal specification of the main agreed concepts, properties, relationships, and constraints for a given domain.
Methodology Capability Quality (1.3)	Degree to which a methodology is suitable and appropriate for supporting and performing the process activities.

Table 7. Specification for the *Degree of Activities Description Availability* Indirect Metric.

Indirect Metric Specification
Quantified Attribute name: Activities Description Availability
Metric name: Degree of Activities Description Availability(DADA)
Objective: Quantify how many enunciated activities in the process are described with respect to the total enunciated activities.
Author: Fernanda Papa. **Version:** 0.1
Calculation Procedure:
Procedure specification: Apply the related function.
Formula:
$\text{DADA} = \begin{cases} 0, & \text{if TEA} = 0 \\ (\#\text{MDA}/\text{TEA})*100, & \text{if TEA} > 0 \end{cases}$
Scale: Numerical. **Scale Type name:** Absolute. **Value Type:** Real. **Representation:** Continuous
Unit name: Percentage **Description:** Value that represents a ratio of a whole. **Acronym:** %
Related Direct Metrics: Total number of Enunciated Activities (TEA)-Number of Minimally Described Activities (#MDA)

this metric, which adheres to the method specification template of **Table 3(b)**. Specifications of all metrics both direct and indirect are in [30], Appendix A.2.

(A3) Implement the Measurement for Resource Quality is the instantiated activity of the quoted strategy pattern. For data collection, we used the most relevant documents for both GQM$^+$ Strategies and GOCAME disregarding documents that were not co-authored by at least one member of the authors of the original research. Moreover, we gave greater priority to the most current documents when they represented a contribution with regard to previous ones. The outcome of this activity is a set of Measures. For the DADA indirect metric, the yielded value was31.91% for GOCAME, and 24.75% for GQM$^+$ Strategies. These values are calculated following the calculation procedure of **Table 7**.

(A4) Design the Evaluation for Resource Quality is the name of the instantiated activity, which produces the Indicator Specification artifact. This activity consists of defining a set of elementary indicators for attributes. An elementary indicator maps a measured value to a new numeric or categorical value in order to interpret the satisfaction level met for a given attribute. Also derived indicators are defined, which are able to interpret the satisfaction level met by characteristics and sub-characteristics. Usually, derived indicators are represented by an aggregation model.

Table 8 shows the elementary indicator specification for the 1.1.1.1 attribute following the specification template shown in **Table 4**. Additionally, specifications of all indicators are in [30], Appendix A.3.

In the case study, for the interpretation of indicator values, three acceptability levels in the percentage scale were used. A value between [0 - 50] represents an *unsatisfactory* acceptability level meaning that *change actions must be taken with high priority*. A value between [50 - 75] represents a *marginal* level meaning that *improvement actions should be taken*. While a value between [75 - 100] corresponds to the *satisfactory* level. For calculating derived indicators, the LSP (*Logic Scoring of Preference*) aggregation model [31] was used. LSP can be represented by the following equation:

$$\text{DI}(r) = \left(W_1 * I_1^r + W_2 * I_2^r + \cdots + W_m * I_m^r\right)^{1/r}$$

where DI represents the derived indicator to be calculated; I_i stands for indicator value and the following holds $0 \le I_i \le 100$ in a percentage scale; W_i represents the weights, where: $W_1 + W_2 + \ldots + W_m = 1$, and $W_i > 0$ for i = 1 to m; and, r is a LSP parameter [31].

Table 8. Specification for the *performance of activities description availability* elementary indicator.

Elementary Indicator Specification
Interpreted Attribute name: Activities Description Availability
Indicator name: Performance of Activities Description Availability (P_ADA). **Author:** Fernanda Papa. **Version:** 0.1
Elementary Model:
Elementary Model specification:
P_ADA = DADA
Decision Criteria:
f**0** ≤ P_ADA < **50**: 🔴 "Unsatisfactory level" means that change actions must be taken with high priority.
50 ≤ P_ADA < **75**: 🟡 "Marginal level" means that improvement actions should be taken.
75 ≤ P_ADA ≤ **100**: 🟢 "Satisfactory level" means a satisfactory quality of the property analyzed.
Scale: Numerical **Scale Type name:** Absolute. **Value Type:** Real **Representation:** Continuous
Unit: Name: Percentage. **Description:** Value that represents a ratio of a whole. **Acronym:** %

(A5) Implement the Evaluation for Resource Quality. From the measures obtained in the A3 activity and using the selected indicators in A4, the A5 activity was executed in 2010. **Table 9** shows a fragment of results for both evaluated strategies.

