View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Servicio de Difusión de la Creación Intelectual

# Quality Evaluation in Agile Process: A First Approach

Noelia Pinto[1], César J. Acuña[1], Liliana Cuenca Pletsch[1]

[1]Grupo de Investigación en Ingeniería y Calidad del Software (GICS), Facultad Regional Resistencia, Universidad Tecnológica Nacional, French 414, Resistencia, Chaco, Argentina
{ns.pinto, csr.acn, lilianacp]@gmail.com

**Abstract.** In recent years, it has been given much importance to the use of models and quality standards on software development processes. Because these are those that facilitate continuous improvement and enable companies to provide higher quality products to its customers by increasing their competitive level.

Today, software development is based on agile processes that allow production characterized by its changing requirements and the need for continuous customer deliveries environments. Thus it is imperative to provide companies with tools for assessing the quality of these cycles agile processes.

QuAM is presented in this article, an approach to design a model of quality, integrated and flexible, that assesses the quality development cycles based on the principles and practices of the agile approach.

**Keywords:** Agility, Quality Software, Software Engineering, Agile methodologies.

## 1 Introduction

The final product quality, low cost and timely deliveries become key elements for the benefit of domestic sales and international projection of the Software Industry. In this sense, and in order to increase the quality and capability of their processes and, consequently, the quality of its products and services, software process improvement [1] becomes the differentiating element that companies in the sector need to increase their competitiveness.

In the case of Argentina, Software Industry consists mainly of PYMES, companies that represent 80% of the sector, according to the latest report of the Permanent Observatory of Software Industry and Information Services (OPSSI) [2]. Thus, taking into account this reality, it is important to note that several authors [3] [4] [5] agree on the difficulty that means for PYMES to implement programs of Software Process Improvement (SPI) mainly due to the lack of monitoring action plans and implementation due to the high cost involved. Thus, the parameters of development time and cost of solutions will directly affect the work done, resulting the quality as the first variable adjustment available.

However it is not correct to consider the study or the impact of those elements that associate the quality of the final product but is also necessary to adjust the parameters associated with the processes that have facilitated obtaining it. Depending on this, there are numerous methodological proposals that guide the software development cycle and affecting different dimensions of the process. The more traditional methodologies are especially focused on a rigorous definition of roles, the activities involved, artifacts to be produced, and the tools and notations that will be used [6].

But these approaches are not the most suitable for many projects related to current scenarios where the system environment is changing and where it is demanded to drastically reduce development time without neglecting high levels of quality. So agile methodologies pursue principles such as incremental delivery of new functionality to the client, which are prioritized according to business value added (so the software product evolves in different deliveries), favoring the continuous improvement and emphasis on close collaboration between the development team and business experts [7].

Previously it has been presented [8] a study on PYMES of Software companies in the NEA, in which the adoption of agile methodologies such as life cycle in their development processes was analyzed. From there the objective of this work is clear, that is to present QuAM, one first approach to the design of a model that allows quality assessment of agile processes, contemplating two perspectives: Process and Product. The article is structured as follows: in section 2 the state of the art is described through the presentation of related work. Then, in section 3, the characteristics of the proposed model by the design quality of experience for the same validation are presented. And finally, conclusions and future work are to be developed from this line of work are exposed.

## 2 Related Work

There are several models in the literature to assess the quality of software, quality trying break into a category of simpler characteristics and from two perspectives: the product and the process.

Among quality models that enable evaluation of the software product, the Model Mc Call, created by Jim Mc Call in 1977 [9] stands out. This defines 3 perspectives (Operability Product, Product Review and Transition Product) for the analysis of the quality of software, together with associated factors and criteria. The proposed metrics are questions that apply a numerical weighting to a particular software product attribute. After obtaining the values for all metrics specific criteria, the average of these is the value for that criterion. Another model worth mentioning is FURPS [10], developed by Hewlett-Packard in 1987, in which a set of quality factors of software are described: Functionality (Functionality) Usability (Usability), Reliability (Reliability), Performance (Performance) and capacity support (Supportability). These elements can be used to establish quality metrics for all software process activities.

