

Evaluación de Calidad de Gestión de Requerimientos en proyectos ágiles

Noelia Pinto, CInApTIC (Centro de Investigación Aplicada en TICS), UTN Facultad Regional Resistencia, ns.pinto@gmail.com

Blas Cabas Geat, CInApTIC (Centro de Investigación Aplicada en TICS), UTN Facultad Regional Resistencia, blasc147@gmail.com

Nicolás Tortosa, CInApTIC (Centro de Investigación Aplicada en TICS), UTN Facultad Regional Resistencia, nicotortosa@gmail.com

Lucas Ibañez, CInApTIC (Centro de Investigación Aplicada en TICS), UTN Facultad Regional Resistencia, lucas.sebib@gmail.com

Cesar Acuña, CInApTIC (Centro de Investigación Aplicada en TICS), UTN Facultad Regional Resistencia, csr.acn@gmail.com

Resumen— En la actualidad, la industria del software requiere de productos y servicios de alta calidad, lo cual se logra mediante la aplicación de modelos y metodologías de calidad reconocidos internacionalmente. Sin embargo, estos modelos en pequeñas y medianas empresas son muy difíciles de implementar ya que supone para estas una gran inversión en dinero, tiempo y recursos.

Con el objetivo de facilitar la adopción de prácticas que aseguren la calidad se ha desarrollado AQF, un framework que integra un modelo de calidad junto a una herramienta de software que permite la automatización de dicho modelo.

AQF permite la evaluación de calidad en entornos ágiles, considerando como objeto de la medición al proceso de desarrollo más que a la metodología seleccionada.

En este artículo se presenta el proceso de evaluación que utiliza AQF enfocándose en la Gestión de Requerimientos y Requisitos en proyectos ágiles junto a los resultados de validación, a través de casos de estudio en ambientes reales de producción.

Palabras clave— *Calidad de Software, Gestión de Requerimientos Ágiles, Ingeniería de Software*

1. Introducción

Existen numerosas propuestas metodológicas que guían el ciclo del desarrollo de software y que inciden en distintas dimensiones del proceso. Las metodologías más tradicionales se centran especialmente en una rigurosa definición de: roles, actividades involucradas, artefactos que se deben producir, herramientas y notaciones que se usarán [1]. Sin embargo, estos enfoques no resultan ser los más adecuados para muchos de los proyectos relacionados a escenarios actuales, donde el entorno del sistema es muy cambiante y en donde se exige reducir drásticamente los tiempos de desarrollo, sin descuidar los niveles de calidad.

Como contraparte a esto, surgen ciclos de desarrollo ágiles que persiguen principios tales como la entrega incremental al cliente, la mejora continua y el énfasis en la colaboración cercana entre el equipo de desarrollo y los expertos del negocio [2]. De esta forma, estos enfoques se caracterizan por prácticas enfocadas en ciclos de desarrollos cortos, iterativos e incrementales, con equipos pequeños y auto organizados, 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 [3].

Diversos estudios demuestran que la adopción de prácticas ágiles se ha incrementado en los últimos años. Según el último reporte de VersionOne [4], más del 80% de las empresas usan algún enfoque guiado por prácticas ágiles para el desarrollo de sus aplicaciones, debido a que favorecen una adaptación más rápida a los cambios del mercado.

Sumado a la filosofía ágil, la aplicación de modelos de calidad promueve la mejora continua en las PYMES, estableciendo procesos con insumos y resultados medibles, reduciendo costos y promoviendo la eficiencia. Las empresas se ven beneficiadas al poder ofrecer a sus clientes productos de mayor calidad, asegurando el cumplimiento de los compromisos en los tiempos previstos [5].

Sin embargo, la mayoría de las investigaciones y estudios científicos, disponibles en la literatura, se han centrado, hasta el momento, solamente en el impacto de la calidad sobre los procesos de software sin evaluar cómo se realiza la implementación de las prácticas ágiles utilizadas.

De hecho, resultados de estudios previos realizados sobre diversas PYMES [6], demuestran la ausencia de estrategias que permitan a las empresas integrar agilidad a sus ciclos de desarrollo sin dejar de lado aspectos relacionados a la calidad de software.

