



**UNIVERSIDAD
NACIONAL
DE LA PLATA**

**Framework para la evaluación de calidad
de proyectos ágiles de software**

Tesista

Ing. Noelia Soledad PINTO

Director

Dr. César J. ACUÑA

Co-Director

Dr. Gustavo ROSSI

Tesis presentada para obtener el grado de

Doctor en Ciencias Informáticas

**FACULTAD DE INFORMÁTICA
UNIVERSIDAD NACIONAL DE LA PLATA**

- JULIO, 2020 -

Resumen

En la actualidad, una condición imperante para el crecimiento de la industria del software es que las empresas ofrezcan productos de mayor calidad, que satisfaga las demandas y exigencias del cliente, pero sobre todo que genere confianza al momento de su uso.

Por ello, las empresas de la Industria del Software tienen el desafío de desarrollar estrategias con el objetivo de incrementar la calidad y capacidad de sus procesos y, en consecuencia, la calidad de sus productos y servicios.

Este proceso de mejora involucra, por un lado, la adopción de un modelo de calidad adecuado a las características de la empresa, que favorezca la medición y seguimiento de diversos aspectos relacionados al proceso de desarrollo de software.

Y, por otro lado, relacionado a lo anterior, resulta necesario automatizar la gestión de elementos del ciclo de desarrollo de software, de forma tal de promover la reducción de costos y la eficiencia en los procesos de obtención de productos finales. En este caso, la filosofía ágil resulta ser el enfoque más adecuado para los entornos de desarrollo actuales, y se posicionan como alternativa a los procesos de desarrollo con alto costo en documentación y procesos excesivamente prolongados.

La presente tesis doctoral presenta, AQF (Agile Quality Framework) un framework cuya propuesta integra el diseño y desarrollo de un modelo que permita evaluar la calidad en procesos ágiles de software, y una plataforma que permitirá el seguimiento de proyectos de software ágiles junto a la evaluación sistemática de calidad del proceso de desarrollo.

Se pretenden unir esfuerzos con esta iniciativa y motivar a la comunidad académica a trabajar en calidad con las empresas desarrolladoras de software para mejorar la competitividad y la calidad global de esta industria a partir de un framework que se adecúe a sus realidades y les permita mejorar sus procesos fácilmente.

Abstract

Today, an important condition for the growth of the software industry is that companies can offer higher quality products, which meet the demands of the client, but above all that generate confidence at the time of use.

Therefore, companies in the Software Industry have the challenge of developing strategies with the objective of increasing the quality and capacity of their processes and, consequently, the quality of their products and services.

This improvement process involves, on the one hand, the adoption of a quality model appropriate to the characteristics of the company, which favors the measurement and monitoring of various aspects related to the software development process.

And, on the other hand, related to the above, it is necessary to automate the management of elements of the software development cycle, in order to promote cost reduction and efficiency in the processes of obtaining final products. In this case, agile philosophy turns out to be the most appropriate approach for current development environments, and they are positioned as an alternative to development processes with high cost in documentation and excessively long processes.

This doctoral thesis presents, AQF (Agile Quality Framework) a framework whose proposal integrates the design and development of a model that allows to evaluate the quality in agile software processes, and a platform that will allow the monitoring of agile software projects next to the systematic evaluation of the quality of the development process.

They intend to join efforts with this initiative and motivate the academic community to work in quality with the software development companies to improve the competitiveness and global quality of this industry from a framework that suits their realities and allows them to improve their processes easily.

Dedicatoria

*A mis abuelos Neito y Cabache,
que estarán conmigo, y para siempre, en cada paso.*

Agradecimientos

Esta tesis ha significado un proceso largo y no exento de desafíos, pero al final se ha completado, gracias a muchas de las personas que nombraré a continuación.

A mi director, y amigo, César por su guía, ayuda y dedicación; quien, a pesar de las dificultades que hemos atravesado, no ha dejado de apoyarme en este largo proceso. Sus recomendaciones y críticas, siempre constructivas, colaboraron en mejorar continuamente este trabajo. Muchas gracias amigo por haber confiado en mí, he aprendido mucho junto a vos, sobre todo en esas tardes de debate sobre temas de la tesis. Gracias por estar ahí.

A Gustavo Rossi, por su codirección, por haber dedicado su tiempo en las revisiones y por haber colaborado, desde su experta mirada, en la presentación final de esta tesis.

A mi equipo de investigación, por su compañerismo, sus charlas de mates en la oficina y su aporte invaluable para este trabajo. Una especial mención para Gaby, mi “profe”, con quien hemos pensado y analizado diversos detalles de este trabajo; y para Nico, Blas y Lucas, quienes han permitido la implementación final de las herramientas de esta tesis.

A Cuny, quien, en su rol de directora del grupo de investigación, confió en mi como becaria doctoral, lo cual contribuyó con parte del financiamiento para realizar los cursos del doctorado; apoyo sin el cual, esta tesis no hubiera sido posible.

A la Facultad de Informática, de la Universidad Nacional de La Plata, agradeciéndole a sus autoridades, docentes y personal administrativo; por su profesionalismo y el dictado de cursos de doctorado, que tanto han colaborado con esta tesis.

A mi querida UTN Facultad Regional Resistencia, lugar donde descubrí mi vocación por la docencia y la investigación, y desde donde sigo desarrollándome profesionalmente en esto que amo. Especialmente a Ale, y al equipo de la Secretaría de Ciencia y Tecnología, por su contaste apoyo y guía. Y a Jorge, por darme ese abrazo y esa palabra de aliento, en el momento justo. Muchas gracias.

A mis amigas y amigos, quienes con su tiempo y cariño han ayudado a levantar mi ánimo en esos momentos difíciles. En especial a Vero y Vale, que con su amor casi “maternal” me han contenido, escuchado y alentado a seguir adelante. Gracias, muchas gracias chicas.

A mi papá, de quién heredé el amor por mi profesión, y a mi mamá, de quién heredé mi vocación por la docencia; sin ustedes no sería quién soy, su sacrificio hizo posible que yo pudiera llegar hasta aquí. Y a toda mi familia, abuelas, Luis, Mari, suegros y cuñadas, por su amor, su compañía y su presencia de siempre. Sobre todo, a Mari, que me acompañó en varios de los viajes a La Plata para que no me sintiera sola. Los amo profundamente.

Y finalmente a Nico, mi esposo y compañero de vida, por su infinita paciencia y por ayudarme a comprender varios aspectos de esta tesis desde su mirada práctica y sencilla. Su incondicional amor fue mi gran motor en este trabajo. Te amo Chiki.

En general quisiera agradecer a todas y cada una de las personas que han vivido conmigo la realización de esta tesis doctoral, con sus altos y bajos, y desde lo profundo de mi corazón les agradezco el haberme brindado todo el apoyo, colaboración, ánimo y, sobre todo, cariño y amistad. ¡Muchas gracias!

ÍNDICE

<i>Capítulo 1 - Introducción</i>	14
1.1 - Motivación	16
1.2 – Hipótesis y Objetivos.....	21
1.3 - Contribuciones	22
1.4 - Enfoque Metodológico	24
1.4.1 Etapa de Resolución y Validación.....	25
1.5 - Estructura de la Tesis	27
<i>Capítulo 2 - Estado del Arte</i>	28
2.1 - Revisión Sistemática	30
2.1.1 Formulación de la pregunta	30
2.1.2 Selección de las fuentes	31
2.1.3 Selección de los estudios.....	32
2.1.4 Valoración de la calidad	33
2.1.5 Extracción de la información	34
2.1.6 Discusiones	39
2.2- Conclusiones	47
<i>Capítulo 3 - AQF: Framework para la Evaluación de calidad de procesos ágiles de software</i>	49
3.1 – Introducción.....	51
3.2 - QuAM: Modelo de Calidad para procesos ágiles de software	52
3.2.1 Relevamiento de la situación actual.....	52
3.2.2 Diseño de QuAM	60
3.2.3 Procedimiento de evaluación en QuAM.....	86
3.2.4 Discusión	88
3.3 - QuAGI: Aplicación Web para la automatización de QuAM.....	89
3.3.1 Características del proceso de Desarrollo	90
3.3.2 Diseño de la arquitectura.....	91
3.3.3 Discusión	94
3.4 – El valor agregado de AQF	95
3.5 - Conclusiones	97
<i>Capítulo 4 - Validación</i>	98
4.1 - Estudios de Casos.....	100
4.1.1 Estudio de Caso N° 1: Validación del framework AQF.....	101

4.1.2 Estudio de Caso N° 2: Evaluación de calidad en procesos ágiles desde la perspectiva del trabajo en equipo.....	113
4.1.3 Estudio de Caso N° 3: Experiencia de validación de QuAGI como herramienta de gestión de proyectos ágiles.....	120
4.2 - Conclusiones	137
<i>Capítulo 5 - Conclusiones y Trabajos Futuros</i>	139
5.1 - Conclusiones	141
5.2 -Trabajos Futuros	Error! Bookmark not defined.
5.3 - Publicaciones	144
ANEXO A	145
ANEXO B.....	155
ANEXO C	156
ANEXO D	159
ANEXO E.....	168
ANEXO F	171
ANEXO G	172
Referencias.....	173

ÍNDICE DE FIGURAS

Figura 1.1. Entidades involucradas en las vistas de calidad y sus relaciones [ISO/IEC 25010, 2011].....	18
Figura 1.2 Elementos de modelos de calidad según McCall y Boehm.....	19
Figura 1.3 Método de Investigación.....	24
Figura 2.1 Etapas del proceso de revisión sistemática utilizado.....	30
Figura 2.2 Formato utilizado para el registro de información relevante de los estudios primarios.....	38
Figura 2.3. Análisis cuantitativo de acuerdo al ámbito de aplicación.....	41
Figura 2.4. Análisis cuantitativo de casos de acuerdo a la contribución.....	42
Figura 2.5. Análisis cuantitativo de acuerdo al tipo de estudio.....	44
Figura 3.1 Resultado de entrevistas “Gestión de Proyectos”.....	56
Figura 3.2 Resultados de entrevistas “Porcentaje de Proyectos ágiles”.....	56
Figura 3.3 Resultados de entrevistas “Enfoques ágiles utilizados”.....	57
Figura 3.4. Resultados de entrevistas “Calidad de Software”.....	59
Figura 3.5 Diagrama de Clases QuAM.....	60
Figura 3.6 Esquema de Componentes de QuAM.....	63
Figura 3.7 Fases del Procedimiento de evaluación de QuAM.....	86
Figura 3.8 Aplicación web QuAGI.....	90
Figura 3.9 Épicas que componen el proyecto QuAGI.....	91
Figura 3.10. Arquitectura conceptual de QuAGI.....	92
Figura 3.11. Relación entre entidades del modelo en la capa de persistencia de QuAGI.....	93
Figura 4.1 Actividades planificadas para la experiencia de validación.....	103
Figura 4.2 Validación del atributo positivo en el Componente N° 1 de QuAM.....	105
Figura 4.3 Validación del atributo negativo en el Componente N° 1 de QuAM.....	105
Figura 4.4 Validación del atributo positivo en el Componente N° 2 de QuAM.....	106
Figura 4.5 Validación del atributo positivo en el Componente N° 3 de QuAM.....	107
Figura 4.6 Validación del atributo positivo en el Componente N° 4 de QuAM.....	107
Figura 4.7 Validación del atributo negativo en el Componente N° 4 de QuAM.....	108
Figura 4.8 Aspectos positivos en la evaluación de usabilidad.....	110
Figura 4.9 Aspectos negativos en la evaluación de usabilidad.....	111
Figura 4.10 Resultados de Informe de Calidad por componentes de los 20 equipos participantes.....	112
Figura 4.11 Utilidad de informes de Calidad de QuAGI en proyectos ágiles.....	113
Figura 4.12 Distribución de enfoques entre los equipos participantes.....	116
Figura 4.13 Niveles de calidad por equipo en el Componente N° 2. Primer ejecución.....	116
Figura 4.14 Distribución de valores en atributos negativo y positivo. Primera ejecución.....	117
Figura 4.15 Valores de atributos positivo/negativo y niveles de calidad luego de la 2da ejecución.....	118
Figura 4.16 Resultados de Encuesta inicial a participantes sobre su experiencia.....	123
Figura 4.17 Niveles de calidad obtenidos por los equipos del grupo QuAGI al inicio de la experiencia.....	124
Figura 4.18 Evaluación por parte de los administradores de proyecto respecto a lo informado por QuAGI.....	124
Figura 4.19 Resultados obtenidos por cada equipo respecto al <i>Informe de Avance del Sprint</i>	126
Figura 4.20 Resultados obtenidos por cada equipo respecto al informe de <i>Priorización</i>	126
Figura 4.21 Producción de historias de usuario por equipos en Taiga.....	128
Figura 4.22 Miembros del equipo con tareas asignadas sin finalizar.....	128
Figura 4.23 Evolución de los niveles de calidad en equipos usando QuAGI.....	129
Figura 4.24 Análisis de atributos del Componente N° 1 de QuAM al usar QuAGI.....	130
Figura 4.25 Análisis de atributos del Componente N° 2 de QuAM al usar QuAGI.....	131

Figura 4.26	Análisis de atributos del Componente N° 3 de QuAM al usar QuAGI.....	131
Figura 4.27	Análisis de atributos del Componente N° 4 de QuAM al usar QuAGI.....	132
Figura 4.28	Grado de comprensión de historias de usuario en los equipos	133
Figura 4.29	Avances en la producción de historias de usuario por equipo.....	133
Figura 4.30	Porcentajes de tareas sin responsable asignado.....	134
Figura 4.31	Grado de colaboración de la herramienta con la calidad del proceso	134
Figura 4.32	Resultados de la encuesta SUS	136
Figura D.1	Interfaz de QuAGI - Información básica del Proyecto.....	161
Figura D.2	Interfaz de QuAGI - Información del equipo	161
Figura D.3	Interfaz de QuAGI - Gestión de Historias de Usuario > Carga de Tareas	162
Figura D.4	Interfaz de QuAGI - Gestión de Historias de usuario > Criterios de Aceptación	163
Figura D.5	Interfaz de QuAGI - Mockups de Historias de Usuario	163
Figura D.6	Interfaz de QuAGI - Product Backlog y Sprint Backlog.....	164
Figura D.7	Interfaz de QuAGI - Dependencias entre historias de usuario de un Sprint.....	164
Figura D.8	Interfaz de QuAGI - Kanban board para Historias de Usuario	165
Figura D.9	Interfaz de QuAGI - Kanban board de tareas por Historia de Usuario por Sprint	165
Figura D.10	Relación entre los componentes de QuAM y los elementos que observa QuAGI	166
Figura D.11	Interfaz de QuAGI - Informe de Niveles de Calidad	166
Figura D.12	Niveles de Calidad según el valor de evaluación obtenido	167
Figura D.13	Interfaz de QuAGI - Descomposición de la Evaluación de Calidad por componente.....	168
Figura D.14	Informe de avance de historias de usuario	169
Figura E.1	Porción de código para la definición de componentes en QuAGI.....	170
Figura E.2	Porción de código en QuAGI para el cálculo de dos métricas	171
Figura E.3	Tecnologías utilizadas en cada capa de la Arquitectura conceptual	171