Among the international quality standards associated with the most relevant software is ISO / IEC 9126 [11], based on a hierarchical model with three levels: Features, Sub-characteristics and Metrics. The first level has six characteristics: Functionality, Reliability, Efficiency, Maintainability, Portability and Ease of Use. These characteristics (factors) are composed in turn by sub-characteristics (sub- factors) related to external quality, and sub-characteristics related to the internal quality.

There are also quality models that evaluate processes for obtaining software product. One of them, based on agile methodologies, is the AGIS (combination of AGILE and ISO) [12] which establishes a mechanism for measuring the degree of agility of software development processes. The ISO model supplemented with 10 dimensions; this configuration is oriented to measure the degree of implementation of the values of the agile manifesto [13] in the areas of engineering knowledge. AGIS aims to meet two needs: one focuses on companies, since this model can achieve differentiation from other companies that have only certified quality through ISO 9001: 2008. Furthermore, AGIS provides a report with suggestions for improvement based on the assessment of the size proposed evaluate.

Another model, similar to the above, is the AGIT (Agile Software Development) [14] which suggests that the best performance is achieved when the goals of all stakeholders are met. This requires an approach considering the views of different stakeholders, for which appropriate indicators are defined for each. AGIT considers four different views for stakeholders: the IT Manager is the actor concerned with traditional aspects of performance considering SW development time, cost and quality; the second actor is represented by team members whose goal is "job satisfaction"; the Scrum Master whose main goal is the "efficient resolution of impediments".

Finally, the main objective seeking customers, the fourth stakeholder is its own satisfaction. This model suggests evaluate the quality of development processes considering the views of the different stakeholders involved, describing the indicators that are appropriate to each of these profiles.

On the other hand, it is also available the  COBIT Model(Control Objectives for Information and Related Technology) [15], a tool that represents a particular collection of documents which can be classified, generally accepted as the best practices for the management, control and IT security. In COBIT, these domains is called: Plan and Organization (PO); Acquisition and implementation (AI); Delivery and Support (DS); Monitoring and Evaluation (ME). Through these four domains, COBIT has identified 34 IT processes. Through these four domains, COBIT has identified 34 IT processes and for each of these domains it defines goals and metrics to determine and measure their results and performances, based on the principles of balanced scorecard business (BSC).

Finally, among the models that apply to processes of software development it is important to emphasize the CMMI (Capability Maturity Model Integration) [16], a model for the improvement and evaluation of processes for development, maintenance and operation of software systems. CMMI has four disciplines to choose from: Systems Engineering (SE), Software Engineering (SW), Processes and Products Development (IPPD) and Distribution (SS). The model itself has two representations. One of them is the staged representation in which it is centered a set of process areas, which are organized by maturity levels (1-5), while in the continuous representation, each process area are classified in terms of capacity levels (0-5). CMMI and agile methods have also been compared in several studies [17] [18] [19], for example, Paulk [20] suggests that the use of stories XP, at the customer's premises and continuous integration can comply with the requirements management objectives CMMI-SW. Moreover, in his study, Turner and Jain [21] determined that several of the components of CMMI and agile methods were in conflict, most of them related to organizational processes.

Among the presented models, it is observed that there is no proposal which allows quality assessment of agile processes themselves. Therefore, it is presented below the QuAM design, an approach which aims to provide a method of evaluation to determine the quality of software development processes based on Agile and its resulting products.

# 3 QuAM: Quality Evaluation Model of Agile processes

## 3.1 Design of the proposal

QuAM defines a scheme of components to set up a quality assessment model that provides an objective measure of the quality of the agile process implemented in any given project, allowing to obtain the agile profile associated with it. In this first instance, the model structure is presented taking into account only the dimension to the process level. Thus, the proposal that has been called Quam Level 1 provides a metric tree (Mi,i=1... 4) composed by measurable attributes (Ai) through a series of criteria with associated measures. Then, according to the scope of this article, the dimension of the quality model presented here will evaluate the following components based on:

- Metrics 1 - Election of Life Cycle: The life cycle of a software project defines the order of the process activities. Quam will consider better Iterative life cycles and incremental over others. Focus will be placed on the implementation of the process, not in the documentation generated. The attributes and criteria to be evaluated are presented in Table 1.