En este sentido, y como primera alternativa, se viene trabajando en la implementación y validación de un marco de trabajo que permita evaluar la calidad cuando se opta por procesos ágiles de desarrollo de software. Dicho framework se denomina AQF v2 (Agile Quality Framework) y su versión actual, está formada por un modelo, QuAM (Quality Agile Model), compuesto por métricas, atributos y criterios que permiten medir los niveles de calidad asociados a las prácticas ágiles, y por QuAGI (Quality Agile), una herramienta de software que brinda soporte a dicho modelo a través de la automatización del seguimiento de proyectos y la visualización de diferentes informes [7].

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 [8]. 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.

Entonces, frente a todo lo antes expuesto, en este artículo se presenta el análisis de los resultados obtenidos luego de validar la implementación de AQF v2 enfocada en

evaluar la calidad respecto a la gestión de requerimientos y requisitos, en ambientes reales de producción sobre proyectos ágiles.

El resto del artículo se estructura como sigue: en la sección 2 se presentan trabajos relacionados a las prácticas ágiles y la Ingeniería de Requerimientos. Luego en la sección 3 se describe el framework AQF v2, sus componentes, y se hace énfasis en la métrica asociada a la gestión de requerimientos y requisitos. En la sección 4 se expone el diseño del caso de estudio, se explica el proceso de validación y se incluye el análisis de los resultados obtenidos. Y finalmente, en la sección 5 se presentan conclusiones y la propuesta de trabajos futuros posibles.

2. Trabajos Relacionados

La gestión de requerimientos trata sobre la comunicación entre los desarrolladores, clientes y usuarios para definir el nuevo sistema que brinde una solución al problema; a esta definición del sistema se le denomina especificación de requerimientos.

El análisis de requerimientos según Pressman [9] es una tarea de ingeniería del software que cubre el hueco entre la definición del software a nivel sistema y el diseño de software. El modelado de análisis debe lograr tres objetivos básicos: (1) describir lo que desea el cliente, (2) construir una base para el diseño de software, y (3) definir un conjunto de requerimientos que se pueda validar.

De acuerdo con lo expresado en los principios del Manifiesto Ágil [10], una de las prioridades radica en satisfacer al cliente mediante la entrega temprana y continua de software con valor, teniendo en cuenta que los requerimientos pueden cambiar en cualquier momento del desarrollo. Y esto, según expresan Serna y Suaza [11], es posible con una correcta documentación comprendida en una gestión adecuada de requerimientos y requisitos favoreciendo la entrega temprana de software requerido con nivel de calidad específico.

En el enfoque XP (Extreme Programming) [12] se establece que las Historias de Usuario (User Stories – US) sustituyen a los documentos de especificación funcional. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

Así, Cohn [13] presenta algunas buenas prácticas para la escritura de historias de usuario:

- *Especificaciones Ejecutables*: Preferentemente priorizar las especificaciones ejecutables a la documentación estática. Especificar los requerimientos en la forma de casos de pruebas ejecutables y el diseño como pruebas unitarias.
- *Requerimientos priorizados*: los equipos de desarrollo implementan las historias de usuario según su prioridad. Estas prioridades son especificadas por los stakeholders.
- *Visión de Requerimientos*: Al iniciar un proyecto nuevo se deberá invertir tiempo en identificar el alcance del proyecto y en crear una lista inicial de requerimientos priorizados.

- *Redacción en voz activa*: Evitar términos confusos y ambiguos al describir las historias de usuario, centrándose en lo que es importante. Para ello, se recomienda que las historias se presenten como oraciones con la siguiente forma: “Como [rol] Quiero [qué] Para [objetivo]”

Respecto a la formalización de la relación entre la calidad y agilidad en [14] se presenta AGIS, una herramienta que basada en los principios de mejora y auditoría de ISO 9001:2008, es capaz de medir el grado de agilidad de un proceso de acuerdo con los valores del manifiesto ágil. El modelo se compone de 10 dimensiones, entre las que se encuentra AGIS 9: “Gestión de Requerimientos y Trazabilidad”. En esta dimensión se evalúa el proceso de ingeniería de software que se está utilizando para la elicitación y como este está apoyado por prácticas asociadas.