ÍNDICE DE TABLAS

Tabla 2.1. Sub preguntas de investigación	29
Tabla 2.2. Cadena de búsquedas utilizadas en cada fuente de datos.....	30
Tabla 2.3. Cantidad de Resultados de la ejecución de búsqueda	33
Tabla 2.4. Cantidad de resultados luego de aplicar criterios de inclusión y exclusión, y eliminar duplicados	34
Tabla 2.5. Estudios primarios seleccionados para revisión	35
Tabla 2.6. Valoración de la calidad de estudios primarios relevantes no duplicados	37
Tabla 2.7. Distribución de casos de estudio de acuerdo al ámbito de aplicación de la propuesta	38
Tabla 2.8. Distribución de casos de estudio de acuerdo a la contribución que se presenta	39
Tabla 2.9. Distribución de casos de acuerdo al tipo de estudio	41
Tabla 3.1 Secciones de la entrevista para el diagnóstico actual.....	51
Tabla 3.2. Población para entrevistas	52
Tabla 3.3 Resultados de entrevistas - “Prácticas ágiles”	56
Tabla 3.4 Resultados de entrevistas “Factores que impactan sobre la calidad de software”	57
Tabla 3.5 Descripción de Métrica N° 1.1	62
Tabla 3.6 Descripción de Métrica N° 1.2	63
Tabla 3.7 Descripción de Métrica N° 1.3	63
Tabla 3.8 Descripción de Métrica N° 1.4	64
Tabla 3.9 Descripción de Métrica N° 1.5	65
Tabla 3.10 Descripción de Métrica N° 1.6	65
Tabla 3.11 Relación entre atributos y criterios del Componente N° 1	66
Tabla 3.12 Descripción de Métrica N° 2.1	67
Tabla 3.13 Descripción de Métrica N° 2.2	68
Tabla 3.14 Descripción de Métrica N° 2.3	68
Tabla 3.15 Descripción de Métrica N° 2.4	69
Tabla 3.16 Descripción de Métrica N° 2.5	69
Tabla 3.17 Descripción de Métrica N° 2.6	70
Tabla 3.18 Descripción de Métrica N° 2.7	71
Tabla 3.19 Descripción de Métrica N° 2.8	71
Tabla 3.20 Descripción de Métrica N° 2.9	72
Tabla 3.21 Relación entre atributos y criterios del Componente N° 2	72
Tabla 3.22 Descripción de Métrica N° 3.1	73
Tabla 3.23 Descripción de Métrica N° 3.2	74
Tabla 3.24 Descripción de Métrica N° 3.3	74
Tabla 3.25 Descripción de Métrica N° 3.4	75
Tabla 3.26 Descripción de Métrica N° 3.5	75
Tabla 3.27 Descripción de Métrica N° 3.6	76
Tabla 3.28 Relación entre atributos y criterios del Componente N° 3	77
Tabla 3.29 Descripción de Métrica N° 4.1	78
Tabla 3.30 Descripción de Métrica N° 4.2	79
Tabla 3.31 Descripción de Métrica N° 4.3	79
Tabla 3.32 Descripción de Métrica N° 4.4	80
Tabla 3.33 Descripción de Métrica N° 4.5	80
Tabla 3.34 Descripción de Métrica N° 4.6	81
Tabla 3.35 Descripción de Métrica N° 4.7	82
Tabla 3.36 Descripción de Métrica N° 4.8	82
Tabla 3.37 Descripción de Métrica N° 4.9	83
Tabla 3.38 Relación entre atributos y criterios del Componente N° 4	83
Tabla 3.39 Ejemplo de evaluación de calidad según QuAM.....	85

Tabla 4.1 Población y Muestra en el estudio de caso N° 1	100
Tabla 4.2 Preguntas de la Encuesta de interacción	100
Tabla 4.3 Aspectos positivos y negativos del Estudio de usabilidad	107

Capítulo 1 - Introducción

*L*a 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 (PYMES) son muy difíciles de implementar ya que ello implica una gran inversión en dinero, tiempo y recursos.

Para mejorar, entonces, los procesos de desarrollo de software en PYMES, y a fin de facilitar la adopción de prácticas ágiles que aseguren la calidad de dichos procesos, se presenta en esta tesis a Agile Quality Framework (AQF), una propuesta que integra un modelo de calidad junto a una herramienta de software que permite la automatización de dicho modelo y que se adapta a las características de las PYMES.

AQF se centra en el seguimiento de proyectos de software desarrollados bajo algún enfoque ágil, permitiendo la evaluación de calidad del proceso que subyace en base a componentes que se irán exponiendo y explicando a lo largo de esta tesis doctoral.

Este capítulo se estructura como se describe a continuación: en la sección 1.1 se realiza una breve presentación a los conceptos básicos y términos relacionados que dieron origen a esta tesis. Luego, en la sección 1.2, se detallan los objetivos y la hipótesis de esta tesis doctoral, teniendo en cuenta la importancia de la evaluación de calidad asociada a los proyectos ágiles en empresas PYMES dedicadas al desarrollo de software. Seguidamente, en la sección 1.3, se presentan los factores que motivaron al desarrollo de este trabajo y las contribuciones esperadas. En la sección 1.4, se detalla la metodología de trabajo que se siguió para el desarrollo de la tesis. Y para finalizar, en la sección 1.5, se describe se estructura el resto del documento.

1.1 - Motivación

En Argentina el sector SSI (Software y Servicios Informáticos), se caracteriza por la prestación de servicios intangibles, haciendo uso intensivo del conocimiento y la innovación, principales fuentes de generación de ventajas competitivas [1]. Este sector presenta un alto potencial para generar valor agregado al ecosistema productivo del país, promoviendo la generación de empleo calificado y evidenciando un crecimiento exponencial en los últimos años. De hecho, el notable desarrollo queda demostrado en el último informe publicado por el Observatorio de la Economía del Conocimiento (OEC) de Argentina [2], el cual expresa que durante el primer semestre de 2017 las exportaciones de servicios basados en el conocimiento sumaron USD 2.975 millones, suponiendo un alto incremento interanual en este sector correspondiente al 27,3%.

Por otra parte, se destaca la penetración del sector correspondiente a Software y Servicios Informáticos (SSI) en otras actividades económicas y se observa un claro predominio de las micro, pequeñas y medianas empresas. Así, y según el reporte anual del año 2018 publicado por el Observatorio Permanente de la Industria del Software y Servicios Informáticos (OPSSI) [3], en Argentina, la Industria del Software se compone mayoritariamente por PYMES (Pequeñas y Medianas Empresas), representando casi el 80% del sector, lo que constituye un eslabón fundamental, en el sector, para el país y refuerza la necesidad de llevar adelante iniciativas que contribuyan con el desarrollo y mejora de competitividad de dichas empresas.

En este sentido, y con el objetivo de incrementar la calidad y capacidad de sus procesos y, en consecuencia, la calidad de sus productos y servicios, la mejora de procesos software [4] se convierte en el elemento diferenciador que las empresas del sector necesitan para aumentar su competitividad.

Teniendo en cuenta esta realidad, resulta destacar que varios autores [5][6][7] coinciden en la dificultad que significa para las PYMES implementar programas de Mejoras de Proceso de Software (SPI) fundamentalmente por la falta de seguimiento de los planes de acción y de implantación debido al alto costo que significan. 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.

Sin embargo, no es correcto considerar el estudio e impacto de aquellos elementos que asocian la calidad sólo al producto final, sino que también resulta necesario ajustar los parámetros asociados a los procesos que han facilitado la obtención de este. Tal como lo sostiene Humphrey en [8], la calidad de un producto está determinada por la calidad del proceso utilizado para desarrollarlo. Esto implica que, para gestionar la calidad del producto, resulta necesario gestionar la calidad del proceso utilizado para desarrollar ese producto.

En función a ello, y como punto de partida, es necesario definir la propuesta metodológica que guíe el ciclo de desarrollo del software, para lo cual existen diversas alternativas que

inciden en distintas dimensiones del proceso. Las metodologías más tradicionales se centran especialmente en una rigurosa definición de roles, de las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán [9]. Pero, 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 altos niveles de calidad. Surgen, entonces, enfoques ágiles los cuales persiguen principios como la entrega incremental de funcionalidad nueva al cliente, que se priorizan según el valor de negocio que agrega (de esta forma el producto de software evoluciona en las diferentes entregas), el favorecimiento a la mejora continua y el énfasis en la colaboración cercana entre el equipo de programadores y los expertos del negocio [10].

La adopción del proceso de desarrollo por parte de la empresa impactará directamente en su calidad, otro factor diferenciador cuando se está optando por adquirir un servicio o un bien. De hecho, en el ámbito de la producción de bienes y servicios, y particularmente en la Industria del Software, los aspectos que involucran a la gestión de la calidad resultan cada vez más determinantes a la hora de diferenciar un producto de otro.

El concepto de calidad tiene muchas acepciones, pero una ampliamente aceptada es la establecida por la ISO 9000 [11] que define la calidad como el “grado en el que un conjunto de características inherentes cumple con los requisitos”. Estos requisitos son establecidos por los usuarios y por tanto se puede decir que la calidad es un tema subjetivo dependiente del nivel de satisfacción que el usuario sienta frente al producto utilizado. El software también está sujeto a la evaluación de su calidad, de tal manera que los usuarios puedan establecer el grado con el cual éste satisface sus necesidades.

Tal y como se ha dicho previamente, existe una relación directa entre la calidad del producto y la calidad del proceso de software. Esta relación es expuesta de manera directa o indirecta por diversos autores. Por ejemplo, en [12] se indica que la calidad del proceso contribuye a mejorar la calidad del producto. Adicionalmente, el estándar internacional ISO/IEC 25010 [13] en su Anexo 3, ilustra las relaciones entre las distintas entidades intervinientes en las vistas de calidad, a saber: calidad de procesos, calidad de recursos, calidad de productos, calidad de sistemas, entre otros.

La Figura 1.1 muestra dicha relación gráficamente, donde se puede visualizar que la *calidad de los recursos* influye a la *calidad de los procesos* del ciclo de vida de software. A su vez, la *calidad del producto* de software es determinada por la *calidad del proceso* y de *los recursos empleados*. La *calidad del sistema* está determinada por la *calidad del producto de software* y del resto de los componentes que lo conforman. Por último, *la calidad del sistema* y el *contexto de uso* influyen la calidad que el usuario percibe al utilizar el sistema, esto es la *calidad en uso*. El contexto de uso puede ser definido por un tipo de usuario, tarea y ambiente [14].

Es importante destacar que existen en la literatura diversas acepciones respecto a la calidad del software, entre las más aceptadas se destaca lo expresado por la IEEE respecto a que “la calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario” [15]. Por su parte Pressman [16], se refiere a la calidad del software como “la concordancia con los requisitos funcionales y de rendimientos explícitamente establecidos, estándares de desarrollo explícitamente documentados y características implícitas que se espera de todo software desarrollado profesionalmente”. Sommerville [17], por su parte, sostiene que la calidad del software es un concepto complejo que no es directamente comparable con la calidad de la manufactura de productos.

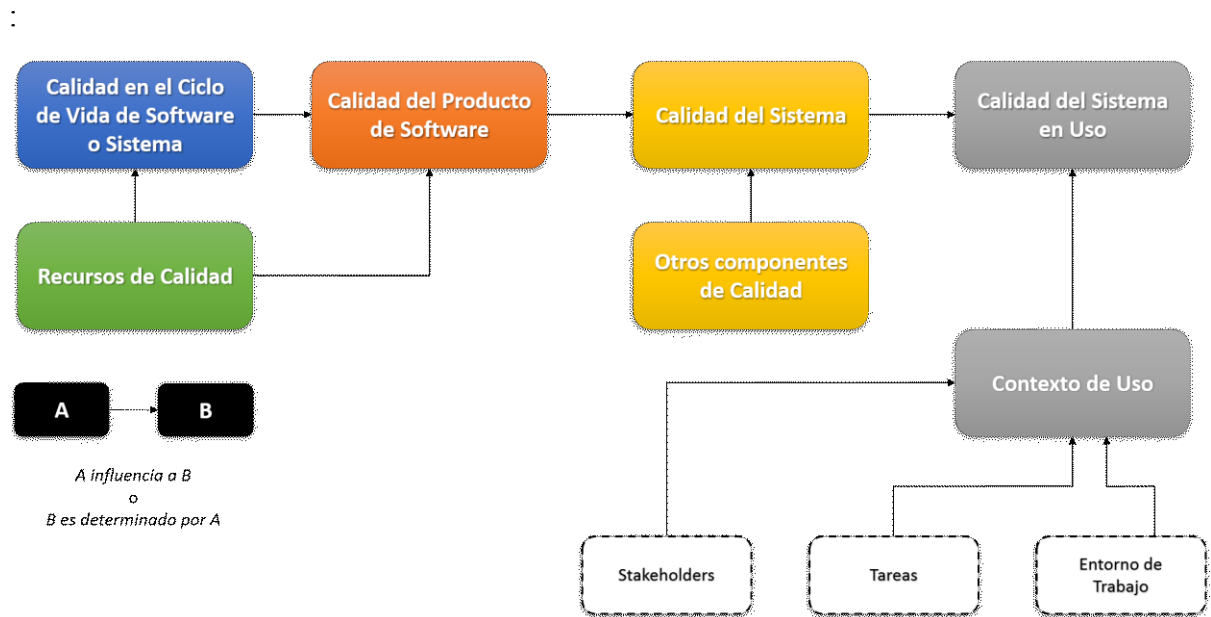


Figura 1.1. Entidades involucradas en las vistas de calidad y sus relaciones [ISO/IEC 25010, 2011]

Fuente: De elaboración propia

La necesidad de desarrollo rápido de aplicaciones de alta calidad ha llevado a que la Industria de Software otorgue gran importancia al concepto de calidad en todas las etapas del ciclo de vida. Por lo que resulta fundamental establecer un marco de trabajo que facilite la gestión y mejora continua de la calidad de los procesos de software.

En primer lugar, es necesario definir un nuevo modelo de calidad que, a diferencia de los existentes, se adapte a las necesidades de las PYMES proporcionando un marco y un lenguaje para comunicarse, como así también un estándar y la experiencia necesaria en el tema a abordar.

En el sentido más general, un *modelo de calidad* describe las características que componen la calidad del software y sus relaciones, en base a un conjunto de características y factores [18]. Estos últimos pueden ser medidos directa o indirectamente a través de la definición de criterios de calidad.

Por un lado, los *factores de calidad* o *atributos externos* representan la calidad desde el punto de vista del usuario. Y, por otro lado, los *criterios de calidad* o *atributos internos* son aquellos

en los que se descomponen los diferentes factores, y representan la calidad desde el punto de vista del producto.