(A6) Analyze and Recommend for Resource Quality is the last activity instantiated of the GoME_1QV strategy pattern. As a result, this activity produced the Recommendation Report artifact, which includes one or more recommendations for attributes that did not meet the Satisfactory level, *i.e.*, for attributes with red-or yellow-colored indicator values. Improvement recommendations arise from GOCAME values with weak performance in addition to GQM$^+$ Strategies indicator values with stronger performance. Although the global satisfaction level achieved for GQM$^+$ Strategies (45.89%) is lower than GOCAME (66.48%), the former has some well-scored elementary indicators that can be taken into account when planning improvements for GOCAME. For example, if we compare in **Table 9** the 1.1.3.4 *Organizational Perspective Suitability* values for both strategies, the recommendation for GOCAME can take into account the GQM$^+$ Strategies strengths for this capability.

In summary, **Table 9** shows that due to the 66.48 global value for GOCAME's *Capability Quality* actions for improvement should be performed. Specifically, it can be seen that 1.1. *Process Capability Quality* is the only sub-characteristic which met a marginal value of 58.88%.Looking closer at its 1.1.1. *Activities Suitability* and 1.1.2. *Artifact Suitability* sub-characteristics, many attributes whose values are marginal or unsatisfactory must be changed.

Table 10 shows three recommendations given for the 1.1.1.1. *Activities Description Availability* attribute, which scored 31.91%. Note that the table also specifies the priority for change.

In conclusion, we have summarized the six activities belonging to the GoME_1QV strategy pattern. This case study first allowed us to understand the GOCAME strengths and weaknesses in 2010. From the issued recommendation report, we decided hereafter to perform changes for improvement. Using the GoMEC_1QV strategy pattern, we decided to perform two cycles of improvements. The first cycle for improvement was devoted to change the low benchmarked indicators for the 1.1. *Process Capability Quality*. While the second cycle for improvement was mainly devoted to change the 1.2. *Conceptual-Framework Capability Quality* by enhancing the ME ontology [3]. In the next sub-section, we describe the two new activities of the GoMEC_1QV strategy pattern by addressing the improvement cycle as a whole.

5.2. Applying GoMEC_1QV

This strategy pattern consist of a generic process with eight activities as depicted in **Figure 7**. As commented before, both strategy patterns share the A1-A6 activities. However, GoMEC_1QV has activities for planning and performing changes also being able to realize re-evaluation and improvement cycles.

In order to improve the GOCAME *Capability Quality* the A7 and A8 activities are instantiated. Now, the purpose of the A1's information need is to "*improve*" the "*Capability Quality*" from the "*SQA leader*" user viewpoint. For repeatability and comparability reasons [32], the used requirements tree, metrics and indicators were the same that in the first (2009) study. **Table 5** presents the full requirements tree in addition to the yielded elementary and derived indicator values for the GOCAME evaluations in 2010 and 2014, respectively. Also in [5] was presented the yielded elementary and derived indicator values for the first (2011) improvement cycle.

(A7) Design Change Actions for Resource Quality is the next activity specified in GoMEC_1QV that should be instantiated for change. The work product of this activity is the Improvement Plan. A fragment of this artifact is shown in **Table 11**. Basically, per each recommendation (R) in **Table 10**, a set of planned change

Table 9. Fragment of the requirements tree with the yielded indicator values in [%] performed in 2010 for both GOCAME and GQM$^+$ Strategies.