Otra propuesta relacionada es Q-Scrum [15], un modelo de procesos basado en Scrum, que integra un conjunto de roles, documentos, y actividades, capaz de generar los documentos necesarios para satisfacer los requisitos del estándar ISO/IEC 29110 [16], incorporando la figura del analista de requerimientos como stakeholder fundamental en el proceso propuesto.

Pero, aunque estos estudios se basan conceptualmente en cuestiones asociadas a calidad y agilidad, poco énfasis realizan en particularidades relacionadas a medir la calidad respecto a las prácticas ágiles de elicitación de requerimientos.

Una propuesta relacionada a ello es QUS [17], un framework que establece 13 criterios de calidad que deben tenerse en cuenta al momento de redactar historias de usuario. Asimismo, como soporte se presenta AQUSA un software que detecta errores y sugiere mejoras a la redacción de estas historias de usuario.

Sin embargo, no se han encontrado estudios que integren todas las prácticas ágiles asociadas a un proceso y se realice sobre ello la evaluación de calidad resultante. Así, este artículo forma parte de un amplio trabajo de investigación que se viene realizando respecto al abordaje de la evaluación de calidad de procesos ágiles, que ya ha experimentado diversos casos de validación [18,19] y en el que se integra, en un mismo proceso, el seguimiento de la gestión de requerimientos ágiles junto a la evaluación de calidad asociada.

3. Resultados y Discusión

El framework de evaluación propuesto AQF v2 ofrece a las empresas valores cuantitativos respecto a la calidad de los procesos ágiles y apreciaciones cualitativas que caracterizan los niveles obtenidos. Su aporte significativo 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 condicionen la ejecución de este.

3.1 Configuración de QuAM

Uno de los componentes de AQF v2 es QuAM [20], 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 y por aplicaciones de software que dan soporte a la gestión de este modelo. Tal como se ve en la Tabla 1, se compone de un esquema organizado por métricas (M_i , $i = 1...4$) que incluyen atributos medibles (A_i) a través de una serie de criterios con medidas asociadas, configurando un modelo de

Evaluación de Calidad de Gestión de Requerimientos en Proyectos ágiles

evaluación de calidad que permite obtener el perfil ágil asociado a determinado proyecto.

El conjunto de atributos de cada métrica se clasifica en: Positivos, aquéllos que se desean enfatizar y promover para aumentar los niveles de calidad, y Negativos, aquéllos que se desean disminuir para que no afecten insatisfactoriamente a la calidad del proceso.

Tabla 1. Configuración de métricas, atributos y criterios de QuAM

Métrica 1: Capacidad de Producción de Entregables	
Atributos Positivos	Atributos Negativos
A1.1 Valor a la estimación	A1.2 Valorar el lead time
-Todas las historias de usuario de un sprint tienen asignado puntos historia	-De los sprints finalizados se encuentra validado menos del 10% de los criterios de aceptación.
- Más del 80% de las US tienen asignado puntos de historia	--De los sprints finalizados se encuentra validado menos del 40% y más del 10% de los criterios de aceptación.
- Entre el 40 % y el 80% de las historias de usuario tienen asignado puntos de historias.	-De los sprints finalizados se encuentra validado menos del 80% y más de 40% de los criterios de aceptación.
-Entre el 10% y el 40% de las historias de usuario tienen asignado puntos de historias.	-De los sprints finalizados se encuentra validado más del 80% de los criterios de aceptación-
-Menos del 10% de las US tienen asignado puntos de historia	-De los sprints finalizados se encuentra validado todos los criterios de aceptación
Métrica 2: Evaluación del Equipo de Trabajo	
Atributos Positivos	Atributos Negativos
A2.1 - Valor a la definición de roles	A2.2 - Valor al proceso por sobre el equipo
-El 100% de las tareas tienen asignado responsable y tipo	Existen US que no están agrupadas dentro de algún Sprint.
- Más del 80% de las tareas tienen definido responsable y tipo.	- Todas las US se encuentran agrupadas en Sprints y menos del 50% tienen tareas definidas.
- Entre el 80% y 50% de las tareas tienen definido responsable y tipo	Todas las US se encuentran agrupadas en Sprints y al menos el 60% tienen tareas definidas.
La definición de responsable y tipo se realizó en menos del 50% de las tareas	Todas las US se encuentran agrupadas en Sprints y al menos el 80% tienen tareas definidas.
Ninguna tarea tiene definido responsable ni tipo.	Todas las US se encuentran agrupadas en Sprints y el 100% tienen tareas definidas.