Para la definición de cada criterio de calidad se utilizan *métricas de la calidad*, representando medidas cuantitativas que indican el grado en el que está presente un atributo en el producto [19]. En cuanto que, para medir estas métricas, varios de los enfoques sobre métodos de evaluación y selección de productos software proponen una fase de evaluación donde se deben caracterizar los posibles atributos candidatos bajo un conjunto de propiedades o características y de otros componentes como subcaracterísticas (que en definitiva representa el modelo de calidad).

Lo anteriormente expuesto puede resumirse en una jerarquía que se representa gráficamente en la Figura 1.2. De hecho, teniendo en cuenta la calidad desde el punto de vista del producto o del proceso, se aprecia que la literatura ofrece diversos modelos de calidad entre los cuales se encuentra el más antiguo McCall [20], del cual derivan otros, como el de Boehm. En ambos, se define a la calidad como un conjunto de elementos organizados jerárquicamente tal como se expone en la Figura 1.2, a continuación.

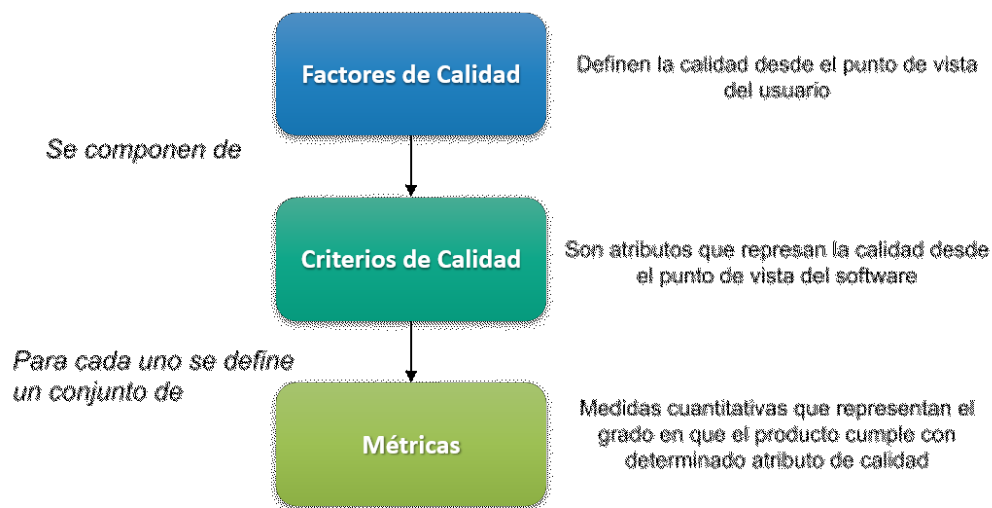


Figura 1.2 Elementos de modelos de calidad según McCall y Boehm.

Fuente: de Elaboración propia

Sin embargo, para evaluar la calidad del software no es suficiente contar solamente con un modelo de calidad, si no que resulta necesario disponer, también, de una herramienta que posibilite automatizar la gestión de los elementos del modelo y analizar los resultados obtenidos a partir de diversos casos evaluados.

Volviendo a la realidad de las PYMES de la región, se observa que, dichas empresas, han iniciado la adopción de prácticas ágiles en sus proyectos con el objetivo de mejorar sus procesos de desarrollo, para incrementar la calidad de sus productos y para poder mantenerse en el mercado y lograr un crecimiento constante. Estos enfoques ágiles se caracterizan por prácticas guiadas por 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 [21].

En términos prácticos, si bien existen numerosos estudios que demuestran una creciente adopción de estos enfoques en empresas de software [22], el rápido crecimiento de la agilidad ha generado confusión, malas interpretaciones e incluso efectos negativos en el desarrollo de algunos proyectos de software [23]. Este escenario implica, por lo tanto, la búsqueda de 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.

Por todo lo expuesto previamente, y en el contexto de esta tesis doctoral, se presenta aquí el diseño, desarrollo y validación de un marco de trabajo que permita evaluar la calidad cuando se opta por trabajar con procesos ágiles de desarrollo de software que se adapte a la realidad expuesta sobre las PYMES. 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 que brinda soporte a la gestión de dicho modelo a través de una aplicación web denominada QuAGI (Quality AGIle), cuyo objetivo es automatizar la gestión y seguimiento de proyectos en entornos ágiles de desarrollo, permitiendo constantes evaluaciones de calidad de los proyectos y ofreciendo información útil para la toma de decisiones referidos a ajustes del proceso de software observado.

1.2 – Hipótesis y Objetivos

Se ha definido para esta tesis la **Hipótesis** de investigación como sigue:

“Será factible mejorar la calidad de los procesos de desarrollo de software guiados por prácticas ágiles a través de la evaluación automática de la calidad de dichos procesos utilizando un framework compuesto por un modelo de calidad y una herramienta que automatice la gestión de dicho modelo”

En base a la hipótesis establecida, el **Objetivo General** de esta tesis doctoral consiste en: *“Proponer un framework que facilite el seguimiento de proyectos y la evaluación de calidad de procesos en empresas PYMES que implementen prácticas ágiles en el desarrollo de software”*.

Para conseguirlo, se establecieron los siguientes objetivos específicos:

1. Realizar un estudio de trabajos previos relacionados a modelos de calidad y herramientas de software aplicables a procesos ágiles en contextos de empresas PYMES.
2. Proponer un modelo de evaluación de calidad sobre procesos de desarrollo de software guiados por prácticas ágiles, en base a la definición de un conjunto de componentes.
3. Diseñar e implementar herramientas de software que soporten la gestión de componentes del modelo de calidad e integren un framework que evalúe la calidad en procesos ágiles.
4. Analizar y estudiar el comportamiento del framework propuesto utilizando un método de validación sobre entornos reales de producción de software.

1.3 - Contribuciones

Las principales contribuciones que se buscan aportar con esta tesis doctoral son:

- Propuesta de un nuevo *modelo de calidad*, QuAM (Quality Agile Model), obtenido luego de realizar un estudio pormenorizado de modelos existentes, que incluye la definición de un conjunto de componentes organizados en *métricas, atributos y criterios* diseñados especialmente para su aplicación en proyectos de software ágiles (es decir desarrollados siguiendo prácticas ágiles).
- Diseño y desarrollo de una *herramienta de software*, QuAGI (Quality AGIle), que de soporte automatizado a la gestión de componentes de QuAM y permita, no solo el seguimiento de proyectos ágiles, sino también evaluación de calidad del proceso de desarrollo.
- Diseño e implementación del framework AQF (Agile Quality Framework), integrado por el *modelo de calidad* junto a la *herramienta* que integra la gestión de componentes del nuevo modelo de calidad junto a herramientas de software que le dan soporte a cada uno de los procesos requeridos. La *herramienta* incluye (i) una *plataforma web* que permite realizar el seguimiento y evaluación de proyectos desarrollados bajo prácticas ágiles (ii) la gestión de componentes de un *modelo de calidad* y (iii) un *agente inteligente* que ofrecerá recomendaciones respondiendo a eventos determinados en el entorno de cada proyecto.
- Definición de estrategias y mecanismos de vinculación que permitan adaptar y flexibilizar el uso del framework a la realidad de cualquier empresa de desarrollo de software. La vinculación se materializa a través de procesos de validación en los que se involucre a empresas PYMES a fin de hacerlas partícipes del proceso de construcción del framework y de verificar la adaptabilidad de la herramienta a cada contexto.

Cabe destacar que esta tesis doctoral se encuentra enmarcada en la realización de los siguientes proyectos de investigación:

- “Framework para la evaluación de Calidad del Software”, que fue financiado por UTN y ejecutado en el Grupo de Investigación en Calidad de Software (GICS) dependiente de la Secretaría de Ciencia y Tecnología de la Facultad Regional Resistencia, con el código EIUTIRE0002205TC en el período 2015-2017.
- “Evaluación de Calidad en Procesos Ágiles de Desarrollo de Software”, que fue financiado por la UTN y fue ejecutado en el Centro de Investigación aplicada en TICS (CInApTIC) dependiente de la Secretaría de Ciencia y Tecnología de la Facultad Regional Resistencia, con el código IAI4445TC de UTN, en el período 2017-2019.
- Asimismo, las actividades desarrolladas en el marco de la tesis fueron compartidas con el Proyecto de Desarrollo Tecnológico y Social (PDTS) “Aporte a la competitividad de las empresas de desarrollo de Software del NEA”, aprobado por el Consejo Interuniversitario Nacional (CIN) en su Convocatoria 2014.

1.4 - Enfoque Metodológico

Se describe, aquí, el enfoque metodológico, las técnicas de recolección y análisis de datos, así como el procedimiento que se siguió para la realización de esta investigación.

Teniendo en cuenta las características de esta tesis, luego de analizar modelos tradicionales y otros más específicos, el método de investigación que se sigue en esta tesis es una adaptación del propuesto por Marcos y Marcos en [24] para la investigación en Ingeniería del Software. El método genérico es un método general de trabajo, que se observa en la Figura 1.3, basado en los pasos a seguir, según Bunge [25], en cada investigación científica. Aunque estos pasos se basan en el *método deductivo hipotético*, debido a su generalidad, son aplicables, con ciertas modificaciones, a cualquier tipo de investigación e incluso a la resolución de cualquier tipo de problema que intentemos resolver. En forma gráfica se presenta en la Figura 1.3 los pasos que conforman el método.

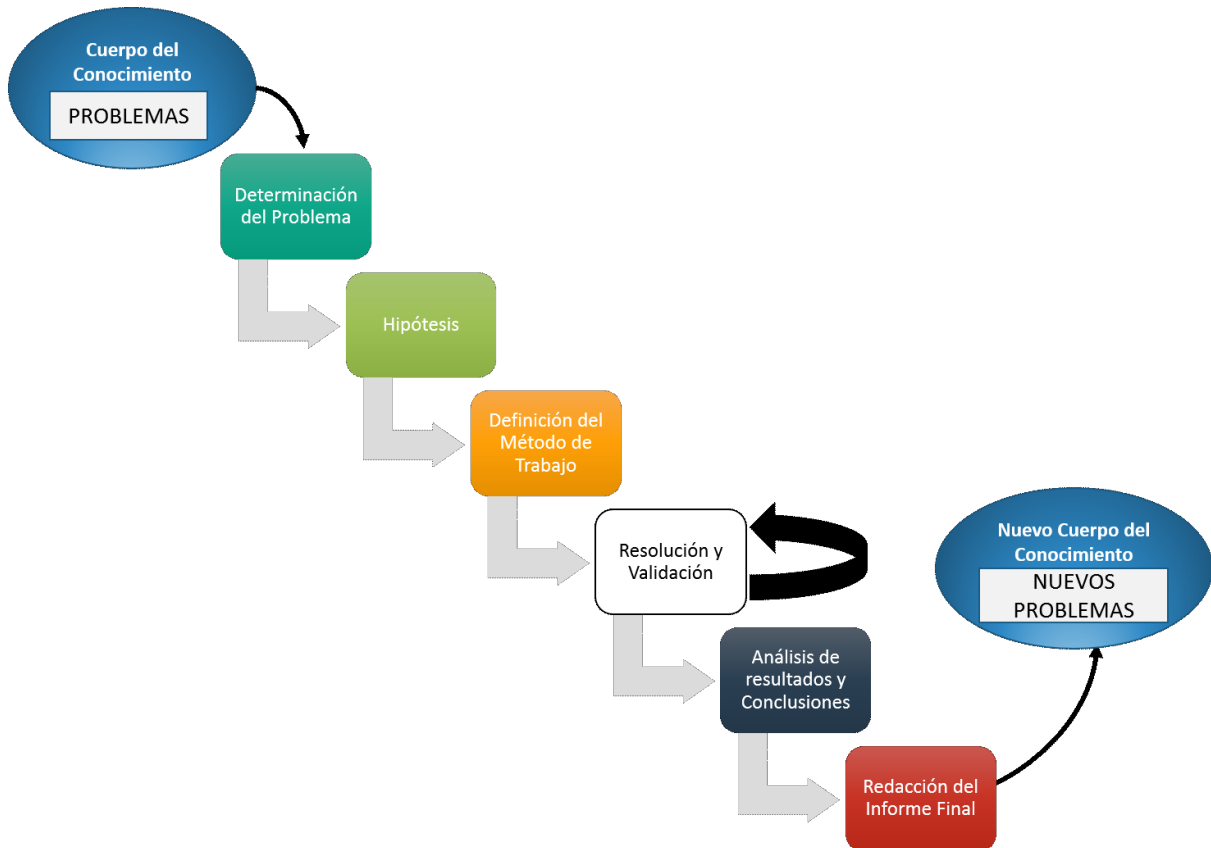


Figura 1.3 Método de Investigación
Fuente: de elaboración propia

Resulta importante destacar que las etapas se realizan de forma iterativa e incremental, generando retroalimentación entre sí, a fin de obtener mejoras en el proceso y realizar los ajustes que se consideren necesarios. Además, como se puede ver en la Figura 1.5 la

definición del método de investigación resulta ser un paso más, pues los autores en [24] recomiendan definirlo de esta manera, ya que la naturaleza de cada investigación tiene sus propias características y, por lo tanto, no sería conveniente aplicar un único método universal de investigación.

A continuación, se describe particularmente el proceso seguido para la etapa de *Resolución y Validación*.

1.4.1 Etapa de Resolución y Validación

El método de *Resolución y Validación* seguido en esta tesis se basa en la teoría de la Ingeniería de Software Empírica, es decir la rama de la Ingeniería de Software que se basa en la experimentación como método para corresponder ideas o teorías con la realidad, la cual refiere a mostrar con hechos las especulaciones, suposiciones y creencias sobre la construcción de software [26]. Por un lado, la *Resolución* comprende todas las actividades que conducen a lograr el producto final y que, en el caso de esta tesis, ha significado la obtención del framework AQF. Y por otro lado la *Validación* implica el proceso del estudio, análisis y observación de la implementación del framework sobre proyectos ágiles reales.

De esta forma, se utiliza durante el desarrollo de la tesis los dos enfoques que plantea la Ingeniería de Software Empírica:

- En primer lugar, el **enfoque cualitativo**, que se basa en estudiar la naturaleza del objeto y en interpretar un fenómeno a partir de la concepción que las personas tienen del mismo, para estudiar la situación actual de la población respecto a la evaluación de calidad del proceso de desarrollo de software en proyectos ágiles.
- Y luego el **enfoque cuantitativo**, el cual se basa en cuantificar factores en estudios de casos que permitan validar la implementación del framework propuesto y obtener conclusiones a partir del análisis estadístico de los resultados.

En síntesis, se usa el enfoque cualitativo como base para definir la hipótesis que luego se corresponderá, a través del enfoque cuantitativo, con la realidad vista desde los estudios de casos.

En ambos enfoques se implementan dos de las técnicas que la Ingeniería de Software Empírica para el proceso de *Resolución y Validación*: la **encuesta** y el **estudio de casos**.