Requirements Tree	GOCAME		GQM$^+$ Strategies	
1. **Capability Quality** (for an integrated ME strategy)	66.48	🟨	45.89	🟥
1.1. **Process Capability Quality**	58.88	🟨	54.34	🟨
1.1.1. Activities Suitability	46.67	🟥	38.37	🟥
1.1.1.1. *Activities Description Availability*	31.91	🟥	24.75	🟥
1.1.1.2. *Activities Description Completeness*	15.47	🟥	14.40	🟥
1.1.1.3. *Process Breakdown Structure Granularity*	70	🟨	70	🟨
1.1.1.4. *Activities Description Formality*	100	🟩	61.39	🟨
1.1.1.5. *Role-to-Activity Allocation Availability*	0	🟥	17.82	🟥
1.1.2. Artifacts Suitability	3	🟥	31.58	🟥
1.1.2.1. *Artifacts Description Availability*	0	🟥	27.59	🟥
1.1.2.2. *Artifacts Description Completeness*	0	🟥	26.96	🟥
1.1.2.3. *Artifacts Breakdown Structure Granularity*	30	🟥	70	🟨
1.1.3. Process Modeling Suitability	83.56	🟩	70.70	🟨
1.1.3.1. *Functional Perspective Suitability*	88	🟩	76.42	🟩
1.1.3.2. *Informational Perspective Suitability*	82.13	🟩	74.97	🟨
1.1.3.3. *Behavioral Perspective Suitability*	88	🟩	60.42	🟨
1.1.3.4. *Organizational Perspective Suitability*	0	🟥	63.78	🟨
1.1.4. Process Compliance	85.79	🟩	71.11	🟨
1.2. **Conceptual-Framework Capability Quality**	75.09	🟩	35.82	🟥
1.2.1. Conceptual Framework Suitability	75	🟩	25	🟥
1.2.2. Conceptual Base Suitability	68.53	🟨	18.53	🟥
1.2.3. Conceptual Framework Compliance	84.31	🟩	81.82	🟩
1.3. **Methodology Capability Quality**	77.43	🟩	57.35	🟨
1.3.1. Methodology Suitability	83.19	🟩	51.88	🟨
1.3.2. Methodology Compliance	73.68	🟨	61.11	🟨

Table 10. Fragment of the Recommendation Report artifact generated in the instantiated A6 activity. (Note: H means High, *i.e*, an urgent action is recommended).

ID	Recommendation (R)	Attribute	Priority
R01	To ensure that all enunciated activities have a description. Note: An activity is enunciated when it has a unique name or label and belongs to the process under analysis.	1.1.1.1. *Activities Description Availability*	H
R02	To ensure that all enunciated activities are completely described. Note: An activity is completely described when it has an explicit and textual specification of objective, description, role, pre-condition, post-condition, input and output, etc.	1.1.1.2. *Activities Description Completeness*	H

Table 11. Excerpt of the Improvement Plan artifact produced in the instantiated A7 activity.

ID	Change Action (CA)	Change Action source	Method
CA01	1. Identify each activity and name it uniquely. 2. Provide a description for each enunciated activity.	Use the Measures/Measurement registry for 1.1.1.1 to find all activities without descriptions.	Document updating
CA02	1. Specify a template with the required metadata for a completely described activity. 2. Fill the template for each enunciated activity.	Use the Measures/Measurement registry for 1.1.1.2 to find the activities that are not completely described.	Template designing Document updating

actions (CA) are identified in addition to the CA source, and method type. Issues to take into account for designing the plan are for example: 1) the quality characteristics and sub-characteristic related to the capabilities of the entity to be improved; 2) the definition of attributes and metrics as well as the measured values that may help to identify the causes of the problem and their solutions; 3) the definition of indicators and their values in order to establish priorities.

(A8) Implement Change Actions for Resource Quality is devoted to implement the planned actions in the A7 activity. The produced artifact is a new version of the strategy. The last updated strategy was named GOCAME version 2. As commented above, changes were focused on the Process and Conceptual-Framework capabilities. **Table 12** summarizes these changes.

After changes were implemented, the new strategy version was re-evaluated. To this aim, A3, A5 and A6 activities were performed again. **Table 5** shows, in the 3rd column, the outcomes of the re-evaluation made in 2014. Note that results can be compared with the first GOCAME version (2nd column), since we used the same non-functional requirements, metrics and indicators specifications. The analysis activity conducted after the re-evaluation shows that the Capability Quality increased 7.39 points (4th column) reaching to a some extent the goal of improving the GOCAME strategy. This gain stems from the significant increase in the Process Capability Quality (22.79 points), and to a lesser extent, in the increase of the Conceptual-Framework Capability Quality (2.51 points).

