

Validación de la reingeniería aplicada sobre la primera versión de Agile Quality Framework

Noelia Pinto¹; Nicolás Tortosa¹; Blas Cabas Geat¹; Lucas Ibáñez¹; César J. Acuña¹

{ns.pinto; nicotortosa; blascl47; lucas.sebib;
csr.acn}@gmail.com

¹CInApTIC (Centro de Investigación Aplicada en TICS),
UTN FRRe, French 414, Resistencia, Chaco, Argentina

Abstract. Currently, the software industry requires high quality products and services, which is achieved through the application of internationally recognized quality models and methods. However, these models in SMEs are very difficult to implement because it require a large investment in money, time and resources. With the aim of facilitating the adoption of practices that ensure quality, we have implemented a framework that integrates a quality model together with the tool that implements it, AQF.

AQF allows the evaluation of quality in agile environments, considering as the object of the measurement to the development process independently of the selected agile focus.

After the results obtained in previous validations, this article presents AQF 2.0 with new features and improvements adapted to NEA Software Industry reality, including evaluation process details by AQF regarding the Requirements Management in agile projects.

Keywords: Software Quality, Agile Requirements Quality, Software Engineering

Resumen. Actualmente las empresas de Software, para mantener la competitividad en el escenario mundial, deben ofrecer productos y servicios de calidad a través de la implementación de modelos y normas reconocidas internacionalmente. Sin embargo, esto no es posible en las Pequeñas y Medianas Empresas (PYMES) debido a los altos costos de implementación, escasas de personal especializado y tiempos acotados. Además, en el Nordeste Argentino (NEA), las PYMES, han iniciado la aplicación de prácticas ágiles de desarrollo de software que se contraponen a formalizaciones exigidas por los procesos de certificación de calidad más usados actualmente.

Por tal motivo se ha desarrollado AQF, un framework que integra un modelo de calidad junto a una herramienta de software para la evaluación de calidad en entornos ágiles. Debido a que se obtuvieron resultados de validaciones previas, en este artículo se presenta AQF 2.0 con nuevas características y mejoras

adecuadas a la realidad de la Industria del Software en el NEA, incluyendo detalles del proceso de evaluación que utiliza AQF respecto a la Gestión de Requerimientos y Requisitos en proyectos ágiles.

Keywords: Evaluación de Calidad, Procesos Ágiles, PYMES de Software

1 Introducción

La mejora e innovación de los procesos de software, con el objetivo de incrementar la calidad de sus productos y servicios, se ha convertido, en los últimos años, en el elemento diferenciador que las empresas necesitan para mejorar sus niveles de competitividad en la Industria del Software.

Sin embargo, diversos estudios coinciden en la dificultad de las PYMES [1][2][3] para implementar programas de Mejoras de Proceso de Software (Software Process Improvement - SPI), fundamentalmente porque la aplicación de estos modelos resulta costosa en términos económicos y de esfuerzo, pues requieren una gran inversión en dinero, tiempo y recursos, sus recomendaciones son complejas de aplicar y el retorno de la inversión se produce a muy largo plazo [4]. De esta forma, los parámetros de tiempos de desarrollo y costo de soluciones afectarán directamente al trabajo que se realice, siendo la calidad la primera variable de ajuste disponible.

Debido a esto, en la búsqueda por lograr mejores procesos de desarrollo para incrementar la calidad de sus productos e incrementar su nivel de competitividad en escenarios actuales, la Industria del Software en el NEA ha iniciado la adopción de prácticas ágiles en sus proyectos enfocadas en ciclos de desarrollos cortos, iterativos e incrementales, con equipos pequeños y auto-organizados, diseños simples con refactorización de código, donde el desarrollo conducido por la prueba es práctica habitual y la participación de los clientes en forma frecuente permite presentar la evolución del producto en cada ciclo de desarrollo [5]. De igual manera, y más allá de la existencia de numerosos estudios que demuestran una creciente adopción de estos enfoques en desarrollo de software [6], el rápido crecimiento de la agilidad ha generado confusión, malas interpretaciones e incluso efectos negativos en el desarrollo de algunos proyectos de software [7].

Por ello, surge la necesidad de elaborar estrategias que permitan no solo guiar a las empresas en la adopción de las prácticas ágiles sino también en cuestiones relacionadas a la gestión de la calidad que contribuyan en la generación de valor en la producción de software. De hecho, resultados de estudios previos, realizados en la Industria del Software del NEA, demuestran la ausencia de instrumentos que permitan a las empresas integrar agilidad a sus ciclos de desarrollo sin dejar de lado aspectos relacionados a la calidad de software [8].

En este sentido, y como primera alternativa, en el CInApTIC se viene trabajando en el diseño y validación de un marco de trabajo que permita evaluar la calidad cuando se opta por trabajar con procesos ágiles de desarrollo de software. Dicho framework se denomina AQF (Agile Quality Framework) y su versión actual está compuesta por un modelo, QuAM (Quality Agile Model) y por una herramienta de software que brinda soporte a dicho modelo, QuAGI (Quality AGile) [9].

Por un lado, el modelo permite determinar, en base a métricas, atributos y criterios, los niveles de calidad asociados al proceso ágil que se evalúa. Y, por otro lado, los componentes de QuAM son gestionados a través de QuAGI [10] que automatiza el seguimiento de proyectos y proporciona informes respecto a la calidad de cada uno.

La plataforma ha sido validada en diversas experiencias de vinculación con PYMES y áreas de Sistemas pertenecientes a la Industria del Software en el NEA[9,11], y en base a los resultados obtenidos se planteó la necesidad de readecuar los componentes actuales de AQF.