Las **encuestas** se han realizado sobre una muestra representativa de la población, y luego los resultados han sido generalizados al resto de la población [27]. Su utilización ha permitido obtener datos tanto para el diagnóstico de la situación actual y a posteriori para el estudio de la reacción de la población analizada al implementar el framework AQF en sus proyectos.

Durante la etapa de *Validación* se han realizado **estudios de casos**, un instrumento adecuado para muchos tipos de investigación de ingeniería de software, ya que los objetos de estudio (en este caso proyectos ágiles de software), son fenómenos contemporáneos, que son difíciles

de estudiar de forma aislada [28]. Este método ha permitido la observación de diversas variables y su comportamiento en cada uno de los equipos donde se han puesto los proyectos ágiles bajo evaluación del framework AQF.

Klein y Myers definen 3 tipos de estudios de caso, según el objetivo de investigación: *positivista*, *crítico* y de *interpretación* [29]. Para esta tesis se trabajan estudios de caso del tipo *positivistas*, es decir aquéllos que buscan evidencias en determinado contexto, miden variables, prueban hipótesis y permiten extraer inferencias a partir de una muestra en una población definida. Se busca, por lo tanto, estudiar el impacto del uso de AQF en equipos trabajando sobre proyectos ágiles de software que, hasta el momento de la experiencia, no consideraban a la calidad de su proceso de desarrollo como estrategia para la mejora continua.

Cada uno de los estudios de caso que se realicen, en el marco de este trabajo de investigación, seguirán los 5 pasos recomendados por Runeson y Host en [30]:

1. Diseño del estudio de caso: se definen los objetivos y se planifica el estudio de caso.
2. Preparación para la recopilación de datos: se definen los procedimientos y protocolos para la recopilación de datos.
3. Recolección de evidencia¹: ejecución con recolección de datos sobre el caso estudiado.
4. Análisis de los datos recopilados: estudio de los datos recolectados junto a la observación de variables y fenómenos ocurridos durante el caso.
5. Conclusiones: A través de realización de diversos informes estadísticos y cualitativos luego del análisis de los datos, se obtienen conclusiones finales respecto al estudio de caso correspondiente.

Cabe destacar también que los estudios de caso no se realizaron en simultáneo, por lo que el carácter iterativo del proceso de *Validación* permite que los resultados de cada uno de los estudios de casos sirvan como punto de control de lo realizado hasta el momento, pudiendo realizar los ajustes que se consideren necesario tanto sobre el modelo como sobre la herramienta que conforman el framework AQF para un próximo estudio.

¹ Evidencia: Registro de resultados y conclusiones obtenidos a partir de ello [31]

1.5 - Estructura de la Tesis

En esta sección se describe la estructura de esta tesis doctoral, presentando una breve reseña de cada uno de los capítulos subsiguientes.

En el Capítulo 2 se describe con mayor profundidad el estado del arte presentando los resultados de un proceso de revisión sistemática respecto a lo existente en la literatura sobre evaluación de calidad en procesos ágiles de desarrollo de software.

Luego, el Capítulo 3 presenta el framework propuesto, al que se ha denominado AQF (Agile Quality Framework). Desde una visión integral, y con suficiente nivel de detalle, se describe, por un lado, el modelo de calidad QuAM junto a sus componentes y metodología de evaluación propuesta. Y, por otro lado, se presenta QuAGI como herramienta que da soporte a la automatización de los componentes de QuAM permitiendo, no solo el seguimiento de proyectos de software, sino también la evaluación de calidad del proceso ágil que subyace a la obtención del producto de software.

Durante el transcurso de esta tesis se realizaron diversas experiencias a través de la ejecución de estudios de casos con el objetivo de validar la aplicabilidad del framework propuesto en entornos reales. En el Capítulo 4 se describe todo el proceso de ejecución de los estudios de casos y se discuten los resultados obtenidos.

Finalmente, en el Capítulo 5 se presentan las conclusiones de este trabajo, analizando y repasando los objetivos planteados y las contribuciones científicas logradas, e incluyendo el planteo de posibles trabajos futuros.

También, se incluyen en esta tesis, anexos que amplían y respaldan el presente documento:

- **Anexo A.** Incluye la lista de estudios considerados para el proceso de revisión sistemática que se describe en el Capítulo 2. -
- **Anexo B.** Enumera las preguntas que componen la entrevista realizada a las organizaciones para describir el diagnóstico actual que se analiza en el Capítulo 3.
- **Anexo C.** Incluye los resultados obtenidos luego de realizar la encuesta descrita en el Anexo B, respecto a datos demográficos que sirven de fuente de información para el análisis preliminar realizado en el Capítulo 3.
- **Anexo D.** Incluye la descripción de las funcionalidades de QuAGI, y su implementación a nivel de interfaces luego de su presentación en el Capítulo 3.
- **Anexo E.** Incluye las características de la implementación técnica como información complementaria a lo presentado en el Capítulo 3.
- **Anexo F.** Incluye el Formulario de seguimiento utilizado en el estudio de caso N° 3 presentado en el Capítulo 4
- **Anexo G.** Incluye la Encuesta de usabilidad que permite el análisis de resultados de evaluación presentado en el Capítulo 4.

Capítulo 2 - Estado del Arte

***E**n este capítulo se presenta el estudio realizado sobre investigaciones que contribuyen al estado del arte de esta tesis y que describen cuestiones asociadas a la evaluación de calidad y agilidad de procesos de software, cuyas limitaciones justifican la necesidad de afrontar el desarrollo de AQF.*

Para presentar el estado del arte, en la sección 2.1 se incluye la planificación, ejecución y resultados de un proceso de revisión sistemática de la literatura [32] con el objetivo de obtener el estado actual de metodologías de evaluación de calidad implementadas sobre procesos ágiles desarrollo de software. Luego en la sección 2.2, se presenta conclusiones respecto a la convergencia entre los dos tópicos abordados, evaluación de calidad y procesos ágiles.

2.1 - Revisión Sistemática

Para el desarrollo de la revisión sistemática que se presenta en este capítulo se ha diseñado una guía en base al protocolo recomendado por Kitchenham en [33], y que incluye las etapas que se muestran, a continuación, en la Figura 2.1.



Figura 2.1 Etapas del proceso de revisión sistemática utilizado
Fuente: de elaboración propia

La principal ventaja de este método es que genera una visión holística del tema a investigar lo que permitirá decidir en qué áreas concretas del campo es interesante abordar una revisión sistemática más detallada [34][35].

2.1.1 Formulación de la pregunta

De acuerdo con lo establecido por Kitchenham en [36] para la formulación de la pregunta de investigación se deben tener en cuenta cuatro componentes que, en el caso de esta tesis, se definen como sigue:

1. *Población*: Conjunto de trabajos y documentos que describen los estudios sobre la evaluación de la calidad en proyectos de software guiados por procesos ágiles.
2. *Intervención*: Modelos de evaluación, metodologías, marcos de trabajo, herramientas de software que integren marcos de trabajo.
3. *Diseño del estudio*: Experiencias reales, estudio de casos, diseño de validación, la investigación-acción.
4. *Resultado*: Análisis y conclusiones que aporten evidencias respecto al estudio.

En base a los componentes antes enumerados, se formuló la siguiente pregunta global de investigación:

“¿Existen evidencias de iniciativas relacionadas al proceso de evaluación de calidad de procesos ágiles y su aplicación en proyectos de software que se hayan llevado a cabo?”.

Interesa enfocar el estudio en conocer si existen evidencias en la literatura que demuestren la relación entre prácticas ágiles y la evaluación de calidad del proceso de desarrollo de

software. Por ello, resulta necesario, a fines de esta tesis, analizar estudios que no solamente realicen una propuesta, sino que además propongan casos de validación por medio de experiencias reales.

A partir de la pregunta principal, se definieron subpreguntas de investigación (SP) que permiten profundizar el estudio del tema de investigación y que se describen en la Tabla 2.1.

Tabla 2.1. Sub preguntas de investigación

Sub preguntas	Motivación
SP 1 - ¿Cuál es el ámbito de aplicación de la propuesta?	Permite identificar el ámbito de aplicación de la investigación analizada: PYME de Software y/o Equipo de Desarrollo de Software y/o Proyecto ágil de Software específico
SP 2 - ¿Cuál es la contribución de los trabajos relevados relacionados al tema de tesis?	Permite identificar las técnicas y soluciones que se han propuesto: Diseño y/o Modelo y/o Herramientas de Software y/o Framework
SP 3 - ¿Cuál es el tipo de estudio presentado?	Permite determinar si el estudio presentado Estudio de caso y/o Resultados de Cuestionarios/Entrevistas y/o Propuesta de investigación-acción

Fuente: de Elaboración propia

Siguiendo el protocolo para ejecutar el proceso de búsqueda, se deben definir un conjunto de términos o palabras claves, que en este caso está compuesto por:

*Software Quality, Quality Evaluation, Agile Practices, Agile Projects,
Agile Software Development, Agile Software*

Resulta importante aclarar que las palabras se escriben en idioma inglés por ser el idioma predominante en las publicaciones científicas existentes.

2.1.2 Selección de las fuentes

Para la ejecución de la cadena de búsqueda se utilizaron las siguientes librerías digitales como fuentes de datos (Nombre[acrónimo]:website):

- Science Direct[SCIENCE]:<https://www.sciencedirect.com/>,
- ACM Digital Library[ACM]:<http://portal.acm.org/> ,
- IEEEExplore[IEEEEX]:<http://ieeexplore.ieee.org/> ,
- Springer Link[SPRINGER]:<https://link.springer.com/> . ,
- Google Scholar[GOOGLE]:<https://scholar.google.com.ar/> . ,

La inclusión de Google Scholar puede resultar redundante dentro del listado citado anteriormente, sin embargo, no puede excluirse porque tal como lo afirma Noruzi en [37] se considera líder entre los motores de búsqueda para la comunidad de investigación académica

y científica. Además, indexa fuentes con documentación técnica que no aparecen disponibles en otras bibliotecas científicas digitales.

A partir del listado de palabras claves expuesto anteriormente, y teniendo en cuenta el conjunto de librerías digitales tomadas como fuentes de datos, se procedió a la obtención de diferentes cadenas de búsqueda en función a la sintaxis utilizada por cada fuente. Esta combinación y generación de cadenas de búsqueda se expone en la Tabla 2.2.

Tabla 2.2. Cadena de búsquedas utilizadas en cada fuente de datos

Fuente	Cadena de Búsqueda
SCIENCE	Software Quality and Quality Evaluation and Agile Practices and Agile Projects and Agile Software Development and Agile Software
ACM	Software Quality + Quality Evaluation + Agile Practices or Agile Projects + Agile Software Development or Agile Software
IEEE X	(((((Software Quality) AND Quality Evaluation) AND Agile Practices) OR Agile Projects) AND Agile Software Development) OR Agile Software)
SPRINGER	Software Quality and Quality Evaluation and Agile Practices and Agile Projects and Agile Software Development and Agile Software
GOOGLE	Software Quality + Quality Evaluation + Agile Practices + Agile Projects + Agile Software Development + Agile Software

Fuente: de elaboración propia.

2.1.3 Selección de los estudios

Una vez definidas las cadenas de búsqueda sobre las fuentes de información primarias, se debe considerar la posibilidad de que cierto número de estudios obtenidos en la búsqueda no aporten ninguna evidencia relacionada a las preguntas de investigación propuestas. Por ello, para una selección efectiva de estudios, se definieron un conjunto de criterios de inclusión y exclusión [38][39], a partir de los cuales se podrá determinar qué estudios serán incluidos en el proceso de análisis y cuáles no.

Entonces, sobre los estudios filtrados que se obtengan de la primera ejecución de cadenas de búsqueda se aplicarán los *criterios de inclusión* que se listan a continuación:

- I1: Trabajos publicados en el período 2010-2019 (incluyendo los extremos) ¹
- I2: Trabajos publicados que utilicen los conceptos como atributos en el título, resumen (abstract) y palabras claves.
- I3: Trabajos publicados en Congresos y Revistas o como Tesis Doctorales.
- I4: Trabajos publicados en formatos que no sean presentaciones o esquemas.

¹ Se estableció como rango de estudio los últimos diez años, por considerarse un lapso adecuado respecto a antigüedad de investigación en la temática abordada.

- I5: Trabajos publicados que muestran reportes de experiencia o casos de estudio sobre evaluación de calidad en procesos ágiles de software

Aplicados estos criterios, se determina si los artículos encontrados dan respuesta a la pregunta formulada para la revisión sistemática y se los incluye en el conjunto de estudios primarios de la misma. En el caso de algunos artículos además de hacer foco en el título y abstract como indica el criterio de inclusión I2, para definir su pertinencia es necesario leer, además, la introducción, para entender claramente sobre la temática que cada uno aborda. Así, y en función a este análisis más detallado, pueden surgir nuevos documentos para ser incluidos en la revisión sistemática.

Además, resulta necesario definir, los *criterios de exclusión* para no considerar aquellos estudios que no sean útiles para los fines de esta tesis. Se listan dichos criterios a continuación:

- E1: Trabajos que no sean artículos científicos publicados.¹
- E2: Trabajos publicados como revisiones sistemáticas o estado del arte respecto a los conceptos claves.²
- E3: Trabajos publicados que incluyen estudios iniciales o diagnósticos sobre el estado actual de la calidad en procesos ágiles de software.
- E4: Trabajos publicados que incluyen estudios basados simplemente en la opinión de expertos sin referenciar una experiencia específica o validación real.

Cabe destacar también que la literatura gris, es decir aquella que se emite por canales no comerciales de distribución y es de difícil acceso [40], se excluyó de las búsquedas por dos razones: es difícil evaluar su calidad y las búsquedas contendrían un elevado número de resultados.

2.1.4 Valoración de la calidad

La *valoración de la calidad* de los estudios individuales que se incluyen en las revisiones sistemáticas es necesaria para limitar los sesgos, formarse una idea más precisa de las potenciales comparaciones y guiar la interpretación de los resultados [41].

Para realizar la valoración de calidad de los artículos seleccionados en este estudio se utilizó una evaluación de tipo cuantitativa, basada en cuatro preguntas acorde a lo que recomiendan Kitchenham y Charters en [42]. El procedimiento de puntuación usado en la evaluación fue Excelente (EX) = 3, Muy Bueno (MB) = 2, Regular (RE) = 1 o Malo (MA) = 0. Las preguntas de verificación definidas en la presente revisión sistemática fueron:

¹ Es común que las fuentes digitales de búsqueda arrojen como resultados documentos tipo presentaciones audiovisuales o portadas de anales de congreso que no son afines a los objetivos del presente estudio.

² Se requieren estudiar propuestas científicas-tecnológicas en etapas más avanzadas y no al inicio de la investigación.