An unexpected result was in the Methodology Capability Quality indicator, which suffered a small decrease of 3.06 points. While changes were not performed on this capability, its indicator value was impacted negatively by changes made on the Conceptual-Framework Capability. One reason is that the new terms introduced in the conceptual framework were not totally spread out in the methodology.

As the reader may surmise, new improvement cycles to the GOCAME strategy using the GoMEC_1QV pattern should be planned, since the yielded value is still 73.87, which implies opportunities for enhancements.

6. Concluding Remarks

As commented in the Introduction Section, we have developed in 2008 an integrated ME strategy so-called GOCAME which relies on three capabilities: the C-INCAMI conceptual frame work, the ME process perspective specifications, and the method specifications. At that moment, after its development, we decided to understand the quality of this resource and compare it with well-established strategies such as GQM$^+$ Strategy. In Section 4, we have thoroughly discussed a strategy pattern that is able to support evaluation-driven changes. We think that strategy patterns can be conceived as a new category of patterns, likewise analysis or design patterns are dealt as categories too.

Table 12. Main changes made to the process and conceptual framework capabilities.

Characteristic	Previous situation (2010)	Change made (2010-2014)
Process Capability (1.1)	Availability and completeness of the activities description were evaluated as unsatisfactory.	A template was designed to describe the activities and tasks of the process with the following information: Name, Purpose, Code, Process Model Perspective, Description, Inputs, Outputs, Pre-conditions, Post-conditions, Roles, and optionally Methods and Tools. All activities, sub-activities and tasks of the process were described using this template.
	The Role to Activity Allocation Availability and the Organizational Perspective Availability were evaluated as unsatisfactory.	The roles involved in the ME activities were specified and defined. Per each activity a role was assigned. The ME process was modeled from the organizational perspective.
	Availability and completeness of the artifacts description were evaluated as unsatisfactory.	A template was designed to describe the process artifacts with the following information: Name, Purpose, Description, Activity that produces it, and consumes it. Some artifacts were modeled from the informational perspective.
	The artifacts granularity was evaluated as marginal.	Artifacts of the process were analyzed and sub-divided by increasing their granularity.
Conceptual Framework Capability Quality (1.2)	The Conceptual Base Completeness was evaluated as unsatisfactory.	A process generic ontology was defined for enriching the ME domain ontology by using stereotypes.

Particularly, in Section 5, we have illustrated the instantiation of the GoME_1QV and GoMEC_1QV strategy patterns for a resource quality view. This resource is the GOCAME strategy, which can be used to support SQA activities. We have specifically analyzed the improvement of the GOCAME strategy after applying two improvement cycles, which drew a global gain of 7.39 points. Basically, the gain in the *Capability Quality* characteristic was achieved by changing some *Process Capability Quality* and *Conceptual-Framework Capability* attributes.

For the sake of concluding, a ME/MEC strategy pattern is a way of packaging general and reusable solutions for common and recurrent measurement, evaluation and change/improvement problems or situations for specific projects' goals. Moreover, according to the project goal and the amount of involved quality views a strategy pattern should be selected and then instantiated.

As future lines of research, we envision the development of a strategy pattern recommender system as a practical use of the quality views ontology (illustrated in sub-section 3.1) in the context of our holistic quality evaluation approach. The recommender system can be useful when an organization establishes ME/MEC project goals. Hence, taking into account the type of project goal and the amount of involved quality views, the strategy pattern recommender system will suggest the suitable pattern that fits that goal. Additionally, we started to work on a conceptual base that links ME information needs with business goals, in order to be able to develop strategies that support goals at different organizational levels, such as operational, tactical and strategic. It is important to remark that this multi-level goal feature was not assessed in this study, even though it was an existing feature in the GQM$^+$ Strategy approach.

Lastly, as mentioned in sub-section 3.2 an integrated strategy simultaneously supports three capabilities, being one of them the ME conceptual base and framework. We have discussed in [3] [4] [33] the rationale for introducing or adapting some terms from ISO 15939 [24]. The terminology in [24] was widespread used in the new series of SQuaRE (*Software Product Quality Requirements and Evaluation*) documents and in many works such as [20] [23] [24], to quote just a few. Considering this issue, we will review our ontology in order to evaluate the terminological adherence to SQuaRE documents.