**Table 1.** Attributes and Metrics Criteria: Election of Life Cycle

| *Positive Attributes* | | *Negative Attributes* | |
|---|---|---|---|
| *A1.1 - Value to Iterative and Incremental Life Cycle* | | *A1.2 - Value to Waterfall Life Cycle* | |
| Not complete iterations are performed, but new features are added to the product | 0 | The project is divided into strictly sequential steps. | -2 |

**Table 1 (cont.).** Attributes and Metrics Criteria: Election of Life Cycle

| | | | |
|---|---|---|---|
| In each iteration, the product is revised and improved through refactoring. | 1 | Phases are executed simultaneously. | -1 |
| At each iteration, not only the functionality are improved, but also new are added to the product. | 2 | At the end of each phase, it is possible to make a backtracking and improve the defined in the previous stage. | 0 |

- Metrics 2 - Assessment of Team: The human component of the project to assess must have adequate skills to the agile philosophy, and the company must have the means to achieve it. For QuAM will be important to evaluate the flow of communication between team members and the ability to face the same agile practices. The attributes and associated criteria to this metric are presented in Table 2.

**Table 2.** Attributes and criteria associated with Metric: Assessment of Team

| *Positive Attributes* | | *Negative Attributes* | |
|---|---|---|---|
| *A2.1 - Value to team meetings* | | *A2.2 - Value of compliance schedule* | |
| No meetings are held in all iterations. | 0 | The schedule is adapted according to the changes and needs that arise throughout the project. | -3 |
| In each iteration a meeting is done virtually at least. | 1 | Control milestones are set out in the schedule and changes can be defined on the scheduled dates. | -1 |
| In each iteration, at least one meeting is held with the physical presence of the entire team. | 3 | The schedule established by stages is strict and does not allow changes. | 0 |
| *A2.3 - Value to the definition of roles.* | | *A2.4 - Value to the process by over the team.* | |
| No roles are defined for individuals. | 0 | Activities, deliverables and development and management tools are defined for the project. | -3 |
| A clear definition of roles is performed on individuals team. | 1 | Activities and project deliverables are defined. | -2 |
| A clear definition of roles and responsibilities is done between team members. | 2 | Activities for each iteration are defined in the project. | -1 |

**Table 2 (cont.).** Attributes and criteria associated with Metric: Assessment of Team

| | | | |
|---|---|---|---|
| A clear definition of roles, responsibilities and interaction between team members are made. | 3 | Activities for the project are defined but not at the level of each iteration | 0 |

- Metric 3 - Production capacity of deliverables: QuAM will evaluate the frequency with which the project produces deliverables versions from the product to the customer. In this component, it will take into account the compliance with the lead time and the validity of each deliverable, favoring those projects whose validation has been automated. The change management process will be also measured the change management process will be also measured about the product and the verification process implementation and validation of them. In Table 3, the attributes and criteria that are considered for this metric are included.

**Table 3.** Attributes and criteria associated with Metric: Production capacity of deliverables

| *Positive Attributes* | | *Negative Attributes* | |
|---|---|---|---|
| *A3.1 - Value of the use of change management tools.* | | *A3.2 - Value to requirements management and requisites.* | |
| There is a single project in the change management tool with a single workflow shared by all team members. | 0 | The Document of Software Requirements Specification (SRS) is updated simultaneously with the software | -3 |
| There is a unique project in the change management tool but not all team members have their workflow (branch). | 1 | SRS is updated only if new requirements are added to Software | -1 |
| There is a unique project in the change management tool used, and workflows (branches) are administered by each team member involved | 3 | SRS can not be upgraded, and must be strictly enforced. | 0 |
| *A3.3 - The value to functional product.* | | *A3.4 - Value to documentation.* | |
| Generate deliverable upon project completion without making testing. | 0 | It requires detailed documentation at project start. | -3 |
| Generate deliverable with manual testing after each iteration. | 1 | It requires only documentation needed at the beginning of each iteration. | -1 |

**Table 3 (cont.).** Attributes and criteria associated with Metric: Production capacity of deliverables

| | | | |
|---|---|---|---|
| Generate deliverable with testing automated and integrated with other functions after each iteration. | 3 | No documentation is required to begin implementing the functionality included in one iteration. | 0 |

- Metrics 4 - Customer communication: The quality model proposed will propitiate the incorporation of the client, as an active member in all stages of the project. Thus, this metric will assess implementation of regular communication mechanisms between the client and the team.