Así, en este artículo se presenta por un lado la reingeniería del proceso de evaluación de calidad implementado por el framework, incluyendo un rediseño del modelo QuAM y una refactorización que permita adecuar QuAGI a las nuevas necesidades de AQF, incluyendo detalles del proceso de evaluación de una de sus métricas incluidas en el modelo que utiliza AQF referidas a la Gestión de Requerimientos y Requisitos en proyectos ágiles. Y, por otro lado, se muestran los resultados de una nueva experiencia de validación que permitirá evaluar esta propuesta mejorada del framework.

El resto del artículo se estructura como sigue, la sección 2 presenta los componentes de la primera versión de AQF y describe los resultados obtenidos a partir de experiencias de vinculación con PYMES de software de Chaco y Corrientes. La sección 3 presenta las características del proceso de reingeniería realizado sobre el framework AQF completo, incluyendo detalles de proceso de evaluación de la métrica asociada a la gestión de requerimientos y requisitos. Luego en la sección 4 se describe el caso de estudio y los resultados de validación obtenidos hasta el momento. Finalmente en la sección 4 se presentan las conclusiones y los posibles trabajos futuros.

2 Descripción de la primera versión del framework AQF

El framework de evaluación propuesto AQF (Agile Quality Framework) ofrece a las empresas valores cuantitativos respecto a la calidad de los procesos ágiles que implementan en sus ciclos de desarrollo de software y apreciaciones cualitativas que caracterizan tales niveles obtenidos.

Resulta importante destacar que, a fines de este trabajo, se denominan *Proyectos Ágiles* a aquéllos que implementan prácticas ágiles en base a cualquier enfoque tal como Scrum, Kanban, Lean Software Development o híbridos. Así, el aporte significativo de AQF se centra en que el nivel de calidad de un proyecto ágil no puede estar definido únicamente por un valor numérico, sino que deben analizarse una serie de cuestiones, en base al contexto en el que se desarrolla el proceso, que seguramente condicionarán la ejecución de éste.

En función a lo expuesto anteriormente, uno de los componentes de AQF es QuAM, un modelo conceptual que tiene por objetivo proporcionar un método que permita evaluar la calidad tanto de los procesos de desarrollo de software basados en prácticas ágiles como de los productos que se obtengan a partir de dicho proceso. Tal como se indica en la tabla 1, la primera versión de QuAM, definía un esquema de componentes [12] en un árbol de métricas (M_i , $i = 1...4$) compuestas por atributos medibles (A_i) a través de una serie de criterios con medidas asociadas, para configurar un modelo de

evaluación de calidad, basado en los principios del manifiesto ágil [13], y que ofrezca una medición objetiva de la calidad del proceso ágil implementado en determinado proyecto.

Tabla 1. Arbol de métricas definido en QUAM para la primera versión de AQF

	<i>Atributos Positivos</i>	<i>Atributos Negativos</i>
<i>Métrica 1: Elección del Ciclo de Vida</i>	A1.1 Valorar al Ciclo Iterativo e Incremental	A1.2 Valorar al Ciclo en Cascada
<i>Métrica 2: Evaluación del Equipo de Trabajo</i>	A2.1 Valorar las reuniones de Equipo	A2.2 Valorar cumplimiento del cronograma
	A2.3 Valorar la definición de roles	A2.4 Valorar al proceso sobre el equipo
<i>Métrica 3: Capacidad de Producción de entregables</i>	A3.1 Valorar el uso de herramientas de gestión de cambios	A3.2 Valorar a la gestión de requerimientos y requisitos
	A3.3 Valorar al producto funcional	A3.4 Valorar a la documentación
<i>Métrica 4: Comunicación con el Cliente</i>	A4.1 - Valorar la colaboración con el cliente	A4.2 - Valorar la negociación contractual

Sin embargo, no es suficiente contar con un modelo que permita medir el nivel de calidad de un proyecto ágil, si no se dispone de una herramienta que posibilite automatizar la gestión de los elementos del modelo y analizar los resultados obtenidos a partir de diversos casos evaluados. Por eso, para dar soporte a QuAM se ha desarrollado QuAGI, una aplicación web que permite el seguimiento de proyectos basados en prácticas ágiles junto a la posibilidad de realizar evaluaciones continuas respecto al nivel de calidad que se esté logrando en el proceso.

En primer lugar, y como se muestra en la Figura 1, QuAGI permite la administración de los proyectos a través de la visualización del diseño preliminar del plan, informes respecto a estados actual del proyecto, provee información integral de las actividades, sirve como herramienta de comunicación interna entre los integrantes del equipo, entre otros. En segundo lugar, da soporte a los procesos de toma de decisiones asistiendo a los responsables mediante reportes que informen sobre evaluación de calidad del proyecto en cuestión en el momento que lo requieran.