Evaluación de Calidad de Gestión de Requerimientos en Proyectos ágiles

Métrica 3 - Comunicación con el cliente	
Atributos Positivos	Atributos Negativos
A3.1 - Valorar la colaboración con el cliente	A3.2 - Valorar la negociación contractual
· El Cliente es parte del equipo, participa de los sprint planning (SP) y redacta las historias de usuario	- 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 historias de usuario y participa en la mayoría de los SP	-El Cliente solo ingresa a la plataforma al finalizar el proyecto.
- El Cliente es parte del equipo y participa en la mayoría de los SP	- El Cliente define el Contrato, e ingresa a la plataforma al final de cada sprint.
· El Cliente colabora a demanda del equipo	- El Cliente define el Contrato y forma parte del equipo de desarrollo
El Cliente no forma parte del equipo	
Métrica 4 - Gestión de Requerimientos y Requisitos	
Atributos Positivos	Atributos Negativos
A4.1 - Valorar la Completitud de Historias de Usuario	A4.2 - Valorar la dependencia entre requerimientos
- La redacción de historias de usuario se realiza de acuerdo al template	- Todos los Sprints incluyen al menos una dependencia entre las US que lo integran.
- La redacción de historias de usuario se realiza de acuerdo al template e incluye criterios de aceptación	- Más del 60% de los sprints incluyen US con dependencias
- La redacción de historias de usuario se realiza de acuerdo al template, incluye criterios de aceptación y están priorizadas	- Se definen dependencias en menos del 60% de los sprints
- La redacción de historias de usuario no respeta formato	- Ningún sprint incluye entre sus US dependencias.

Fuente: De elaboración propia.

Sin embargo, no es suficiente contar con un modelo de calidad, que permita medir el nivel de calidad de un proceso 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.

3.2 Gestión de Requerimientos y Requisitos

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 [21]. QuAM evalúa las historias de usuario basándose en las características propuestas por el modelo INVEST [22], que establece que cada historia de usuario debe ser: Independiente, Negociable, Valiosa, Estimable, Pequeña y Testeable.

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 nombradas en la sección anterior, con la estructura predefinida preferentemente en voz activa “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 [23] (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 que conforman el Product Backlog hayan sido priorizadas en función del valor que cada una proveerá a los clientes y stakeholders, teniendo en cuenta requerimientos esenciales y deseables. Si el equipo conoce las prioridades de realización, se eficientiza el proceso en la definición del alcance de cada iteración, promoviendo la calidad al proyecto ágil. Para implementarlo en QuAGI, se utiliza el método MOSCOW [24], 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.

Respecto al atributo negativo, éste mide la dependencia entre requerimientos por cada sprint o iteración. De acuerdo con experiencias de validación, ya citadas en este artículo, queda demostrado 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.

Tanto el atributo positivo como el negativo implementan un sistema de medición, en el primer caso el atributo puede tomar un valor en base a una escala que va de 0 a 4, y el 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):

$$M[4]=(A4.1)+(A4.2) \quad (1)$$

El proceso de evaluación de cada atributo surge a partir del análisis de las historias de usuario en cada sprint en base a lo especificado en la Tabla 2.

Tabla 2. Métrica 4: Gestión de Requerimientos y Requisitos

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 US se realiza de acuerdo al template, incluyen criterios de aceptación y se encuentran priorizadas en hasta un 60% del total de US.	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 US se realiza de acuerdo al template, incluyen criterios de aceptación y se encuentran priorizadas en hasta un 60% del total de US.	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 ninguna US hace uso del Template, no se encuentran priorizadas y tampoco incluyen criterios de Aceptación	0	No se redactaron las US según el estándar, ni se encuentran priorizadas y tampoco incluyen criterios de Aceptación
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 60% de los requerimientos	-1	En al menos el 60% de los sprints definidos existen historias de usuario con dependencia.
- El Cliente define dependencia en más del 60% de los requerimientos	-2	En al menos el 60% 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.