**Table 4.** Attributes and criteria associated with Metric: Customer Communication

| *Positive Attributes* | | *Negative Attributes* | |
|---|---|---|---|
| *A4.1 - Assess collaboration with the customer.* | | A4.2 - *Assess contract negotiation.* | |
| Customer collaborates to team demand. | 0 | There is detailed recruitment at the beginning and no changes accepted. | -3 |
| Customer is part of the team, answers queries and plans iterations. | 1 | The contract requires consider changes during the project. | -1 |
| Customer is part of the team. He responds consultations, planning iterations, and collaborates on writing and testing requirements | 3 | The contract exists but does not affect the level project development process. | 0 |

It is worth mentioning that for the design of this proposal, and taking into account defined in [22], positive attributes were considered (those who try to emphasize), and negative attributes (those who try to belittle). Thus, the positive attribute is measured on a scale from 0 to 3, and the negative attribute on a scale from -3 to 0. Thus, each metric could obtain a measure of -3, in the case that both attributes take the worst value (-3 for negative and the positive attribute 0), and 3, in the case that both attributes take the best value (0 negative attribute and 3 positive attribute). If a zero or near zero value is obtained, it means that the measurement values are not significantly above the positive negative.

Therefore, and taking into account the details of the associated criteria for the final value of each metric, it must consider both the corresponding measure to positive as the associated to the negative and the sum of its values, shown in (1):

$$Mi = M(Ai.1) + M(Ai.2) \quad i=1..4 \tag{1}$$

For example, the metric 1 (M1) - Election of the life cycle is measured by adding the measure of the value that the process gives the cycle of iterative and incremental life (A1.1) over the life cycle cascade (A1. 2).

### 3.2 Validation: Experience design

It is also necessary, design the process of validation of information that QuAM provides with the tools necessary to do so. To do this, in principle, the PYMES companies of the NEA Software Industry are convened, to assist in the validation of QuAM with real production environments to detect successes or issues to be improved in the model definition.

One of the instruments to be used will be an online survey with closed questions referred to the SW development processes of these companies. The same, it was designed and implemented through Google Forms, to facilitate dissemination among participants of experience and maximum reliability in the process of gathering information. The target population is made up of a group of 15 NEA companies, characterized by work on development projects web applications.

Currently the validation process has started with 25% of companies invited to take part of it. And it is expected that once the process of data collection is completed, an analysis of the obtained will be conducted to generate partial reports to determine the level of quality associated with agile business processes. The process will not end there, but the information obtained must be filed with the involved to achieve the feedback to determine if the concluded from the use of the proposed model is approximately or not, the reality perceived by the companies.

## 4 Conclusions and Future Work

The main contribution of this work is the preliminary definition of the components that form part of a quality model that helps to assess the quality of agile processes in PYMES dedicated to software development. There are several papers in the literature with the aim of improve the quality of the development process, submit proposals to adapt norms and standards to the philosophy of agile methodology. However, they are not specifically focused on the evaluation of the results obtained by processes under the agile philosophy. Thus, in principle, the QuAM presentation as a new approach to quality model will allow to start the cycle of quality assessment in real software projects guided through streamlined processes.

As future work, it is intended to obtain results of the validation experience presented in this article. And based on that, start defining a framework, including guidelines and practical guides, whose objective is the automation of measuring the quality of software projects based on agile processes.

# 5 Acknowledgements

This work was done under the accredited research project "Assessment Framework for Software Quality" EIUTIRE0002205TC of UTN. It should also be noted that the article is part of the activities planned in the Project on Technology and Social (PDTS) Development presented, "Contribution to the competitiveness of Software development NEA companies", IP253, evaluated and approved by the National Council of Scientific and Technical Research of Argentina (CONICET).
It is necessary to acknowledge the assistance of researchers Engineer Nicolás Tortosa, Blas Cabas Geat and Maximiliano Ulibarrie.