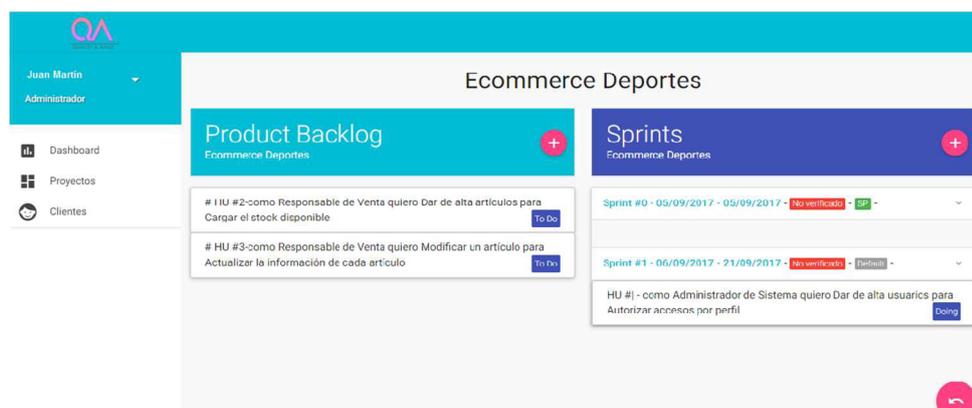


Fig. 1. Dashboard de cada Proyecto en QuAGI

Luego de la etapa de desarrollo, se hizo necesario que, la integración de QuAM y QuAGI en el framework, fuera validada por su implementación en ambientes reales de producción con proyectos ágiles. Los resultados de dichas experiencias fueron presentados en [14] y [15] respectivamente.

Así, como resultado positivo se pudo destacar que las empresas participantes lograron incorporar fácilmente el uso de QuAGI como herramienta para el seguimiento de los proyectos gracias a, según lo manifestaron, una interfaz amigable y con flujos de trabajos claramente definidos. Otro punto a destacar tuvo que ver con la importancia que otorgaron, por un lado, a la evaluación de la participación del cliente en el proyecto, cuestión considerada en la métrica 4 “Comunicación con el cliente” y por otro lado, a lo relacionado con roles, tareas y organización del proyecto, cuyas características se evalúan en la métrica 2 “Evaluación del Equipo de Trabajo”.

Por contrapartida, las empresas expresaron que los valores de calidad ofrecidos por AQF, si bien reflejaron en gran medida lo esperado de acuerdo con las características de los proyectos tomados como casos de estudio, no dejaban expuesta la ausencia o debilidades de algunas prácticas relevantes de acuerdo a cada proyecto. Estas cuestiones, en base a lo manifestado por las empresas, que afectan la calidad del proceso influyen, luego, en calidad a nivel de producto. Algunos de estos factores se relacionan con, por ejemplo, la completitud en el relevamiento de requerimientos, estimación del esfuerzo para cumplimentar requerimientos, independencia de requerimientos, entre otras. Asimismo, respecto a la Métrica 2 “Evaluación del Equipo de Trabajo”, resultaba muy tedioso para las empresas cargar en la plataforma información solicitada respecto a las reuniones de equipo; de hecho, el 100% de las empresas coincidieron en que no utilizarían esa característica de la plataforma.

Frente a todo lo expuesto anteriormente, y como parte de una estrategia de mejora continua para el desarrollo de AQF, se hizo necesario realizar ajustes al modelo propuesto a partir de su redefinición provocando la reingeniería del framework completo y la reelaboración de más casos de estudios que permitan validar, nuevamente, AQF en su totalidad.

3 Nueva versión de AQF.

3.1 AQF 2.0: Reingeniería a la primera versión de AQF

Teniendo en cuenta los resultados de las primeras experiencias de validación, se procedió a realizar un estudio comprensivo más amplio del framework, y se trabajó en la configuración de una nueva versión a la que se denominó AQF 2.0.

Esta nueva versión permite maximizar la comprensión y dar soporte a la integración adecuada de los procesos existentes en la empresa. Por otro lado, la reingeniería del framework se enfoca en mejorar individualmente la interpretación de los valores de calidad obtenidos por cada métrica, de forma tal de flexibilizar el proceso de gestión de cambio de los proyectos asociados.

Así, la estructura mejorada del modelo, expuesta en la Tabla 2, propone, en primer lugar, eliminar la métrica que evalúa el ciclo de vida elegido, pues el framework solo se aplicará sobre entornos ágiles cuyos ciclos sean guiados por prácticas de SCRUM, XP, Kanban, etc. La información del ciclo que guíe el proceso de desarrollo será, en su lugar, considerada información asociada al proyecto, pero no influenciará la medición de calidad ya que la evaluación contempla aspectos relacionados a las prácticas y no al enfoque que se elija. Así, el framework mide la calidad en la gestión de requerimientos independientemente de si el ciclo elegido es SCRUM o XP.

Otra modificación es la incorporación de la métrica “Gestión de Requerimientos y Requisitos” (Métrica 4) como un nuevo componente del modelo. Esto permite medir cómo se registran las necesidades asociadas al proyecto y si existe dependencia entre ellas (por ejemplo, si se usa SCRUM, se evaluará cómo se registran las historias de usuario y si su definición se realiza en función a secuencialidad para su ejecución en un sprint o iteración).

Además, el nuevo modelo implementa cambios respecto a los atributos y criterios medibles en cada una de las demás métricas. Por ejemplo, en la métrica “Capacidad de Producción de Entregables” se considera relevante la evaluación de calidad respecto a factores de estimación del proyecto y la posibilidad que el cliente realice testing sobre prototipos. En el caso de la métrica “Evaluación del Equipo de Trabajo” se evalúa lo referente a roles de los miembros y la especificación de tareas asociadas al proyecto. Para la métrica “Comunicación con el Cliente” se actualizaron solamente los criterios medibles incorporando mayor detalle a cada uno, teniendo en cuenta aspectos tales como participación del cliente en la gestión de requerimientos, validación de iteraciones, interacción con la plataforma, etc.