- VC1: ¿Qué tan bien aborda el estudio sus objetivos y propósitos originales?
- VC2: ¿Qué tan bien definidos están el diseño de la muestra / selección objetivo de casos / documentación del estudio de caso?
- VC3: ¿Qué tan clara es la relación entre los datos recolectados, la interpretación de estos y las conclusiones a la que arriba el estudio?
- VC4: ¿Qué tan bueno es el estudio comparado con otros relacionados?

2.1.5 Extracción de la información

La última etapa planificada se relaciona al proceso de extracción de datos que se resume al conjunto de estudios seleccionados que dan respuesta a las preguntas de investigación.

En primer lugar, se procede, a partir de las cadenas de búsqueda previamente diseñadas y enumeradas en la Tabla 2.2, a ejecutar automáticamente la búsqueda en cada librería digital. El siguiente paso consiste en almacenar y enumerar todos los estudios obtenidos por cada fuente de datos. Esto implicará, asimismo, la extracción de datos tales como título, abstract y año de publicación por cada artículo.

Luego, se procederá a la aplicación de criterios de inclusión definidos previamente en este capítulo, y luego con ese conjunto de artículos se realizará el filtro eliminando los estudios duplicados.

Sobre esa selección se aplicarán, los criterios de exclusión, de forma tal de refinar la selección y obtener el conjunto de estudios primarios listos para su análisis y posterior valoración de calidad. Los resultados de la fase de extracción de la información se exponen a continuación.

2.1.5.1 Resultados del proceso

Una vez planificado el proceso de revisión sistemática, se dio lugar a su ejecución, cuyos resultados se presentan en esta sección.

2.1.5.1.1 Ejecución de la Búsqueda y selección de estudios primarios

Tal como se indicó en la introducción de este capítulo, el primer paso en la ejecución de la revisión sistemática fue la búsqueda automática de los estudios de acuerdo con las cadenas definidas en la Tabla 2.2 conforme a las características de cada una de las librerías digitales consideradas como fuentes de datos.

La cantidad de resultados obtenidos luego de la ejecución de la búsqueda se discriminan en la Tabla 2.3, la cual incluye en su primera columna la fuente de dato, luego en la segunda columna se muestra la cantidad de artículos encontrados y en la tercera columna la cantidad de estudios analizados.

Tabla 2.3. Cantidad de Resultados de la ejecución de búsqueda

Fuente	Estudios Obtenidos	Estudios Analizados
SCIENCE	4727	50
ACM	375167	50
IEEEEX	4178	50
SPRINGER	6350	50
GOOGLE	82100	50
TOTAL		250

Fuente: de elaboración propia

Cabe destacar que por la cantidad de resultados arrojados en la ejecución de las cadenas de búsqueda se ha decidido aplicar un refinamiento, seleccionando, entonces, los primeros 50 artículos de las fuentes que superasen los 1000 estudios, todas en este caso, por considerarse este un límite aceptable de estudios basados en la experiencia empírica en investigación.

Luego se procedió a la aplicación de los criterios de inclusión sobre la selección anterior de estudios, obteniéndose, como se muestra en la Tabla 2.4, un total de 62 estudios relevantes de los 250 analizados lo que representa aproximadamente un 25% de relevancia. A posteriori se eliminaron los estudios duplicados, reduciéndose el conjunto de artículos válidos a un total

de 55. Y sobre este conjunto final se aplicaron los criterios de exclusión ya enumerados previamente, lográndose un total definitivo de 13 estudios a analizar, tal como se resume en la Tabla 2.4.

También se observa en la Tabla 2.4 que, la fuente con mayor relevancia resulta ser Google Scholar, por lo que se puede considerar como la librería digital con mejor precisión de las que se han utilizado. Y, por el contrario, la fuente SCIENCE es la que menos resultados ha arrojado debido a que, si bien las palabras claves formaban parte de los estudios que arrojó la búsqueda, no presentan relación semántica con el objetivo de esta revisión por lo que la mayoría han sido excluidos.

Tabla 2.4. Cantidad de resultados luego de aplicar criterios de inclusión y exclusión, y eliminar duplicados

Fuente	Estudios Analizados	Criterios de Inclusión
SCIENCE	50	2
ACM	50	7
IEEEX	50	15
SPRINGER	50	5
GOOGLE	50	33
TOTAL		62
Eliminados por duplicados		7
TOTAL		55
Criterios de Exclusión		13

Fuente: de elaboración propia

Finalmente, en la Tabla 2.5 se presenta la lista de los 13 artículos que han sido seleccionados como estudios primarios para esta revisión.

Tabla 2.5. Estudios primarios seleccionados para revisión

Estudio #	Título	Autores	Año
G1	Assisting the continuous improvement of scrum projects using metrics and bayesian networks. Journal of Software: Evolution and Process	Perkusich, M., Gorgônio, K. C., Almeida, H., & Perkusich, A.	2017
G2	Quality of software requirements specification in agile projects: A cross-case analysis of six companies. Journal of Systems and Software	Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M.	2018
G3	An Empirical Study on the Role of Requirement Engineering in Agile Method and Its Impact on Quality.	Rahman, A	2015
S1	Patterns of Collaboration Driven by Requirements in Agile Software Development Teams.	Inayat, I., Marczak, S., Salim, S. S., & Damian, D	2017
S2	Agile quality requirements management best practices portfolio: a situational method engineering approach.	López, L., Behutiye, W., Karhapää, P., Ralyté, J., Franch, X., & Oivo, M.	2017
S3	Scrumban/XP: A New Approach to Cover the Third Level of CMMI Model.	Bougroun, Z., Zeaaraoui, A., & Bouchentouf, T.	2016
S4	The Disciplined Agile Process Decision Framework.	Ambler, S. W., & Lines, M.	2016
S5	AGILUS: A Method for Integrating Usability Evaluations on Agile Software Development.	de Freitas, R. C., Rodrigues, L. A., & da Cunha, A. M.	2016
A1	Procedural Assessment Process of Software Quality Models Using Agility.	Al-Elaimat, A., & Al-Ghuwairi, A. R	2015
A2	GuideGen: a tool for keeping requirements and acceptance tests aligned.	Hotomski, S., & Glinz, M.	2018
I1	Formative evaluation of a tool for managing software quality.	Guzmán, L., Vollmer, A. M., Ciolkowski, M., & Gillmann, M.	2017
I2	A reference model for agile quality assurance: combining agile methodologies and maturity models.	Silva, F. S., Soares, F. S. F., Peres, A. L., de Azevedo, I. M., Pinto, P. P., & de Lemos Meira, S. R.	2014
I3	A Quantitative Framework for Task Allocation in Distributed Agile Software Development.	Aslam, W., & Ijaz, F.	2018

Fuente: de elaboración propia

2.1.5.1.2 Valoración de la Calidad de los estudios primarios

Como se ha dicho, se han identificado un total de 13 estudios relevantes de acuerdo al objetivo de esta revisión sistemática y que responden a la pregunta formulada en 2.1.1, en el *Anexo A* de esta tesis se incluye con más detalle información respecto a cada uno de los estudios seleccionados de acuerdo con el formato que se indica en la Figura 2.2.

Título	<i>Se consigna el título de la publicación</i>
Año de Publicación	<i>Se indica año de la publicación.</i>
Autores	<i>Se enumeran los autores de la publicación</i>
Información relevante para el estudio - <i>Se incluye aquí la información complementaria que resulta importante</i>	
Metodología	<i>Se describe brevemente la metodología de investigación y la propuesta</i>
Resultados obtenidos	<i>Se explican, de forma resumida, los resultados logrados y presentados en la publicación</i>

Figura 2.2 Formato utilizado para el registro de información relevante de los estudios primarios
Fuente: de elaboración propia

Una vez identificados los estudios primarios, el siguiente paso consiste en realizar la valoración de calidad de acuerdo con las preguntas definidas en la sección 2.1.4.

A fin de mejorar la selección, se estableció un umbral de calidad para los estudios primarios de forma tal que fueron descartados aquellos estudios con puntaje inferior a 7, es decir que hayan obtenido una valoración menor al 60% del mejor puntaje posible, que en este caso era 12.

De los 13 estudios considerados inicialmente como primarios, 11 (85% de los estudios relevantes no duplicados) resultaron tener un índice de calidad por encima del umbral establecido. El puntaje asignado a cada estudio primario por cada pregunta en función a la validación de la calidad y los estudios eliminados del conjunto (resaltados en rojo) puede verse en la Tabla 2.6.

Tabla 2.6. Valoración de la calidad de estudios primarios relevantes no duplicados

Estudio #	VC1	VC2	VC3	VC4	TOTAL
G1	2	3	2	2	9
G2	2	3	3	2	10
G3	2	2	3	3	10
S1	2	2	3	1	8
S2	1	2	2	1	6
S3	2	2	2	3	9
S4	2	2	1	1	6
S5	3	3	2	3	11
A1	2	2	2	3	9
A2	2	3	3	2	10
I1	2	2	2	2	8
I2	3	2	2	3	10
I3	3	3	2	3	11

Fuente: de elaboración propia

2.1.6 Discusiones

Una vez obtenidos los resultados, se procede a realizar el análisis de estos en función a dos aspectos.

Por un lado, un *análisis cuantitativo* cuyo objetivo es obtener la cantidad de estudios que se adecúan a cada aspecto considerado en las sub-preguntas definidas durante la fase de “*Formulación de la Pregunta*” del proceso de revisión sistemática que aquí se describe. Y, por otro lado, se lleva a cabo un *análisis cualitativo*, el cual tiene como fin

2.1.6.1 Análisis Cuantitativo

En este caso se llevó a cabo una revisión cruzada de casos o cross-case, buscando patrones a través de una profundización respecto a lo propuesto por cada estudio. Para mayor claridad del análisis, se muestran a continuación los resultados de forma cuantitativa que surgen de la comparación entre los diferentes estudios en base al conjunto de sub-preguntas definidas en la Tabla 2.1. Luego se exponen las conclusiones para dar respuesta a la pregunta principal “*¿Existen evidencias de iniciativas relacionadas al proceso de evaluación de calidad de procesos ágiles y su aplicación en proyectos de software que se hayan llevado a cabo?*”.

2.1.6.1.1 Sub-pregunta 1: ¿Cuál es el ámbito de aplicación de la propuesta?

En la Tabla 2.7 se expone la clasificación y comparación entre casos desde el punto de vista del *ámbito de aplicación* de la propuesta o experiencia de los artículos estudiados. Los trabajos en las Tablas se identifican de acuerdo con los acrónimos utilizados en la Tabla 2.5 en la columna “Estudio #” de este capítulo.

Tabla 2.7. Distribución de casos de estudio de acuerdo al ámbito de aplicación de la propuesta

SP 1 - ¿Cuál es el ámbito de aplicación de la propuesta?			
Fuente	Google Scholar		
Artículo	Empresas de Software	Equipo de Desarrollo de Software	Proyectos ágiles de Software
G1			X
G2	X		
G3			X
Fuente	Springer Link		
Artículo	Empresas de Software	Equipo de Desarrollo de Software	Proyectos ágiles de Software
S1		X	X
S3		X	
S5		X	
Fuente	ACM Digital Library		
Artículo	Empresas de Software	Equipo de Desarrollo de Software	Proyectos ágiles de Software
A1			X
A2	X		
Fuente	IEEE Digital Library		
Artículo	Empresas de Software	Equipo de Desarrollo de Software	Proyectos ágiles de Software
I1	X		X
I2			X
I3		X	
TOTALES	4	4	6

Fuente: de Elaboración propia

En este sentido, como se aprecia en la Figura 2.2, aproximadamente el 38% de los artículos presentan propuestas validadas en organizaciones dedicadas a desarrollo de software, casi un 31% lo hace sobre equipos de desarrollo de software en diversas organizaciones y en su

mayoría, con casi un 54% de los trabajos, expone resultados de validación haciendo referencia a determinados proyectos ágiles de software. Sobre estos artículos solo un 5% tipifica la organización entre las que se indican experiencias en Grandes Empresas IT y PYMES.

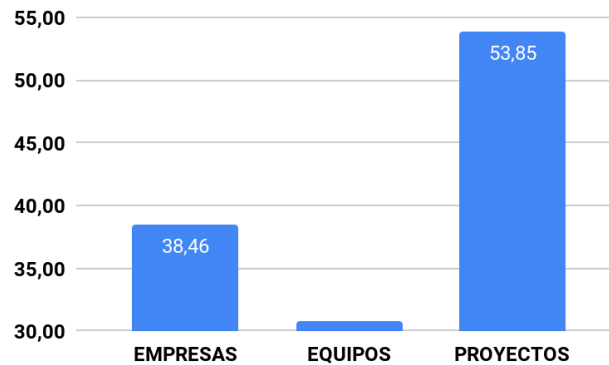


Figura 2.3. Análisis cuantitativo de acuerdo con el ámbito de aplicación
Fuente: de elaboración propia

2.1.6.1.2 Sub-pregunta 2: ¿Cuál es la contribución de los trabajos relevados relacionados al tema de tesis?

Respecto al análisis de la contribución de cada estudio, se exponen los resultados obtenidos en la Tabla 2.8.

Tabla 2.8. Distribución de casos de estudio de acuerdo a la contribución que se presenta

SP 2 - ¿Cuál es la contribución de los trabajos relevados relacionados al tema de tesis?				
Fuente	Google Scholar			
Artículo	Diseño	Modelo	Herramienta de Software	Framework
G1	X			
G2		X		
G3		X		
Fuente	Springer Link			
Artículo	Diseño	Modelo	Herramienta de Software	Framework
S1		X		
S3				X
S5				X
Fuente	ACM Digital Library			
Artículo	Diseño	Modelo	Herramienta de Software	Framework
A1				X

A2			X	X
Fuente	IEEE Digital Library			
Artículo	Diseño	Modelo	Herramienta de Software	Framework
I1				X
I2		X		
I3				X
TOTALES	1	4	1	6

Fuente: de Elaboración propia

Tal como se observa en el gráfico de la Figura 2.3, más del 50% de los artículos presenta un framework. Se observa además que, aproximadamente el 39% de las propuestas presentan la definición de un modelo que contempla elementos para evaluar la calidad de procesos de desarrollo de software. Finalmente, de los 11 estudios seleccionados, sólo 1 artículo hace referencia al diseño de un prototipo que automatiza la evaluación de calidad; y sólo 1 artículo presenta el caso de implementación de una herramienta de software, incluyendo su validación.