Acknowledgements

We thank the support given by Science and Technology Agency of Argentina, in the PICT 2014-1224 project at UNLPam. Also, Belen Rivera thanks the support given by the co-funded CONICET and UNLPam grant.

References

[1] ISO/IEC 25010 (2011) Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models.

[2] Rivera, B., Becker, P. and Olsina, L. (2015) Extending the Conceptual Base for a Holistic Quality Evaluation Approach. 1*st Argentine Symposium on Ontologies and Their Applications* (*SAOA*'15), Rosario, Vol. 1449, 121-130. http://ceur-ws.org/Vol-1449/

[3] Becker, P., Papa, F. and Olsina, L. (2015) Process Ontology Specification for Enhancing the Process Compliance of a Measurement and Evaluation Strategy. *CLEI Electronic Journal*, **18**, 1-26. http://dx.doi.org/10.19153/cleiej.18.1.2

[4] Olsina, L., Papa, F. and Molina, H. (2008) How to Measure and Evaluate Web Applications in a Consistent Way. In: Rossi, G., Pastor, O., Schwabe, D. and Olsina, L., Eds., *HCIS Springer book Web Engineering*: *Modeling and Implementing Web Applications*, 385-420.

[5] Papa, M.F. (2012) Toward the Improvement of a Measurement and Evaluation Strategy from a Comparative Study. In: , Grossniklauss, M. and Wimmer, M., Eds., *LNCS* 7703, *Springer*: *Current Trends in Web Engineering*, ICWE Int'l Workshops, 189-203. http://dx.doi.org/10.1007/978-3-642-35623-0_20

[6] Lew, P., Olsina, L., Becker, P. and Zhang, L. (2012) An Integrated Strategy to Systematically Understand and Manage Quality in Use for Web Applications. *Requirements Engineering Journal*, Springer London, **17**, 299-330. http://dx.doi.org/10.1007/s00766-011-0128-x

[7] Olsina, L., Lew, P., Dieser, A. and Rivera, B. (2012) Updating Quality Models for Evaluating New Generation Web Applications. *Journal of Web Engineering*, *Special Issue*: *Quality in New Generation Web Applications*, **11**, 209-246.

[8] Olsina, L., Rossi, G., Garrido, A., Distante, D. and Canfora, G. (2008) Web Applications Refactoring and Evaluation: A Quality-Oriented Improvement Approach. *Journal of Web Engineering*, **7**, 258-280.

[9] Folmer, E. and Bosch, J. (2008) Experiences with Software Architecture Analysis of Usability. *International Journal of Information Technology and Web Engineering*, **3**, 1-29. http://dx.doi.org/10.4018/jitwe.2008100101

[10] Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) Design Patterns: Elements of Reusable Object-Oriented Software. Addisson-Wesley, Massachusetts.

[11] Fowler, M. (1997) Analysis Patterns: Reusable Object Models. Addison-Wesley, Indianapolis.

[12] Folmer, E., van Gurp, J. and Bosch, J. (2003) A Framework for Capturing the Relationship between Usability and Software Architecture. *Software Process*: *Improvement and Practice*, **8**, 67-87. http://dx.doi.org/10.1002/spip.171

[13] Hamza, H. and Fayad, M.E. (2004) Applying Analysis Patterns through Analogy: Problems and Solutions. *Journal of Object Technology*, **3**, 197-208. http://dx.doi.org/10.5381/jot.2004.3.4.a11

[14] Rivera, B., Becker, P., Lew, P. and Olsina, L. (2015) Strategy Patterns for Evaluating and Improving Usability. *Proceeding of Argentine Symposium on Software Engineering* (ASSE'15), 44 JAIIO, Rosario, 69-84.

[15] Basili, V.R., Heidrich, J., Lindvall, M., Münch, J., Regardie, M., Rombach, H.D., Seaman, C.B. and Trendowicz, A. (2007) GQM$^+$Strategies: A Comprehensive Methodology for Aligning Business Strategies with Software. MetriKon, Kaiserslautern, 1-14.