3.3 QuAGI: Aplicación Web para el seguimiento de proyectos ágiles

La herramienta que da soporte a QuAM es 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, QuAGI a través de sus interfaces interactivas permite la administración de los proyectos a través de la visualización del diseño preliminar del plan, informes respecto a estados del mismo, 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 la calidad del proyecto ágil.

QuAGI, puede ser accedida por todos los miembros del proyecto ágil de acuerdo a los permisos del perfil que le corresponda. Además el administrador del Proyecto podrá contar de manera just-in-time con información relativa a la gestión de requerimientos.

4. Caso de Estudio y Análisis de resultados

Debido a que AQF ha sufrido un proceso de reingeniería, se hace necesaria la evaluación de cada elemento en forma individual a fin de obtener la mejor versión del framework.

Se consideró que a fines de esta investigación sería necesario comparar el antes y el después de varios proyectos ágiles implementados en diferentes empresas PYMES del Nordeste Argentino.

4.1 Caso de Estudio

Para llevar a cabo la experiencia se trabajó junto a 20 equipos de proyectos de software guiados por algún enfoque ágil, localizados geográficamente en Chaco y Corrientes. Cada uno de estos varía en su tamaño, entre 3 y 10 integrantes, según la complejidad del proyecto. El objetivo del caso de estudio se centró en validar la información que AQF proporciona a los equipos respecto a la calidad de la gestión de requerimientos y requisitos, y cómo esto impacta en la mejora de los procesos asociados.

Previo a la adopción de AQF a cada proyecto, se realizó un diagnóstico del estado actual respecto a la calidad de la gestión de requerimientos tal como se venía realizando hasta el momento. Para ello, se realizó una encuesta online que analizaba:

- Formalización en la redacción de historias de usuario
- Priorización de requerimientos
- Inclusión de criterios de aceptación por parte del cliente

De este estudio preliminar se obtuvo por un lado que el 85% de las empresas no establecen criterios únicos para la redacción de las historias de usuario, sino que la forma queda liberada a deseo de cada equipo. Además se observó que el 50% no prioriza requerimientos, sino que establece aspectos a cumplir en momentos determinados.

Finalmente, ninguna de las empresas incluía criterios de aceptación en el 100% de las historias de usuario sino solo en aquellas que representan requisitos críticos para el proyecto. Al realizar la validación con el cliente, los requerimientos eran aceptados o no y se generaba una actualización de información del proyecto.

Los 20 equipos manifestaron que todas las cuestiones analizadas en la encuesta impactan negativamente en la calidad del proyecto, debido a que retrasan la entrega y muchas veces generan conflicto con el cliente pues no se interpreta correctamente el contexto y por ende sus necesidades.

Luego de este diagnóstico inicial, se procedió a implementar AQF, para lo cual se solicitó al equipo de la empresa que se registre en la plataforma QuAGI. Luego desde la administración de QuAGI se establecieron permisos, indicando quién tendría el rol de Administrador del proyecto, que a su vez debía dar de alta al Cliente o Product Owner.

Una vez configurado el equipo, se continuó con la carga de información relevante al proyecto (Fecha de Inicio, Enfoque ágil, Nombre del Proyecto, etc). Luego se realizó el registro de datos del sprint en ejecución, haciendo foco en el conjunto de historias de

Evaluación de Calidad de Gestión de Requerimientos en Proyectos ágiles

usuario involucradas en el Sprint Backlog. Este proceso, desde el alta inicial de proyecto a la carga de información completa sobre el sprint, duró aproximadamente 3 días en promedio por equipo.

Luego se indicó al Administrador de cada proyecto que solicite obtener la calidad correspondiente a la métrica 4, permitiendo que esa información quede registrada y pueda ser analizada a posteriori.

Respetando acuerdos de confidencialidad no es posible publicar información referente a la empresa, ni al proyecto ágil considerado en este estudio. Sin embargo, se exponen a continuación los resultados cuanti y cualitativos obtenidos de la experiencia.