Desde su génesis la definición de QuAM, considera un conjunto de atributos por cada métrica: Positivos, aquellos factores que se desean enfatizar y promover para aumentar los niveles de calidad, y Negativos, aquellos factores que se desean disminuir para que no afecten insatisfactoriamente a la calidad del proceso. Así, el atributo positivo se mide en una escala del 0 al 4, y el atributo negativo en una escala del -4 al 0, como resultado de diversas validaciones previas. Entonces, cada métrica podría obtener una medida entre -4, en el caso que ambos atributos tomen el peor valor (-4 para el atributo negativo y 0 el atributo positivo), y 4, en el caso que ambos atributos tomen el mejor valor (0 el atributo negativo y 4 el atributo positivo). Si se obtiene un valor cero o cercano al cero,

significa que la medición no valora significativamente el atributo positivo por sobre el negativo.

Por tanto, y teniendo en cuenta el detalle de los criterios asociados, para obtener el valor final de cada métrica, se deben considerar tanto la medida correspondiente a los atributos positivos como la asociada a los negativos y la suma de sus valores, según lo especificado en (1):

$$M_i = \sum n M(A_i, n) \text{ donde } i = 1..4(1)$$

Por ejemplo, para el caso de la Métrica 3 (M3) – “Comunicación con el Cliente”, el atributo *A3.1 - Valorar la colaboración con el cliente*, obtendrá el valor máximo si el Cliente ha participado en la redacción del 100% de las historias de usuario y de los Sprint Planning. En el caso del atributo negativo, el valor de *A3.2 - Valorar la negociación contractual* será muy negativo si el Cliente no participa del equipo de trabajo y solo es considerado a fines contractuales.

Table 2. Configuración de métricas, atributos y criterios para QUAM 2.0

Métrica 1: Capacidad de Producción de Entregables	
<i>Atributos Positivos</i>	<i>Atributos Negativos</i>
<i>A1.1 Valor a la estimación</i>	<i>A1.2 Valorar el lead time</i>
-Todas las historias de usuario (US) de un sprint tienen asignado puntos historia	- El cliente no cuenta con una versión para realizar testing de aceptación
- Algunas US no tienen asignado puntos de historia	- El cliente puede realizar testing de aceptación al finalizar el proyecto
- Ninguna US tiene asignado puntos de historia	- El cliente puede realizar testing de aceptación al finalizar cada sprint
Métrica 2: Evaluación del Equipo de Trabajo	
<i>Atributos Positivos</i>	<i>Atributos Negativos</i>
<i>A2.1 - Valor a la definición de roles</i>	<i>A2.2 - Valor al proceso por sobre el equipo</i>
- Al asignar tareas se indica el tipo y responsable, distinguiendo rol de los miembros del equipo	- Se define las US sin agruparlas en sprints ni especificar tareas
- Al asignar tareas se indica solo el responsable pero no el tipo	- Se define las US, agrupándolas en Sprint pero no se especifican tareas
- No se indica responsable de la tarea	- Se define las US, agrupándolas en Sprint y especificando tareas
Métrica 3 - Comunicación con el cliente	
<i>Atributos Positivos</i>	<i>Atributos Negativos</i>
<i>A3.1 - Valorar la colaboración con el cliente</i>	<i>A3.2 - Valorar la negociación contractual</i>

· El Cliente es parte del equipo, participa de los sprints planning (SP) y redacta las US	- El Cliente solo es considerado desde el punto de vista contractual y no ingresa a la plataforma
· El Cliente es parte del equipo, redacta la mayoría de las US y participa en la mayoría de los SP	
- El Cliente es parte del equipo y participa en la mayoría de los SP	- El Cliente define el Contrato, y utiliza la herramienta
· El Cliente colabora a demanda del equipo	
- El Cliente no forma parte del equipo	- El Cliente define el Contrato y forma parte del equipo de desarrollo
Métrica 4 - Gestión de Requerimientos y Requisitos	
<i>Atributos Positivos</i>	<i>Atributos Negativos</i>
<i>A4.1 - Valorar la Completitud de US</i>	<i>A4.2 - Valorar la dependencia entre requerimientos</i>
- La redacción de historias de usuario se realiza de acuerdo con el template, incluye criterios de aceptación y están priorizadas	- El Cliente define dependencia en el 100% de los requerimientos.
- La redacción de historias de usuario se realiza de acuerdo con el template e incluye criterios de aceptación.	- El Cliente define dependencia en más del 75% de los requerimientos.
- La redacción de historias de usuario se realiza de acuerdo con el template	- El Cliente define dependencia en menos del 50% de los requerimientos.
- La redacción de US no respeta formato	- No hay dependencia de requerimientos

En este sentido, la reingeniería del framework AQF es producto de un análisis llevado a cabo respecto al impacto que se espera produzca esta nueva versión en las empresas usuarias de la plataforma. Así, las modificaciones propuestas se enfocan en optimizar la experiencia de usuario con AQF, colaborar en la mejora continua de los procesos ágiles que se lleven a cabo en los equipos y ofrecer una solución más aproximada a la realidad de las PYMES.

Así mismo, el framework tendrá impacto no solamente en la mejora de la calidad de las practicas agiles de los proyectos, sino también buscar un cambio estratégico para mejorar la imagen y el posicionamiento de las empresas en el mercado.

El impacto esperado de la reingeniería de AQF 2.0 en base a cada métrica es el siguiente:

- M1 - Capacidad de producción de entregables: basándose en los principios del Manifiesto Ágil y atendiendo la necesidad de las empresas de dar respuesta al problema de la “no entrega continua”, con esta nueva métrica,

se plantea mejorar la calidad del proceso a través de la estimación óptima de los requerimientos utilizando el sistema de puntos de historia. Así se intentan enfatizar prácticas que permitan realizar entregas continuas y funcionales al cliente.