# 6 References

1. Navarro, J. M., & Garzás, J. (2010). Experiencia en la implantación de CMMI-DEV v1. 2 en una micropyme con metodologías ágiles y software libre. REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software, 6(1), 6-15.
2. Reporte anual sobre el Sector de Software y Servicios Informáticos de la República Argentina. OPSSI, ABril 2016. Disponible en http://www.cessi.org.ar/descarga-institucionales-2007/documento2-130347cd83ae771a9f3db3da5407269a
3. Mas A., Amengual E. (2005). "Las mejoras de los procesos de Software en las pequeñas y medianas empresas (pymes). Un nuevo modelo y su aplicación a un caso real". Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.1, No. 2
4. Pasini, A. C., Esponda, S., Bertone, R. A., & Pesado, P. (2008). "Aseguramiento de Calidad en PYMES que desarrollan software." XIV Congreso Argentino de Ciencias de la Computación.
5. Pflegger, S. (2002) "Ingeniería de Software. Teoría y Práctica." Pearson Education.
6. Letelier, P., Penadés, P. (2006) "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)" Técnica Administrativa, Buenos Aires. ISSN 1666-1680
7. Alliance, A. (2001). "Agile manifesto". Disponible en http://www. agilemanifesto. org
8. Rujana, M., Romero Franco, N., Tortosa, N., Tomaselli, G., & Pinto, N. (2016, May). Análisis sobre adopción de metodologías ágiles en los equipos de desarrollo en pymes del NEA. In *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*.
9. Córdoba, J., Cachero, C., Calero, C., Genero, M., & Marhuenda, Y. (2007, October). Modelo de Calidad para Portales Bancarios. In *XXXIII Conferencia Latinoamericana de Informática (CLEI'07)*.
10. Behkamal, B., Kahani, M., & Akbari, M. K. (2009). Customizing ISO 9126 quality model for evaluation of B2B applications. Information and software technology, 51(3), 599-609.
11. ISO/IEC 9126: "Software Engineering - Product quality", International Organization for Standardization, 2000.
12. Matalonga, S., & Rivedieu, G. (2015). AGIS: hacia una herramienta basada en ISO9001 para la medición de procesos ágiles. Computación y Sistemas, 19(1), 163-175. Disponible en http://www.agilemanifesto.org/iso/es/ Último acceso 06/2016
13. International Organization for Standardization. (2000). ISO 9001: 2008: Quality Management Systems-Requirements. International Organization for Standardization.
14. Cohen, D., Lindvall, M., & Costa, P. (2003). Agile software development. DACS SOAR Report, 11.

15. Paulk, M. C. (2001). Extreme programming from a CMM perspective. Software, IEEE, 18(6), 19-26.
16. Piattini, Oktaba, Orozco, “COMPETISOFT. Mejora de procesos software para pequeñas y medianas empresas”, Editorial Ra-Ma, Año 2008.
17. D. Kane and S. Ornburn, "Agile Development: Weed or Wildflower?" CrossTalk, The Journal of Defense Software Engineering, http://www.stsc.hill.af.mil/crosstalk/2002/10/kane.html, 2002. (1.3.2006)
18. J. Nawrocki, W. Bartosz, and A. Wojciechowski, "Toward Maturity Model for eXtreme Programming," In proceedings of the 27th Euromicro Conference, pp. 233-239, 2001.
19. M. C. Paulk, "Extreme Programming from a CMM Perspective," Software, vol. 18, issue 6, pp. 19-26, 2001
20. R. Turner and A. Jain, "Agile Meets CMMI: Culture Clash or Common Cause," In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe, pp. 153-165, 2002.
21. Turner, R., & Jain, A. (2002, August). Agile meets CMMI: Culture clash or common cause?. In *Conference on Extreme Programming and Agile Methods* (pp. 153-165). Springer Berlin Heidelberg.
22. Mendes Calo, K., Estevez, E. C., & Fillottrani, P. R. (2009). Un framework para evaluación de metodologías ágiles. In *XV Congreso Argentino de Ciencias de la Computación*.