4.2 Análisis de Resultados

En primera instancia, y como se muestra en la Figura 1, la mayoría de los proyectos participantes utilizan Scrum como framework de desarrollo ágil de software.

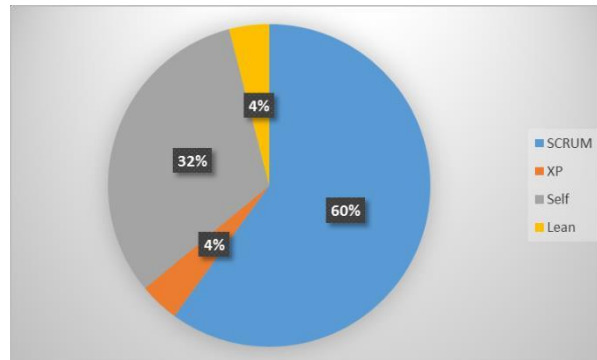


Figura 1. Enfoques ágiles – Fuente: De elaboración propia

Considerando la totalidad de proyectos estudiados, tal como se observa en la Figura 2, el valor obtenido durante la primera ejecución de evaluación de calidad de la métrica 2, fue en promedio igual a 1, lo que significa que el nivel de calidad de la “Gestión de Requerimientos y Requisito” en la mayoría de los casos es Regular.

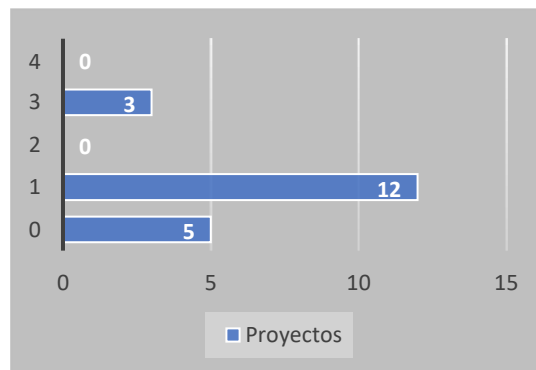


Figura 2. Valor de Calidad para la métrica por Proyecto – Fuente: De elaboración propia

Asimismo, en la Figura 3 se muestra la descomposición que se obtuvo, en general, para los valores del atributo positivo y del atributo negativo.

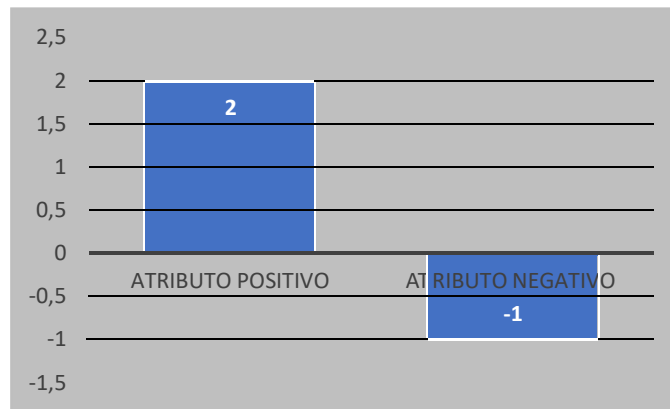


Figura 3. Resultados obtenidos luego de la primera evaluación – Fuente: De elaboración propia

En el caso del atributo positivo su valor indica, de acuerdo con lo establecido por QuAM, que entre el 60% y 90% de las historias de usuario cumplen con: redacción en base al template, incluyen criterios de aceptación y se han priorizado. Para mejorar este rendimiento, se han identificado las historias de usuario que no se correspondían con lo recomendado por el modelo y se ha entregado dicha información a la empresa.

Respecto al atributo negativo, se obtuvo un valor de -1 lo que según QuAM significa que menos del 60% de historias de usuario se definieron con alguna dependencia. Esto afectaría la implementación independiente de las historias de usuario, sin embargo y teniendo en cuenta las características del proyecto, el equipo ha manifestado que resultaría casi imposible no violar este criterio de calidad. Por lo que, si bien no forma parte del alcance de AQF v2 resolver dependencias entre requerimientos, se recomendó al equipo: (1) identificar las historias de usuario dependientes entre sí; (2) intentar segmentarlas en dos o más historias de usuario; (3) verificar si en lugar de historias de usuario podrían generarse tareas.