- M2 - Evaluación del equipo de trabajo: tomando como principio que las mejores arquitecturas, requisitos y diseño del software surgen de equipos auto-organizados, se considera como una buena práctica que los roles del equipo en el proyecto mantengan una definición clara de los mismos, permitiendo una adecuada distribución de tareas.
- M3 - Comunicación con el cliente: otro problema detectado se relaciona con la difícil comunicación que se mantiene con el cliente. Así con esta métrica, y tomando como referencia uno de los principios del manifiesto ágil, AQF 2.0 busca que los responsables del negocio y el equipo de desarrollo trabajen de forma conjunta a lo largo de todo el proyecto estableciendo canales a través de QuAGI que posibiliten interacción directa entre el cliente y el proyecto, facilitando redacción de requisitos, definición de criterios de aceptación, validación de entregas, etc.
- M4 - Gestión de requerimientos y requisitos: Si bien la versión anterior contemplaba esta cuestión entre los criterios, luego de las primeras validaciones se estableció que el orden de importancia que debía darse a este componente justificaba la definición de una nueva métrica que oriente la calidad del proceso a través de una mejor definición de requerimientos y requisitos. Así, esta métrica busca acentuar aquellas prácticas ágiles relacionadas a una buena y completa definición de historias de usuario, a la inclusión de criterios de aceptación y a la disminución de dependencias entre requerimientos.

3.2 AQF desde la perspectiva de Gestión de Requerimientos y Requisitos en proyectos ágiles

Uno de los componentes de AQF v2 se enfoca en la evaluación de calidad respecto a la gestión de requerimientos y requisitos del proyecto ágil, como demanda de las empresas para enfrentarse al desafío que presenta el desarrollo de software moderno a través de la adopción de un enfoque ágil a los procesos que impliquen participación de cliente y, entonces, reaccionar rápidamente al dinamismo de requerimientos y entregas continuas del producto [16]. Sin embargo, a pesar de que los enfoques ágiles son cada vez más utilizados, la comunidad de desarrollo de software aún presenta mucho desconocimiento acerca de cómo las prácticas referidas a la ingeniería de requerimientos pueden resolver problemas comunes asociados a esto y colaborar en la mejora continua de la calidad.

Por tal motivo en la configuración de métricas de QuAM, la métrica 4 permite la evaluación de calidad respecto a la gestión de requerimientos y requisitos que son expresados a través de historias de usuario, la notación más utilizada en Ingeniería de Requerimientos ágiles [12]. Estas historias deben ser creadas para satisfacer las necesidades del equipo, por tal motivo, QuAM realiza su proceso de evaluación

basándose en las características propuestas por el modelo INVEST [17], un acrónimo que ayuda a recordar un conjunto de criterios, o lista de verificación, para evaluar la calidad de las historias de usuario. Establece que las mismas deben ser independientes entre sí; se debe poder negociar los detalles entre los usuarios y los desarrolladores; las historias deben tener valor para el cliente; deben ser lo suficientemente claras para que los desarrolladores puedan estimarlas; deben ser pequeñas; y deben poder probarse usando los casos de prueba predefinidos.

Como ya se describió, cada métrica del modelo QuAM se compone, además, de atributos positivos (aspectos que se intentan enfatizar), y atributos negativos (aspectos que se intentan atenuar).

Así en el caso de la métrica 4, la evaluación del atributo positivo considera la completitud de las historias de usuario. En primer lugar, verifica que la redacción se ajuste a las buenas prácticas definidas por Cohn [18], donde se especifica que esta acción se debe realizar en voz activa, evitando términos confusos y ambiguos, centrándose en lo que es importante. Para ello, se recomienda que se presenten con la forma “Como [rol] Quiero [qué] Para [objetivo]”, de esta forma la historia de usuario permite construir el entendimiento acerca del problema, describiendo la persona que detalla la necesidad junto a la salida esperada.

Por otro lado, este atributo añade más valor de calidad al proceso en el cual las historias de usuario incluyan criterios de aceptación, debido a que se busca mejorar la especificación de requerimientos y ofrecer una guía al equipo respecto a cuándo una necesidad del cliente se implementa por completo. Para lograr una mejor calidad en la definición, todo el equipo debe estar involucrado en el proceso, discutiendo de manera colaborativa los criterios de aceptación, siendo ésta la mejor manera de asegurar que todos comparten el mismo entendimiento acerca de lo que se va a construir, incrementando la madurez técnica del equipo de proyecto.

QuAM tiene por objetivo enfatizar el uso de esta práctica para lograr la conversión de diversos escenarios en pruebas de aceptación automatizadas, conduciendo al equipo a la utilización de técnicas como ATDD [19] (Desarrollo guiado por pruebas de aceptación) para agilizar el proceso de testing, dando lugar a que los diferentes roles en el proyecto trabajen en paralelo y se retroalimenten desde el inicio del proyecto.

Por último, el atributo positivo evalúa que las historias de usuario hayan sido priorizadas de acuerdo con el método MOSCOW [20], una manera de clasificar requerimientos revisando su prioridad: (1) “Must” – fundamentales y obligatorios para el éxito del producto (2) “Should” – importantes y deberían ser incluidos, pero no son obligatorios (3) “Could” – deseables, adicionales, pero no necesarios (4) “Won’t” – no serán incluidos por el momento.