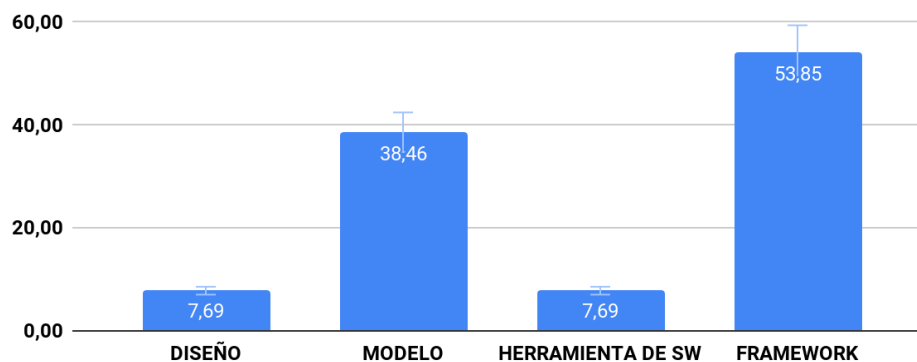


Figura 2.4. Análisis cuantitativo de casos de acuerdo con la contribución
Fuente: de elaboración propia

2.1.6.1.3 Sub- pregunta 3: ¿Cuál es el tipo de estudio presentado?

A continuación, en la Tabla 2.9 se presenta la comparación realizada al analizar los artículos en función al tipo de estudio presentado en cada uno.

Tabla 2.9. Distribución de casos de acuerdo al tipo de estudio

SP 3 - ¿Cuál es el tipo de estudio presentado?			
Fuente	Google Scholar		
Artículo	Casos de Estudio	Resultados de Cuestionario/Entrevista	Propuesta de investigación-acción
G1	X		
G2		X	
G3		X	
Fuente	Springer Link		
Artículo	Casos de Estudio	Resultados de Cuestionario/Entrevista	Propuesta de investigación-acción
S1	X	X	
S3			X
S5	X		
Fuente	ACM Digital Library		
Artículo	Casos de Estudio	Resultados de Cuestionario/Entrevista	Propuesta de investigación-acción
A1			x
A2	X		
Fuente	IEEE Digital Library		
Artículo	Casos de Estudio	Resultados de Cuestionario/Entrevista	Propuesta de investigación-acción
I1	X	X	
I2		X	X
I3			X
TOTALES	7	5	4

Fuente: de Elaboración propia

Por último, al analizar el componente referido a tipo de estudio, tal como se ve en la Figura 2.4, se ha observado que el 46% de los artículos presenta estudios de casos para validar o evaluar la propuesta que describen. Luego, de forma equilibrada se observa que casi un 38%

de los estudios seleccionados presentan resultados de cuestionarios y/o entrevistas a diversos roles miembros de equipos ágiles de desarrollo de software y otro 38% definen propuestas de investigación y acción a ser implementados.

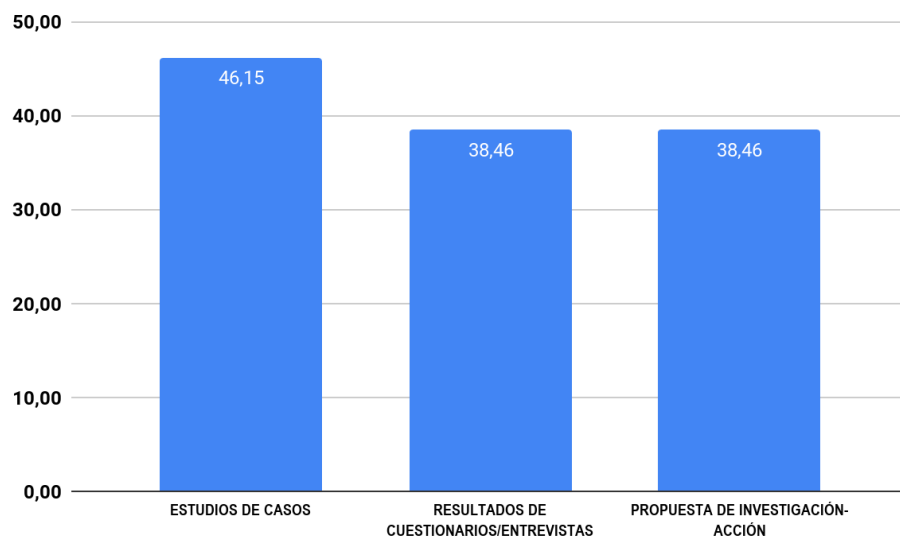


Figura 2.5 Análisis cuantitativo de acuerdo al tipo de estudio
Fuente: de elaboración propia

2.1.6.2 Análisis Cualitativo

Hasta el momento, en este capítulo se ha llevado a cabo la revisión sistemática de la literatura teniendo en cuenta el estudio de propuestas relacionadas al proceso de evaluación de calidad de procesos ágiles y su implementación final. Luego, en base a la aplicación de criterios de inclusión y de exclusión, se han seleccionado un total de 11 estudios para su evaluación respecto a 3 dimensiones de análisis: ámbito de aplicación de la propuesta, contribución del trabajo y tipo de estudio. Entonces, y habiéndose presentado anteriormente los resultados del análisis cuantitativo se exponen aquí, por cada dimensión, los estudios más relevantes para su análisis cualitativo.

De esta forma, respecto al *ámbito de aplicación* de la propuesta se destaca, en primer lugar, el trabajo presentado por Medeiros et al [43] en el cual se define un modelo para determinar los factores de calidad que afectan la especificación de requerimientos cuando se opta por utilizar Desarrollo de Software Ágil (ASD, Agile Software Development). En dicha propuesta se expone el estudio comparativo realizado sobre 6 casos que cubren organizaciones que responden a diversas características (organizaciones gubernamentales, empresas, industrias, entre otros) y concluyendo en una comparación final. Sin embargo, lo presentado contempla un modelo que solamente evalúa calidad durante la elicitación de requerimientos en proyectos ágiles de software sin incluir una evaluación integral sobre el resto de los aspectos del proceso ágil.

En segundo lugar, se destaca la contribución propuesta por Guzmán et al. en [44] que incluye la participación de miembros especializados de 2 empresas para la validación de un prototipo (ProDebt) para la gestión de calidad de software. La evaluación del prototipo en conjunto

con las compañías de software obtuvo resultados relevantes que delinearon las mejoras para una versión posterior. En cuanto a limitaciones, es importante resaltar que ProDebt propone un modelo orientado solo a disminuir deuda técnica a través de medición de factores que impactan directamente a la calidad y, si bien se orienta a procesos ágiles, la evaluación se realiza sobre el producto de software más que sobre el proceso.

Se ha realizado, también, el análisis desde el punto de vista de *contribución de los estudios* y en este sentido se destaca por un lado la propuesta denominada AGILUS [45] que, como contribución, ofrece un método aplicado a la evaluación de usabilidad en el desarrollo de software ágil bajo SCRUM, enfocado en mejorar la calidad del producto y la satisfacción del usuario. El alcance del método AGILUS inicia después de la definición de Product Backlog y finaliza cuando el Product Owner valida los incrementos del producto de software, respaldados por los resultados de las pruebas de usabilidad, que involucran a usuarios representativos del sistema. La propuesta incluye la definición de etapas del proceso de evaluación, los métodos utilizados en cada una y el objetivo a alcanzar. Lamentablemente, y a pesar de su contribución a través de un framework, AGILUS solo se enfoca en evaluación de calidad desde el punto de vista de usabilidad del producto, obtenido en un proceso de software guiado por SCRUM, dejando de lado otros enfoques de desarrollo ágiles.

Por otro lado, se considera relevante respecto a su contribución el framework para distribución de tareas en equipos ágiles, propuesto por Aslam e Ijaz [46], que presenta un modelo a través del cual se determinan factores que influyen en la asignación de tareas en entornos distribuidos de desarrollo de software ágil. La propuesta asigna roles a los miembros del equipo en función de sus capacidades y experiencias específicas. Y si bien aún no fue validado, el framework propone evaluación de calidad del proceso en torno a las tareas del equipo ágil, teniendo en cuenta solamente dicho aspecto lo que resulta su principal limitación. Además, otro inconveniente de la propuesta radica en que la distribución de tareas sólo considera aspectos cualitativos y no cuantitativos, sesgando y limitando a ello la evaluación del rendimiento del equipo.

Finalmente se realizó el análisis cualitativo en función al *tipo de estudio* que llevan adelante cada uno de los artículos seleccionados. Así, se destacan las propuestas de GuideGuen [47] y AGILUS pues en ambos casos las herramientas fueron evaluadas en estudios de casos donde los actores formaban parte de empresas de desarrollo de software.

GuideGen es una herramienta que ayuda a los ingenieros de requisitos, testers y otros stakeholders a mantener alineados los requisitos y las pruebas de aceptación, cuya inconsistencia, según los autores, junto con una comunicación deficiente de los cambios, conducen a problemas de calidad del software, costos imprevistos y retrasos en los proyectos. En este sentido los autores proponen la validación de la herramienta a través de experiencias en contextos reales de proyectos basados en enfoques ágiles y proyectos guiados por metodologías tradicionales. La evaluación se llevó a cabo sobre 262 cambios de requerimientos en 3 organizaciones, 2 de las cuales presentaron proyectos guiados por prácticas ágiles. Como resultado final, se obtuvo en promedio un 80% de acierto en las sugerencias ofrecidas por GuideGen en proyectos ágiles y un 67% en proyectos bajo enfoques tradicionales. En las entrevistas de seguimiento, todos los expertos consideraron

que GuideGen era útil para comunicar los cambios a tiempo y con menos esfuerzo. Sin embargo, y si bien esta propuesta aborda la calidad de software desde el punto de vista del proceso, solo se enfoca en la consistencia entre los cambios en requerimientos y las pruebas de aceptación para su validación.

Respecto a AGILUS, una herramienta a la que ya se ha hecho referencia anteriormente, valida la propuesta a través de un proceso de evaluación con casos reales, y cuyos resultados derivarían en mejoras para su próxima versión. Así, los autores proponen verificar la calidad de uso mediante la validación de AGILUS en base al análisis sobre 2 casos: un Sistema de monitoreo socio-pedagógico (SMS); y un Sistema de Gestión de Eventos (EMS), ambos desarrollados para una institución educativa. De la experiencia han participado seis desarrolladores como voluntarios de cada equipo y veinte voluntarios representando a usuarios finales de cada sistema a evaluarse. En el proceso de validación se han utilizado las siguientes métricas: tiempo de ejecución de tareas para medir eficiencia, error y tasa de finalización para medir efectividad y escala de usabilidad de sistema (SUS) [48] para medir la satisfacción del usuario. De acuerdo con este esquema, los resultados se presentaron en función a la evaluación de calidad en uso y satisfacción de usuario, sin hacer referencia a la calidad del proceso ágil sino evaluando productos obtenidos en 3 etapas (Diseño de Interacción, Verificación de Implementación sobre el producto, Testing con usuarios) que caracterizan, según AGILUS, a ciclos de desarrollo de software bajo SCRUM.

Otro de los artículos que se destaca respecto a la contribución que ofrece, es la propuesta de Perkusich et al [49], en la cual se presenta una red bayesiana para ayudar en la evaluación de la calidad del proceso de software en el contexto de los proyectos Scrum. Las inferencias obtenidas por cada versión de la red bayesiana fueron validadas aplicándose sobre 10 escenarios simulados, diseñados con ayuda de Scrum Masters que participaron en la construcción de cada uno ofreciendo datos respecto a lotes de datos de entrada y resultados esperados. Si bien en el artículo se exponen los resultados obtenidos sobre 1 escenario, el diseño del estudio de caso contempla la configuración de acuerdo con el contexto que se debe evaluar, por ejemplo: Product Owner no especializado con Desarrolladores capacitados, o viceversa o incluyendo requerimientos incompletos, etc. De esta forma, y destacando la posibilidad de adaptar la red bayesiana de acuerdo con el contexto, el principal inconveniente de la propuesta radica en que la interpretación de las inferencias que resultan de la implementación del proceso es subjetiva y depende del punto de vista de quién realice la evaluación.

2.2- Conclusiones

A continuación, y luego de haberse llevado adelante la revisión sistemática con posterior análisis cuali-cuantitativo, se exponen conclusiones respecto al estado del arte.

Luego de la revisión sistemática se comprueba que, aunque investigaciones recientes han demostrado interés sobre la evaluación de calidad de proyectos de software desarrollados bajo enfoque ágil, muy pocos estudios han investigado cuál es el impacto de aspectos de calidad sobre procesos ágiles en equipos, áreas o empresas de desarrollo de software. Hasta el momento, y respondiendo a la pregunta *“¿Existen evidencias de iniciativas relacionadas al proceso de evaluación de calidad de procesos ágiles y su aplicación en proyectos de software que se hayan llevado a cabo?”* que ha guiado el proceso de revisión sistemática en esta tesis, se han identificado propuestas que evalúan la calidad de procesos ágiles, pero solo teniendo en cuenta componentes individuales asociados. Por ejemplo, teniendo en cuenta solo el equipo de desarrollo, evaluando solo los requerimientos del proceso, considerando únicamente al cliente, etc.

Producto de esta revisión sistemática se ha podido concluir, también, que existe en la literatura disponible una diversidad de aproximaciones relacionadas a la evaluación de calidad de proyectos de software guiados por prácticas ágiles. Sin embargo, dichos estudios ofrecen enfoques solamente hacia productos o artefactos obtenidos al finalizar cada iteración del proyecto sin detenerse en aspectos asociados al proceso de desarrollo en sí mismo.

Además, resulta relevante destacar que, si bien todos los estudios se enfocan en Empresas o Equipos de Desarrollo o Proyectos de Software, ninguno identifica aquellos factores que impactan positiva o negativamente sobre la calidad del proceso, ni tampoco ofrecen una visión integradora que permita definir estrategias transversales para mejorar el proceso de la evaluación de calidad que se estudia. En este sentido, se ha establecido para la tesis que aquí se presenta, trabajar sobre una propuesta que inicie con el relevamiento de necesidades en empresas o áreas dedicadas al desarrollo de software, luego considere un esquema de implementación en diversos equipos abocados a proyectos de software y finalmente ofrezca herramientas para la evaluación de calidad que se enfoque, fundamentalmente, en el proceso guiado por prácticas ágiles.

Otro aspecto para destacar se relaciona con el análisis cuantitativo obtenido a partir de la revisión sistemática presentada en este capítulo, en el cual se observa que más del 50% de los estudios desarrollan frameworks para la evaluación de calidad. Sin embargo, se ha podido concluir que estas investigaciones se ven limitadas por falta de información que permita identificar componentes de cada framework. Por ello, se ha considerado conveniente para presentar este trabajo, que AQF ofrezca una propuesta a partir de la integración de un modelo de calidad, un proceso de evaluación y un conjunto de herramientas que permitan automatizar la gestión de componentes. Esto permite obtener mayor detalle del análisis de calidad del proceso ágil que se observa, de manera tal de identificar cuáles son los factores que obstaculizan la ejecución del proyecto, y las prácticas que deberían ajustarse.

Resulta interesante destacar también que, la mayor parte de los estudios, exactamente un 46%, han presentado la validación de su propuesta a través del estudio de casos, sin embargo,

en cada artículo solo se contemplan resultados iniciales sin hacer referencia a la experiencia luego de una implementación de mejoras o ajustes respecto a la primera vez. De hecho, es importante destacar que la mayoría de los estudios seleccionados parecen no haber tenido continuidad en el tiempo por parte de sus autores, o fueron absorbidos por otras líneas de investigación.