[16] Basili, V., Lindvall, M., Regardie, M., Seaman, C., Heidrich, J., Jurgen, M., Rombach, D. and Trendowicz, A. (2010) Linking Software Development and Business Strategy through Measurement. *Computer*, **43**, 57-65. http://dx.doi.org/10.1109/MC.2010.108

[17] Moraga, M.A., Bertoa, M.F., Morcillo, M.C., Calero, C. and Vallecillo, A. (2008) Evaluating Quality-in-Use Using Bayesian Networks. *QAOOSE* 2008, Paphos, 7-11 July 2008, 1-10.

[18] ISO/IEC 9126-1 (2001) Software Engineering Product Quality—Part 1: Quality Model.

[19] Basili, R., Caldiera, G. and Rombach, H.D. (1994) The Goal Question Metric Approach. *Encyclopedia of Software Engineering*, **1**, 528-532.

[20] Rodríguez, M., Genero, M., Torre, D., Blasco, B. and Piattini, M. (2010) A Methodology for Continuos Quality Assessment of Software Artefacts. 10*th International Conference on Quality Software* (*QSIC* 2010), Zhangjiajie, 14-15 July 2010, 254-261. http://dx.doi.org/10.1109/QSIC.2010.24

[21] McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J. and Hall, F. (2001) Practical Software Measurement: Objective Information for Decision Makers. Addison-Wesley Professional, Boston.

[22] Basili, V. (1989) Software Development: A Paradigm for the Future. *Proceedings of the* 13*th Annual International Computer Software & Applications Conference* (*COMPSAC*), Orlando, 20-22 September 1989, 471-485. http://dx.doi.org/10.1109/cmpsac.1989.65127

[23] Mora, B., García, F., Ruiz, F., Piattini, M., Boronat, A., Gómez, A., Carsí, J.A. and Ramos, I. (2008) Software Generic Measurement Framework Based on MDA. *IEEE Latin America Transactions*, **6**, 363-370. http://dx.doi.org/10.1109/TLA.2008.4815290

[24] ISO/IEC 15939 (2007) Systems and Software Engineering—Measurement Process. 2nd Edition.

[25] Barišić, A., Monteiro, P., Amaral, V., Goulão, M. and Monteiro, M. (2012) Patterns for Evaluating Usability of Domain-Specific Languages. 19*th Conference on Pattern Languages of Programs* (PLoP), Tucson, 19-21 October 2012, Article No.: 14.

[26] Gruber, T.R. (1993) A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, **5**, 199-220. http://dx.doi.org/10.1006/knac.1993.1008

[27] van Heijst, G., Schreiber, A.T. and Wielinga, B.J. (1997) Using Explicit Ontologies in KBS Development. *International Journal of Human-Computer Studies*, **46**, 183-292. http://dx.doi.org/10.1006/ijhc.1996.0090

[28] OMG-UML (2005) Unified Modeling Language Specification, Version 2.0.

[29] Curtis, B., Kellner, M. and Over, J. (1992) Process Modelling. *Communications of the ACM*, **35**, 75-90. http://dx.doi.org/10.1145/130994.130998

[30] Papa, M.F. (2015) Quality Assurance of an Organizational Resource: Evaluating and Improving an Integrated Measurement and Evaluation Strategy. PhD Thesis, School of Informatics, National University of La Plata, Buenos Aires. (In Spanish) http://hdl.handle.net/10915/44792

[31] Dujmovic, J.J. (1996) A Method for Evaluation and Selection of Complex Hardware and Software Systems. *Proceedings of the* 22*nd International Conference for the Resource Management and Performance Evaluation of Enterprise CS*, *CMG* 96, **1**, 368-378.

[32] Kitchenham, B., Hughes, R. and Linkman, S. (2001) Modeling Software Measurement Data. *IEEE Transactions on Software Engineering*, **27**, 788-804. http://dx.doi.org/10.1109/32.950316

[33] Olsina, L. and Martin, M. (2004) Ontology for Software Metrics and Indicators. *Journal of Web Engineering*, **2**, 262-281.

[34] Abran, A. (2010) Software Metrics and Software Metrology. Wiley-IEEE Computer Society Press, Hoboken. http://dx.doi.org/10.1002/9780470606834