Con respecto al proceso de evaluación del atributo negativo, se mide la dependencia entre requerimientos por cada sprint. De experiencias de validación anteriores logramos recabar que los proyectos ágiles se ven seriamente retrasados cuando se definen historias de usuario dependientes entre sí, de hecho, la calidad del sprint se ve disminuida cuando esto ocurre porque dificulta la planificación, la priorización y la estimación del proyecto en su conjunto.

Como se mencionó anteriormente, durante el proceso de medición, el atributo positivo puede tomar un valor en base a una escala que va de 0 a 4, y el atributo negativo en una escala que va del -4 al 0. Entonces, cada métrica podría obtener una medida entre -4, en el caso que ambos atributos tomen el peor valor (-4 para el atributo negativo y 0 el atributo positivo), y 4, en el caso que ambos atributos tomen el mejor valor (0 el atributo negativo y 4 el atributo positivo). Si se obtiene un valor cero o cercano al cero,

significa que la medición no valora significativamente el atributo positivo por sobre el negativo.

Por tanto, para obtener el valor final de métrica, se deben considerar tanto la medida correspondiente al atributo positivo como la asociada al negativo y la suma de sus valores, según lo especificado en (1):

Tabla 3. Configuración de métricas, atributos y criterios de QUAM

Atributo Positivo		
A4.1 Valorar la completitud de Historias de Usuario		
Criterio	Valor	Descripción
- La redacción de historias de usuario se realiza de acuerdo al template, incluye criterios de aceptación y están priorizadas	4	Se considera un nivel excelente cuando el 100% de las historias de usuario están redactadas según el estándar, incluyen criterios de aceptación y además se encuentran priorizadas.
- La redacción de historias de usuario se realiza de acuerdo al template e incluye criterios de aceptación	3	La redacción de US se realiza de acuerdo al template e incluye criterios de aceptación y priorización en al menos el 90% del total de US.
- La redacción de historias de usuario se realiza de acuerdo al template	2	La redacción de US se realiza de acuerdo al estandar, incluyen criterios de aceptación y se encuentran priorizadas en hasta un 60% del total de US.
- La redacción de historias de usuario no respeta formato	0	No se redactaron las US según el estándar, ni se encuentran priorizadas y tampoco incluyen criterios de Aceptación

Tabla 4. Configuración de métricas, atributos y criterios de QUAM

Atributo Negativo		
A4.1 Valorar la dependencia entre los requerimientos		
Criterio	Valor	Descripción
-No hay dependencia de requerimientos.	0	Dentro de los sprint definidos en el proyecto, no existe dependencia entre las historias de usuario.
- El Cliente define dependencia en menos del 50% de los requerimientos	-1	En al menos el 50% de los sprints definidos existen historias de usuario con dependencia.
- El Cliente define dependencia en más del 75% de los requerimientos	-2	En al menos el 75% de los Sprints existen historias de usuarios con dependencia.
- El Cliente define dependencia en el 100% de los requerimientos	-4	En la totalidad de los sprints definidos existen historias de usuario con dependencias.

4 Caso de Estudio y resultados obtenidos

Como se comentó en la sección anterior la herramienta que da soporte a QuAM es QuAGI[11], una aplicación web que permite el seguimiento de proyectos basados en prácticas ágiles junto a la posibilidad de realizar evaluaciones continuas respecto al nivel de calidad que se esté logrando en el proceso.

La arquitectura propuesta de QuAGI, tal como se muestra en la figura 2, consiste en una plataforma basada en componentes reutilizables, permitiendo así que las aplicaciones que se integren a la plataforma hagan uso de dichos componentes. En la capa de presentación, el diseño de interfaz y la interacción con el usuario de la aplicación fue realizado utilizando el framework Materialize junto a funciones jQuery que facilitan su implementación. Como framework de desarrollo se utiliza Django, basado en un patrón MVC (Modelo-Vista-Controlador) que separa la lógica de negocios de la lógica de presentación mediante la utilización del sistema de plantilla. La elección de Django se debe fundamentalmente a la filosofía de re-utilización, conectividad y extensibilidad de componentes sumado al desarrollo rápido de aplicaciones web. Relacionado a esto la decisión de utilizar PostgreSQL se debe a la robustez que proporciona integrado a este framework.



Fig. 2. Arquitectura de QuAGI

QuAGI hasta el momento permite el acceso a tres tipos de usuario: Rol de Administrador de Proyecto (responsable máximo del Proyecto), Miembros del Equipo (aquellos a quienes se les asignan tareas por cada proyecto) y Cliente (quien define los requisitos del proyecto). En la figura 3 se muestra la interfaz a la que pueden acceder desde cualquiera de los 3 roles posibles, teniendo la posibilidad de, en base a los permisos otorgados, gestionar proyectos, realizar el seguimiento de tareas, visualizar informes, etc.



Fig. 3. Arquitectura de QuAGI

Finalizada la nueva definición de componentes para QuAM, se procedió a incorporar las modificaciones sobre la herramienta web QuAGI de forma tal de integrar y obtener la versión AQF 2.0.

Sin embargo, a fines de este estudio, no es suficiente con implementar técnicamente las mejoras, sino que resulta necesario validar a través de experiencias reales de uso. Así, se dio inicio a una nueva experiencia de validación que, en esta oportunidad, contempló el análisis del impacto de AQF 2.0 en 20 equipos pertenecientes a empresas PYMES de Desarrollo de Software y áreas de Sistemas, cuya convocatoria se hizo gracias a contactos obtenidos de experiencias anteriores.