En función a lo antes expuesto, una de las principales fortalezas de la propuesta de AQF radica en la validación de cada uno de sus componentes a través del estudio de diversos casos, obteniendo una visión integral de forma incremental hasta obtener una plataforma estable que permita mejorar la calidad de los procesos de empresas PYMES cuando optan por prácticas ágiles de desarrollo de software. Resulta necesario destacar que esta tesis busca trabajar en base al análisis de los procesos ágiles en entornos reales, a través de métodos de obtención de datos de personas, equipos y organizaciones in situ de forma tal de minimizar errores en la exposición de la información, basándose así en la premisa definida por Benbasat et al. al hacer referencia a la importancia de los casos de estudio en investigaciones [50].

Capítulo 3 - AQF: Framework para la Evaluación de calidad de procesos ágiles de software

Tal como se ha dicho en el capítulo 1, el objetivo principal de esta tesis es proponer un framework que facilite el seguimiento de proyectos y la evaluación de calidad de procesos cuyos ciclos de desarrollo implementan prácticas ágiles.

*Dicho framework se denomina AQF (Agile Quality Framework), y la versión que se define en esta tesis, está compuesta por: un **modelo de calidad**, denominado QuAM (Quality Agile Model) descrito a través de una estructura formal y una estrategia de medición de la calidad; una **herramienta de software**, QuAGI (Quality AGIle) que da soporte al modelo, constituyendo, además, un medio para su validación.*

En este capítulo se describe en la sección 3.1 el modelo QuAM explicando cada uno de sus componentes, y luego, en la sección 3.2, se presenta la herramienta QuAGI que da soporte a dicho modelo. Finalmente, en la sección 3.3 se exponen conclusiones generales al respecto.

3.1 – Introducción

El framework AQF (Agile Quality Framework), que se presenta en esta tesis, evalúa cuantitativamente la calidad de los procesos ágiles de desarrollo de software y ofrece apreciaciones cualitativas respecto al nivel de calidad obtenido.

El aporte significativo de AQF se centra en que el nivel de calidad de un proyecto ágil no sea definido únicamente a través de un valor numérico, sino que sea el resultado también de la evaluación de otros aspectos, en base al contexto en el que se desarrolla el proceso y los factores que lo impactan. Aquí, es necesario aclarar que, cuando se hace referencia al término “Proyecto Ágil”, se incluye a aquéllos que implementan prácticas ágiles para el proceso de desarrollo del producto software en base a cualquier enfoque tal como Scrum, Kanban, Lean o híbridos.

AQF está compuesto por:

- Un modelo de calidad, denominado QuAM (Quality Agile Model) que consta de:
 - Una estructura formal definida en función a componentes, atributos, métricas y criterios relacionados entre sí.
 - Una estrategia de medición que establece el nivel de calidad asociado a cada componente.
- Una herramienta de software, QuAGI (Quality AGIle) que da soporte al modelo, constituyendo un medio para su validación, y se compone de:
 - Una arquitectura que define cada uno de los componentes que forman la herramienta.
 - Una representación lógica que permite implementar el proceso de medición sobre cada uno de los componentes del modelo QuAM.
 - Un conjunto de interfaces que facilitan la interacción de actores externos (tales como administrador del proyecto, equipo de desarrollo, etc) con la herramienta.

En este capítulo se describe en la sección 3.1 el modelo QuAM explicando cada uno de sus componentes, y luego, en la sección 3.2, se presenta la herramienta QuAGI que da soporte a dicho modelo. Finalmente, en la sección 3.3 se exponen conclusiones generales al respecto.

También, se incluyen en esta tesis, anexos que amplían y respaldan el presente documento:

- **Anexo D:** En el que se describen las funcionalidades de QuAGI junto al diseño de sus interfaces
- **Anexo E:** En el que se describe cómo se logró la integración entre el modelo QuAM y la implementación de la herramienta QuAGI.

3.2 - QuAM: Modelo de Calidad para procesos ágiles de software

En esta sección se describe QuAM (Quality Agile Model), uno de los elementos principales de AQF, que es un modelo conceptual cuyo objetivo es describir el conjunto de componentes que influye en la calidad de proyectos ágiles junto a un procedimiento de evaluación de calidad que los integra. QuAM permite asociar los resultados de la medición con un perfil de calidad asociado al proyecto ágil evaluado [51].

El modelo QuAM surge, en primer lugar, de las conclusiones obtenidas luego de la revisión sistemática de la literatura existente, y cuyo proceso ha sido descrito en el capítulo anterior. Y, por otro lado, se basa en el análisis del contexto actual a partir del relevamiento de información realizado en la Industria del software en el nordeste argentino (NEA) del cual se han obtenido factores de relevancia que fueron incluidos al esquema de componentes de QuAM.

Para ilustrar la definición de QuAM se incluyen en esta sección: los resultados del relevamiento respecto al contexto actual que han servido de base para la definición de componentes del modelo. Luego se hace foco en el diseño de QuAM a partir del detalle de cada uno de sus componentes. Y, finalmente, se exponen conclusiones a las que se han arribado en esta sección.

3.2.1 Relevamiento de la situación actual

En el contexto de esta tesis se han llevado adelante dos estrategias que, combinadas, han colaborado en la construcción y configuración de componentes de la versión final de QuAM. Por un lado, el análisis de la información que se ha obtenido a partir de la revisión sistemática (presentada en el capítulo 2 de esta tesis) ha permitido obtener una fotografía actual respecto a trabajos relacionados y otras investigaciones sobre la evaluación de calidad en proyectos ágiles.

Y, por otro lado, se han planificado y ejecutado reuniones con empresas y áreas dedicadas al desarrollo de software localizadas en el NEA y que actualmente estén trabajando sobre proyectos ágiles. Este estudio ha permitido no solo obtener un diagnóstico situacional respecto a la calidad en procesos ágiles de software desde la óptica de la Industria, sino que, además, ha colaborado en la definición del esquema de QuAM.

El proceso de relevamiento fue definido en función a 3 fases: Preparación, Ejecución y Documentación, las cuales se describen a continuación.

Preparación. El instrumento que se ha utilizado para la captura de datos ha sido la entrevista. La misma se ha llevado a cabo con la participación de personas de diferentes organizaciones de software, lo que ha facilitado el registro de la conversación o reunión con cada uno de ellos. Para evitar problemas relacionados con cuestiones éticas, en los datos sin procesar no se incluyó información que permitiese identificar a los participantes de la entrevista. Se registraron los nombres de los entrevistados y sus direcciones de correo electrónico, sólo para poder contactarlos cuando fuera llevado a cabo el procesamiento de resultados de las entrevistas y en caso de que surgiera alguna duda, pero no fue posible vincular sus nombres

a sus respuestas, manteniendo de esta forma la confidencialidad de los datos provistos por cada uno.

Si bien, a los fines de este estudio, se han utilizado *entrevistas del tipo semi estructuradas*, es decir aquellas que representan conversaciones guiadas [52], éstas se han llevado a cabo con *profundidad* de modo tal que se realizaron preguntas de manera neutral, de seguimiento o complementarias a medida que el entrevistado realizaba referencia a su perspectiva en determinado tópico [53].

La entrevista se ha definido en torno a 15 preguntas, y ha sido estructurada en 3 secciones, tal como se muestra en la Tabla 3.1:

- **Sección A: Datos demográficos.** Representa el relevamiento correspondiente a datos propios de la institución entrevistada y del responsable a cargo de responder la entrevista.
- **Sección B: Proyectos de Software.** Identifica información propia al seguimiento de los proyectos de software, clasificando las preguntas en dos subsecciones:
 - B.1. Gestión de Proyectos: Relevamiento centrado en obtener datos respecto a prácticas relacionadas a la gestión de los proyectos de software (técnicas, herramientas, etc.).
 - B.2. Enfoque Ágil: Recopilación de prácticas ágiles utilizadas para la gestión de proyectos de software y el nivel de expertise o conocimiento que hay al respecto.
- **Sección C: Calidad del Proceso.** Registra información sobre actividades relacionadas a la gestión de calidad del proceso de desarrollo de software, grado de conocimiento al respecto y necesidad de contar con procesos de evaluación de calidad en las instituciones relevadas.

Tabla 3.1 Secciones de la entrevista para el diagnóstico actual

Sección	Sub sección	Cantidad de preguntas
A. Datos demográficos		5
B. Proyectos de Software	B.1. Gestión de Proyectos	3
	B.2. Enfoques ágiles	4
C. Calidad del Proceso		3

Fuente: de elaboración propia

Las encuestas se han realizado, a cada persona, de acuerdo con una plantilla definida, sin embargo, las preguntas fueron abiertas, es decir, que había suficiente espacio para que los entrevistados se expresaran con mayor detalle. También resulta importante subrayar que la mayoría de las preguntas de las entrevistas no contaban con opciones, por lo tanto, en esos

casos, los resultados de este estudio son respuestas espontáneas por parte de los entrevistados. El compendio de preguntas de la entrevista se muestra en el *Anexo B*.

Las 15 preguntas fueron seleccionadas a partir del refinamiento y adaptación de las preguntas de otros estudios anteriores [54][55]. Esta lista de preguntas fue complementada con las necesidades de información enfocadas en el presente estudio como, por ejemplo: Gestión de Proyectos, Percepción de la Calidad, etc.

Una vez establecida la estructura del instrumento, se procedió a la identificación de los perfiles que participarían como entrevistados. Entonces se determinó que, la *población*, objeto de estudio, debía estar representada por perfiles que tengan injerencia en la gestión de proyectos ágiles y, por ende, en la toma de decisiones referidas a éstos. Esta decisión se basa fundamentalmente en que la opinión y decisión de expertos juega un rol importante en este proceso, su valor se encuentra representado en la riqueza de información que entregan a la investigación quienes forman parte de proyectos ágiles y trabajan en la mejora continua de los mismos [56].

Se debe destacar que el relevamiento de datos se ha llevado a cabo sobre el área de influencia directa de la Universidad Tecnológica Facultad Regional Resistencia, pues la tesis se realiza en el marco del proyecto de investigación IAI4445TC “Evaluación de calidad de procesos ágiles de software” radicado en esta institución y, asimismo, serán necesarias experiencias de validación in situ a través de las cuales se vinculará la propuesta que aquí se describe con las organizaciones correspondientes. Por ende, la *población* estuvo constituida por alrededor de 200 personas entre los que se encontraron empresarios, jefes de área de Sistemas y responsables de proyectos de software que forman parte de empresas o áreas de software, localizados en la región Noreste de Argentina (NEA) constituida por las provincias de Formosa, Misiones, Chaco y Corrientes. Luego, para obtener el tamaño de la *muestra*, se aplicó, sobre la *población* objetivo, la *técnica de muestreo aleatorio simple*, la cual establece que la probabilidad de selección de un sujeto de estudio "x" es independiente de la probabilidad que tienen el resto de los sujetos que forman parte de la población [57]. Entonces, tal como se muestra en la Tabla 3.2, se realizó una evaluación proporcional acorde a la distribución por provincia en la población, obteniéndose entonces una *muestra* de 60 personas, identificando el número de elementos de acuerdo con la proporción asignada.

Tabla 3.2. Población para entrevistas

Provincia	Número de Personas	Proporción	Tamaño de Muestra
Chaco	82	62%	51
Corrientes	33	25%	8
Formosa	5	4%	0
Misiones	12	8%	1
Total	130	100%	60

Fuente: de elaboración propia

Ejecución. El proceso de entrevistas se ha realizado durante 2 meses, a través de reuniones presenciales o a distancia, usando sistemas de videoconferencias (Hangout o Skype).

Cada uno de los encuentros se estructuraron en 2 fases:

- *Inicio:* Al iniciar cada reunión se han registrado datos respecto al tamaño de la organización, rol del entrevistado, tipos de proyectos en los que trabajan en la organización y experiencia en prácticas de desarrollo de software ágil. Para más información en el *Anexo C* se incluyen los resultados obtenidos a partir de los datos demográficos relevados en las encuestas.
- *Diagnóstico:* Para la segunda etapa de cada reunión, se ha preparado un compendio de preguntas, con el objetivo de obtener información útil para recopilar la experiencia del grupo de encuestados. Además, se ha logrado diagnóstico actual del sector respecto a prácticas ágiles de desarrollo de software y su relación con cuestiones de calidad. Las respuestas obtenidas en la fase de Diagnóstico se han analizado y organizado a fin de documentar el proceso y poder obtener resultados que colaboren en la definición de QuAM. Se exponen estos resultados a continuación.

Documentación. A partir de la ejecución de las reuniones, todas las respuestas que se han logrado en la Sección *Proyectos de Software* y la Sección *Calidad de Software* de las encuestas, se han organizado en una planilla de Excel, a fin de tabular resultados y realizar un análisis gráfico de los mismos que permita exponer conclusiones al respecto.

En la Figura 3.1 se muestran los resultados respecto al relevamiento de gestión de los proyectos y la utilización de software como soporte técnico a esta actividad. Así se destaca que casi el 15% de los entrevistados tienen experiencia trabajando con equipos distribuidos, y que aproximadamente el 80% utiliza alguna herramienta de gestión de proyectos (entre las más utilizadas se encuentran Trello y Jira). Lo más relevante de la evaluación de este aspecto, es que más del 50% expresó que estaría dispuesto cambiar de herramienta de gestión de proyectos, siempre y cuando observen beneficios y le aporte valor a la producción.