Luego de esta primera iteración, el proceso de validación continuó a través de las modificaciones realizadas por el equipo del proyecto estudiado, tomando como base las recomendaciones obtenidas a partir de la primera evaluación realizando AQF v2.

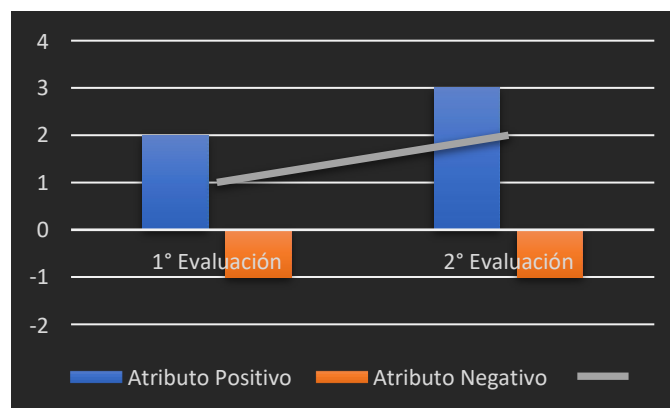


Figura 4. Resultados obtenidos de la segunda evaluación – Fuente: De elaboración propia

Como se ve en la Figura 4, los niveles de calidad asociados a la Gestión de Requerimientos y Requisitos mejoraron notoriamente al incorporarse las mejoras recomendadas respecto a la información registrada en cada sprint backlog.

Para fortalecer aún más los resultados cuantitativos obtenidos con AQF v2, se realizó una reunión con el equipo a fin de obtener un feedback cualitativo a través de la recolección de sensaciones y percepciones in situ.

Así, la mayoría de los Administradores manifestaron que la primera acción llevada adelante para mejorar el rendimiento fue mejorar la redacción de las historias de usuario identificadas como principal problema durante la primera evaluación. De hecho, en el proceso de readecuación se reconocieron errores en la interpretación de requerimientos, lo que permitió una redefinición de la historia de usuario con mayor precisión y exactitud. Además, si bien la mayoría de las historias de usuario habían sido priorizadas, no estaban definidas completamente pues algunas carecían de criterios de aceptación.

Por otro lado, se identificaron dependencias de historias de usuario en varios proyectos. En algunos casos se resolvió a través de la definición de tareas en cada historia de usuario favoreciendo la independencia. Pero en la mayoría no fue posible encontrar solución a la dependencia debido a que una historia de usuario debía completarse para dar inicio a la próxima.

5. Conclusiones y trabajos futuros

En este trabajo se ha presentado el proceso de evaluación que implementa AQF para medir la calidad respecto a la Gestión de Requerimientos y Requisitos de un proyecto ágil; junto a los resultados obtenidos luego de una experiencia de validación en un contexto real.

La propuesta del framework AQF v2 incluye un modelo de calidad, QuAM, que entre sus métricas considera la Gestión de Requerimientos y Requisitos a través de la evaluación del backlog de historias de usuario de cada sprint, independientemente de la metodología ágil que utilice el proyecto en cuestión.

Durante la experiencia de validación en ambientes reales, se ha podido verificar que el framework ofrece a los equipos de desarrollo una herramienta potente, fácil de usar y confiable al momento de medir la calidad asociada a la Gestión de Requerimientos y Requisitos. Esta conclusión se basa en la experiencia empírica cuyos resultados obtenidos demuestran, luego de la implementación de AQF v2, que no solo se redujeron considerablemente los tiempos de producción, sino que favoreció la comunicación con el cliente y eficientizó las entregas continuas disminuyendo el carry over (cantidad de historias de usuario no aceptadas por el cliente en una demo) de un sprint a otro.

A partir de este caso de estudio se registraron mejoras a nivel de interfaz para ser implementadas sobre QuAGI, y surgió la necesidad de automatizar las recomendaciones que puedan desencadenarse a partir del uso de AQF v2.