El método de validación significó acordar, con los equipos participantes, la selección de un proyecto en ejecución y guiado por prácticas ágiles, registrando información respecto a un sprint completo en QuAGI y completando una encuesta online, con un esquema de preguntas con valoración como se ve en la Tabla 3, que permitiera obtener resultados respecto a esta nueva interacción con el framework.

Table 5. Preguntas de la encuesta de validación

Escala de 1 (Nada Útil) - 3 (Muy Útil) <i>¿Le resulta útil medir la calidad del proyecto asociado con....</i>	Escala de 1(Nada de acuerdo)-3 (Muy de acuerdo) <i>¿Está de acuerdo en que resulta negativo....</i>
P1 ¿la estimación?	P2 ¿no validar cada release con el cliente?
P3 ¿la definición de roles y asignación de tareas?	P4 ¿no organizar las historias de usuario en sprints?
P5 ¿la comunicación activa con el cliente y su participación?	P6 ¿que el cliente solo participe del contrato?
P7 ¿la definición de requisitos y requerimientos?	P8 ¿definir dependencias excesivas entre requerimientos?

En líneas generales la experiencia duró aproximadamente 20 días, durante los cuales, al momento de finalización del trabajo, se establecía una reunión con los equipos (muchas veces no presencial) en la cual se registraron problemas o percepciones obtenidas luego del uso de la plataforma AQF.

El estudio de los resultados de validación se enfoca en la evaluación respecto a cada uno de los componentes considerados en el modelo. Así, como se observa en la Figura

2, para la métrica 1 “Capacidad en la producción de entregables” casi el 70% de las empresas considera “Muy útil” evaluar el proceso de estimación en cada sprint, considerando puntos de historias. Solo un 2% opinó que no sería relevante a la calidad del proceso ágil contemplar la estimación realizada en cada sprint. Por otro lado, se ha obtenido que el 60% de los participantes está de acuerdo en que resulta negativo no contemplar la validación que el cliente realiza sobre cada release.



Fig. 2. Validación de la métrica 1

En cuanto a la métrica 2 “Evaluación del Equipo de Trabajo”, y como se observa en la Figura 3, casi el 90% de las empresas ha considerado “Muy útil” valorar la definición de roles y asignación de tareas para el aseguramiento de calidad de procesos ágiles. De hecho, muchas de los participantes han manifestado que una de las principales debilidades en la calidad de los proyectos es la ausencia de definición clara de roles en el equipo. Respecto al atributo negativo, el 100% de las respuestas coinciden en estar de acuerdo con incluir la evaluación de configuración de sprints.

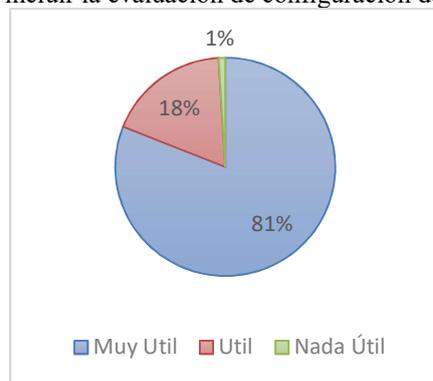


Fig. 3. Validación de la Métrica 2

En el caso de la métrica 3 “Comunicación con el Cliente”, el 100% de los participantes, tanto en la validación del atributo positivo como del negativo, han establecido que resulta muy útil considerar la participación del cliente en la evaluación

de calidad de los proyectos ágiles, no solo al inicio sino durante todo el proceso. Sin embargo no se ha podido validar su uso en QuAGI, pues esta métrica solo obtiene un valor asociado a localidad al finalizar el proyecto.

Respecto a la métrica 4 “Gestión de requerimientos y requisitos”; como se ve en la Figura 5, el 85% de las empresas opina que resulta “Muy útil” evaluar la completitud de las historias de usuario al realizar el seguimiento de los proyectos guiados por ciclos de desarrollo ágil. Sin embargo, un 15% considera “Poco útil” evaluar este aspecto como componente en la calidad del proceso, pues manifiestan que debido a límites de tiempo y capacitación, los requisitos se registran sin seguir un patrón y sin incluir criterios de aceptación. En cuanto al atributo negativo un 60% está de acuerdo que la evaluación de las dependencias entre requerimientos impacta negativamente en la calidad asociada al proyecto, el resto no está nada de acuerdo con considerar este aspecto.



Fig. 4. Validación de Métrica 4

5 Conclusiones y Trabajos Futuros

En este trabajo se ha presentado la nueva configuración para el framework AQF luego de un proceso de reingeniería, junto a los resultados de una primera etapa de validación de sus componentes.

De esta forma, por una parte, se ha comprobado que AQF aporta a la Industria del Software una herramienta que permite a las empresas medir aspectos relacionados a la calidad en base a las prácticas ágiles que guían el desarrollo de sus proyectos. Asimismo, con AQF se promueven la aplicación de estrategias en pos de la mejora continua enfocada en lograr madurez técnica de los equipos.

Gracias a la comparación de las experiencias realizadas a través de la implementación de las dos versiones de AQF, en idénticos ambientes de producción, se evidencia una mejor aceptación de los valores obtenidos por parte de los participantes, y se ha logrado reducir el gap entre el modelo conceptual propuesto y el contexto real de aplicación. Además se ha comprobado que, si bien AQF no reemplaza una certificación internacional, representa una herramienta que logra de manera más simple y eficiente que las empresas alcancen niveles de calidad necesarios para enfrentar el mercado actual. Mejorar la calidad en las prácticas ágiles favorecerá la

relación con los clientes, los tiempos de entrega y la competitividad de las PYMES en el contexto actual de la Industria.

Como trabajos futuros se pretende ampliar el caso de estudio incorporando más empresas y áreas de Desarrollo de Software, validando el uso de AQF 2.0 desde el inicio hasta el fin de proyectos ágiles y relevando la experiencia de su implementación sobre ambientes reales de producción, realizando los ajustes que fueran necesarios. Además, y a fin de facilitar aún más la vinculación con el framework, se pretende incorporar a la plataforma un agente inteligente que permita alertar sobre ciertos eventos de forma tal de acelerar la toma de decisiones por parte de los responsables en pos de mejorar la calidad del proyecto ágil en cuestión.

6 Agradecimientos

El presente artículo está enmarcado en el proyecto “Evaluación de Calidad en Procesos Ágiles de Desarrollo de Software”, que es financiado por la UTN y ejecutado en el Centro de Investigación aplicada en TICS (CInApTIC) de la Facultad Regional Resistencia, con el código IAI4445TC.

7 Referencias

1. 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
2. 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.
3. Pflegger, S. (2002) “Ingeniería de Software. Teoría y Práctica.” Pearson Education.
4. Javier Garzás, Carlos Manuel Fernández & Mario Piattini (2009). “Una aplicación de la Norma ISO/IEC 15504 para la evaluación por nivel de madurez de Pymes y pequeños equipos de desarrollo”. Revista Española de Innovación, Calidad e Ingeniería del Software, Vol.5, No. 2.
5. Swebok, [Online] Available: <http://www.computer.org/portal/web/swebok>
6. VersionOne (2015). 9th Annual State of agile Survey. VersionOne, 9, 16.
7. Bollati, V. A., Gaona, G., Pletsch, L. C., Gonnet, S., & Leone, H. (2017). “The state of agile development adoption in Argentine software companies”. In *Computer Conference (CLEI), 2017 XLIII Latin American* (pp. 1-10). IEEE.
8. Rujana, M., Romero Franco, N., Tortosa, N., Tomaselli, G., & Pinto, N. (2016). “Análisis sobre adopción de metodologías ágiles en los equipos de desarrollo en pymes del NEA”. XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016) ISBN 978-950-698-377-2
9. Pinto, Noelia; Acuña, César; Tortosa, Nicolás & Cabas Geat, Blas. (2017). “Evaluating Quality in Agile Developments. A first validation experience with NEA Software SMEs”. XIV Workshop de Ingeniería de Software (WIS). XXIII Congreso Argentino de Ciencias de la Computación. ISBN 978-950-34-1539-9. p. 722-731

10. Noelia Pinto, Gabriela Tomaselli, César Acuña, Liliana Cuenca Pletsch, Nicolás Tortosa, Blas Cabas Geat, Maximiliano Ulibarrie (2016) “Evaluando la Calidad en Desarrollos Ágiles. Caso de estudio con Pymes de Software de Chaco y Corrientes” 5° Congreso Nacional de Ingeniería Informática / Sistemas de Información. CONAIISI 2017 Universidad Tecnológica Nacional (UCASAL). Noviembre 2017. Santa Fe, Argentina.
11. Noelia Pinto, Gabriela Tomaselli, César J. Acuña, Liliana Cuenca Pletsch “QUAGI: Una propuesta para el seguimiento y evaluación de proyectos de Software Ágiles” V SABTIC, VIII STIN e XVIII Fórum, Mayo 2017. Três de Maio, Brasil
12. Noelia Soledad Pinto, Liliana Raquel Cuenca Pletsch, César Javier Acuña (2016) “Quality Evaluation in Agile Process: A First Approach” XXII Congreso Argentino de Ciencias Informáticas y Computación (CACIC 2016) Universidad Nacional de San Luis. Octubre 2016. San Luis, Argentina
13. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). Manifesto for agile software development.
14. Noelia Pinto, Gabriela Tomaselli, César Acuña, Liliana Cuenca Pletsch, Nicolás Tortosa, Blas Cabas Geat, Maximiliano Ulibarrie (2016) “Hacia un modelo de Evaluación de Calidad de Procesos Ágiles” 4° Congreso Nacional de Ingeniería Informática / Sistemas de Información. CONAIISI 2016 Universidad Católica de Salta (UCASAL). Noviembre 2016. Salta, Argentina.
15. Noelia Pinto; Gabriela Tomaselli; Analía Montero; Liliana Cuenca Pletsch; César Acuña 1er. Congreso Latinoamericano de Ingeniería. ENTRE RÍOS, ARGENTINA, 13-15, SEPT., 2017 CLADI “Evaluación de Calidad en Proyectos Ágiles: Una experiencia en pequeñas y medianas empresas del NEA”.
16. Inayat, I., Salim, S. S., Marczak, S., Daneva, M., & Shamshirband, S. (2015). A systematic literature review on agile requirements engineering practices and challenges. *Computers in human behavior*, 51, 915-929.
17. Cao L, Ramesh B (2008) Agile requirements engineering practices: an empirical study. *Software* 25(1):60–67
18. Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional
19. Pugh, K. (2010). *Lean-Agile Acceptance Test-Driven-Development: Better Software Through Collaboration*. Pearson Education.
20. ORACLE. *Custom Development Method Fast Track (CDM Fast Track) (2000) Method Handbook*, Release 1.2.0. Parkway, California, USA: Oracle Corporation.