Como trabajo futuro se propone, por un lado, el framework AQF v2 debe validarse respecto a todos sus componentes, ampliando el set de casos de estudio y comparando informes de calidad obtenidos entre uno y otro a fin de obtener mejoras adicionales. Esta comparación debe realizarse a partir de recomendaciones automatizadas que puedan generarse directamente desde QuAGI. Por otro lado, se pretende incorporar al framework una tercera herramienta representada por un Asistente Virtual que responda a eventos y emita avisos/alertas a los equipos, de forma tal de realizar ajustes en etapas tempranas que permitan corregir el desvío de calidad que pueda sufrir el proyecto.

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] Letelier, P. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).
- [2] Alliance, A. (2001). Agile manifesto. Online at [http://www. agilemanifesto. org](http://www.agilemanifesto.org), 6(1).
- [3] Pardo, C., HURTADO, J. A., & COLLAZOS, C. A. (2010). Mejora de procesos de software ágil con Agile-SPI Process. *Dyna*, 77(164).
- [4] State of Agile Report. Disponible en <http://stateofagile.versionone.com/>
- [5] Instituto de Fomento Empresarial – IFE. “Polo IT - Hacia la Certificación de un Sistema de Gestión de Calidad”, <http://www.ife.gov.ar/articulo/articuloDetalle.aspx?articuloid=622>
- [6] 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).
- [7] Pinto, N., Acuña, C., Tortosa, N., & Cabas Geat, B. (2017). Evaluating Quality in Agile Developments. A first validation experience with NEA Software SMEs. In XXIII Congreso Argentino de Ciencias de la Computación (La Plata, 2017)..
- [8] 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.
- [9] Pressman, R. S. 2002, *Ingeniería de Software: Un enfoque práctico*.
- [10] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). *Manifiesto for agile software development*.
- [11] Serna, E., & Hernán Suaza, J. (2016). Documentar la elicitación de requisitos: Una revisión sistemática. *Ingeniare. Revista chilena de ingeniería*, 24(4), 703-714
- [12] Beck, K., & Gamma, E. (2000). *Extreme programming explained: embrace change*. addison-wesley professional.
- [13] Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional
- [14] 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.
- [15] Pasini, A. C., Esponda, S., Boracchia, M., & Pesado, P. M. (2013). Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110. In XVIII Congreso Argentino de Ciencias de la Computación.
- [16] ISO/IEC 29110:2011, “Software engineering -- Lifecycle profiles for Very Small Entities (VSEs)” 2011, ISO

- [17] Lucassen, G., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2016). Improving agile requirements: the quality user story framework and tool. *Requirements Engineering*, 21(3), 383-403.
- [18] Noelia Pinto, Gabriela Tomaselli, César J. Acuña, Liliana Cuenca Pletsch (2017) “QUAGI: Una propuesta para el seguimiento y evaluación de proyectos de Software Ágiles” V SABTIC, VIII STIN e XVIII Fórum, Três de Maio, Brasil
- [19] Noelia Soledad Pinto, Liliana Raquel Cuenca Pletsch, César Javier Acuña (2016) “Quality Evaluation in Agile Process: A First Ap-proach” XXII Congreso Argentino de Ciencias Informáticas y Computación Universidad Nacional de San Luis.
- [20] Pinto, N., et al (2016). Validación del diseño de componentes de QuAM: un Modelo de Calidad para procesos Ágiles. Publicado en Anales del IV Seminario Argentina-Brasil de Tecnologías de la Comunicación y la Comunicación (SABTIC 2016). ISBN 978-987-3619-15-1
- [21] Cao L, Ramesh B (2008) Agile requirements engineering practices: an empirical study. *Software* 25(1):60–67
- [22] Wake, B.: INVEST in Good Stories, and SMART Tasks (2003). <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- [23] Pugh, K. (2010). *Lean-Agile Acceptance Test-Driven-Development: Better Software Through Collaboration*. Pearson Education.
- [24] Clegg, D., & Barker, R. (1994). *Case method fast-track: a RAD approach*. Addison-Wesley Longman Publishing Co., Inc..