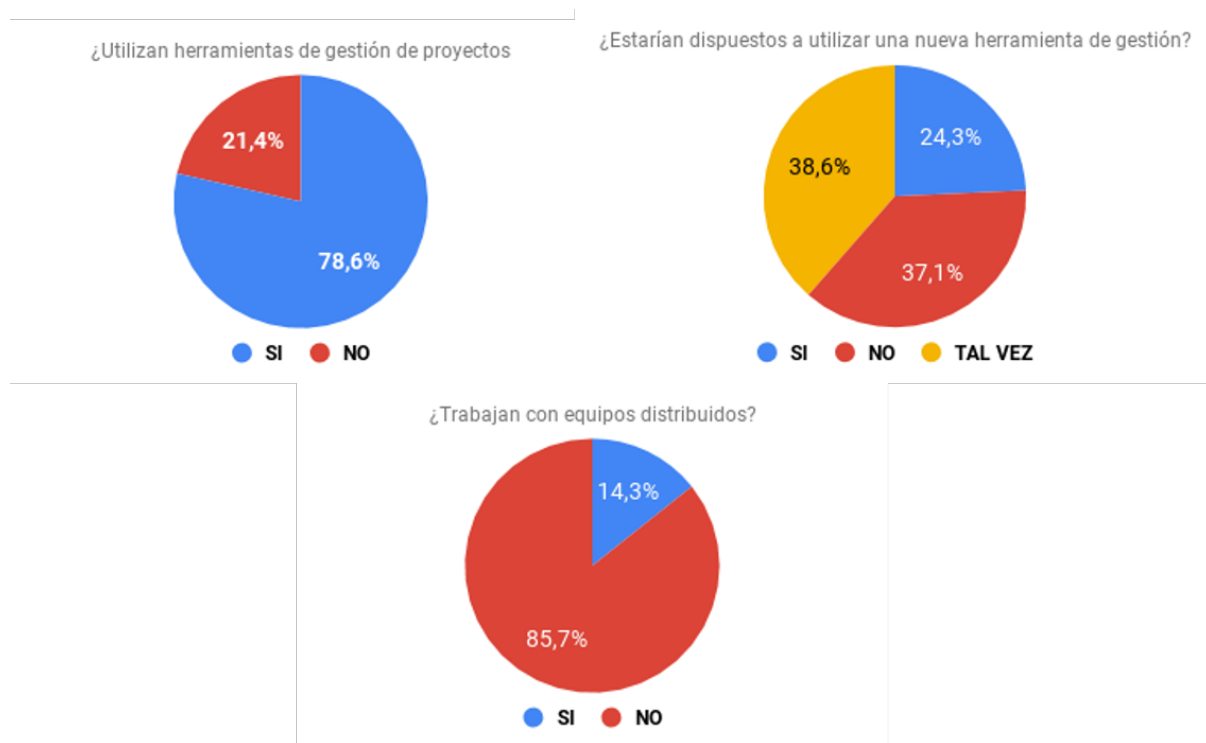


Figura 3.1 Resultado de entrevistas “Gestión de Proyectos”
Fuente: De elaboración propia

Asimismo, a partir de estas entrevistas, se obtuvo el porcentaje de proyectos ágiles sobre el conjunto de proyectos llevados a cabo por la organización, este número sirvió para ilustrar la relevancia de la gestión de proyectos ágiles en la población estudiada. Tal como se observa en la Figura 3.2, los entrevistados podían elegir entre las siguientes opciones: 0%, 10%, 25%, 50%, 75%, 100% teniendo en cuenta el porcentaje que representan los proyectos ágiles en sus organizaciones. En este sentido, se destaca, por un lado, que todos los entrevistados expresan trabajar con al menos 1 (un) proyecto ágil y, por otro lado, aproximadamente el 80% indicó que más de la mitad de sus proyectos son ágiles.

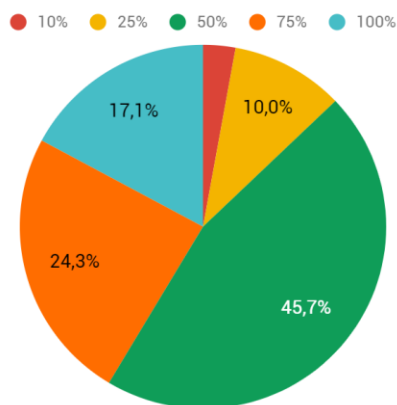


Figura 3.2 Resultados de entrevistas “Porcentaje de Proyectos ágiles”
Fuente: De elaboración propia

Para identificar las prácticas ágiles más utilizadas en los proyectos de software de las organizaciones encuestadas, se han realizado algunas preguntas al respecto.

En primer lugar, respecto al grado de utilización de enfoques ágiles, como se observa en la Figura 3.3, aproximadamente el 80% de las organizaciones utiliza alguno, siendo SCRUM la opción más elegida.

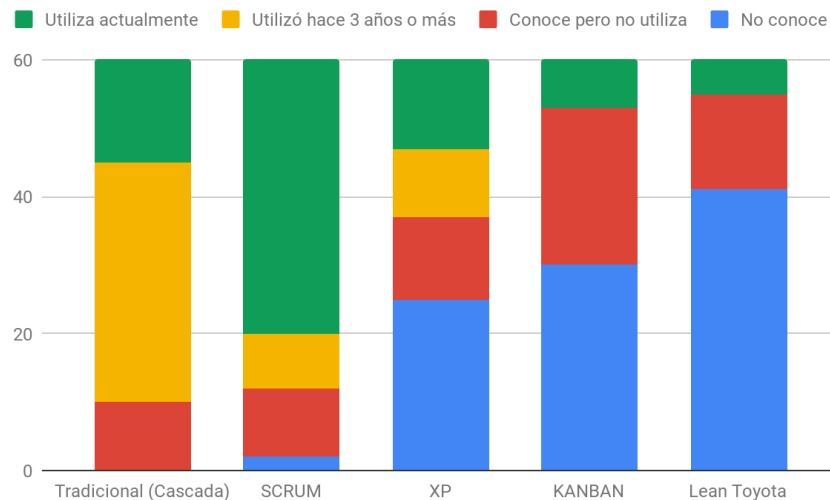


Figura 3.3 Resultados de entrevistas “Enfoques ágiles utilizados”
Fuente: De elaboración propia

En segundo lugar, en base al reporte publicado por VersionOne [22], se elaboró una lista de prácticas ágiles para que los entrevistados seleccionen las utilizadas en sus proyectos de software. En algunos casos se debió explicar al entrevistado el concepto para identificar la práctica, de hecho, se pudo notar que si bien tenían experiencia o utilizaban las prácticas no podían relacionarlo a su definición técnica.

Relacionado a ello, se observa en la Tabla 3.3 que las prácticas ágiles más utilizadas son las técnicas propuestas en SCRUM, como las iteraciones cortas, daily meetings, backlogs priorizados, revisión de iteraciones y retrospectivas. Aparecen, entre las más utilizadas, otras prácticas relacionadas tales como tablero Kanban e integración continua que forman parte de metodologías tales como Lean Kanban, XP, etc.

Tabla 3.3 Resultados de entrevistas - “Prácticas ágiles”

Prácticas ágiles	SI la utiliza	NO la utiliza	No la conoce
Sprint Iteration	50	6	4
Daily Meeting	40	10	10
Kanban	34	16	10
Retrospectiva	32	23	5
Backlogs priorizados	30	10	20
Sprint Review	15	40	5
Release Planning	12	25	23
Definition of Done	11	14	35
Integración Continua	5	33	22
Calendario niko-niko	0	15	45
Burndown charts	0	18	42

Fuente: de elaboración propia

El relevamiento ha permitido, también, obtener diagnóstico sobre aspectos relacionados a la calidad en el proceso de desarrollo de software que sirvieron de base para la definición de elementos del modelo. Tal como se muestra en la Figura 3.4, si bien aproximadamente el 75% conoce entre poco y nada sobre Calidad de Software, cerca del 90% considera que tener en cuenta factores que impactan sobre la calidad, mejoraría no solo el proceso de desarrollo de software, sino también el resultado de este.

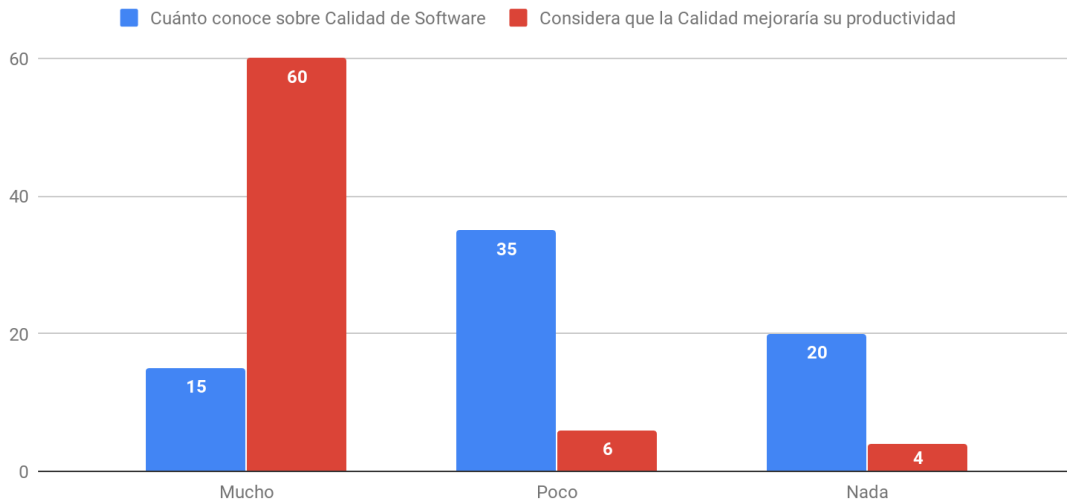


Figura 3.4. Resultados de entrevistas “Calidad de Software”
Fuente: De elaboración propia

Además, durante cada reunión se registraron problemas que los entrevistados destacaban como factores que afectaban, positiva o negativamente, a la calidad de sus procesos. En la Tabla 3.4 se listan en forma general los 7 problemas identificados y con mayor frecuencia de aparición entre los comentarios de los entrevistados. Es importante detenerse aquí para remarcar que, no necesariamente, los entrevistados han hecho referencia al término tal como se incluye en el listado de la Tabla 3.4, sin embargo, al analizar los resultados de cada una de las reuniones se logró unificar características comunes en un único término. Por ejemplo: comentarios tales como “El mayor problema lo tenemos al momento de definir la necesidad”, “Nos focalizamos primero en que el Cliente nos ayude a definir las historias de usuario”, “Es imposible evaluar la calidad del proceso si hay una mala definición de requerimientos”, entre otros, se nuclearon en el factor identificado como “Definición de requerimientos”.

Tabla 3.4 Resultados de entrevistas “Factores que impactan sobre la calidad de software”

Factor identificado	Cantidad de entrevistados que lo consideran
Definición de requerimientos	55
Participación del Cliente	54
Testing de Aceptación	38
Organización de equipos	33
Estimación	31
Priorización de requerimientos	25
Dependencia de requerimientos	25

Fuente: de elaboración propia

El resultado de esta experiencia de colaboración con organizaciones sirvió de base para que, junto a la información relevada en la literatura disponible, se definan los componentes del modelo QuAM para que éste respóndase adecúe, por un lado, a los marcos conceptuales que hoy en día estudian a la calidad del software y por otro lado se focalice en las particularidades de proyectos ágiles.

3.2.2 Diseño de QuAM

En [58] se plantea un modelo conceptual, denominado WebQM, en el cual se presenta la definición, documentación y explotación de métricas de calidad desde una perspectiva común y aplicable a diversos dominios. Esta propuesta sirve de punto de partida al presente trabajo respecto a la definición de atributos, métricas y criterios utilizando, además, las plantillas que WebQEM propone como guía para describir cada métrica directa e indirecta que integra el modelo.

De esta forma, el modelo QuAM, que se propone en esta tesis y que está basado en la estructura de WebQEM, complementa su definición incorporando nuevos conceptos enfocados en la evaluación de calidad de prácticas asociadas a los enfoques ágiles de desarrollo de software.

3.2.2.1 Estructura del modelo QuAM

Para representar gráficamente la estructura de QuAM, se presenta en la Figura 3.5 el diagrama de clases correspondiente, incluyendo elementos y relaciones que caracterizan al modelo propuesto.

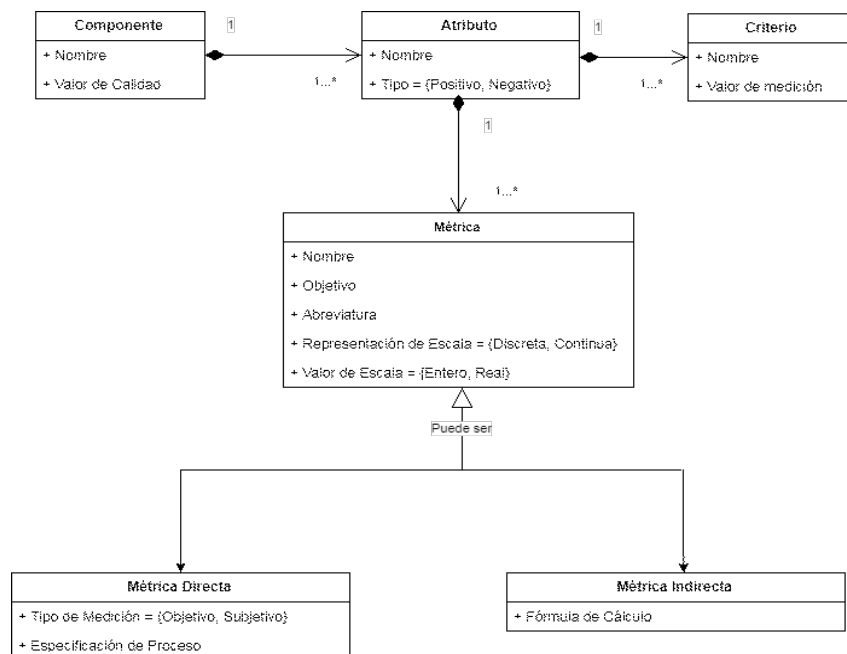


Figura 3.5 Diagrama de Clases QuAM
Fuente: de elaboración propia

Clase Componente. De acuerdo con lo especificado por Olsina et. al. en [59], una *entidad* representa un objeto, tangible o intangible, que exhibe un comportamiento observable en el mundo real. Por ejemplo, una entidad es un proceso.

En esta tesis el término equivalente a entidad es *componente*, pues, a diferencia de lo que se propone en [59], en QuAM se consideran cada uno de los aspectos que impactan en la calidad del proceso de desarrollo de software y que *componen* el proyecto ágil a evaluar. De esta forma, un ejemplo de componente es la Gestión de Requerimientos y Requisitos (que se describe más adelante en esta tesis), este aspecto forma parte de todo proyecto ágil y su análisis permitirá, en este caso, evaluar la calidad del proceso de software desde esa óptica.

De acuerdo a lo que se observa en la Figura 3.5, los atributos de la clase Componente son:

- *Nombre*: Identifica al componente en el modelo
- *Valor de Calidad*: Representa un valor numérico que define el resultado de la evaluación de Calidad. Más adelante se muestra cómo se utiliza este valor para informar al usuario sobre el perfil de calidad asociado.

Clase Atributo. Para evaluar la calidad de las entidades, una práctica común es, realizarlo a través de la cuantificación de *atributos* [58]. En el caso de QuAM, los *atributos* representan las propiedades de los componentes que son de interés para fines de medición. Por ejemplo, para el componente de Gestión de Requerimientos y Requisitos se considerará a la Completitud en la definición de cada uno como atributo para su evaluación.

En el diagrama de clases representado en la Figura 3.5 se muestra que los elementos de la clase Atributo son:

- *Nombre*: Identifica al atributo en el modelo
- *Tipo*: El conjunto de atributos de cada componente puede clasificarse como uno de dos tipos posibles: *Atributos Positivos*, aquéllos que se desean enfatizar y promover para aumentar los niveles de calidad, y *Atributos Negativos*, aquéllos que se deberán minimizar para que no afecten disminuyendo los niveles de calidad del proceso.

Clase Métrica. Un atributo puede ser cuantificado por una o varias *métricas*, dependiendo de cómo se realice la medición. Lo anterior significa que una *métrica* debe ser comprendida en relación con los *atributos* que está cuantificando, y que, a su vez, se relacionan a un determinado *componente*. Esto implica, además, identificar claramente el tipo de valor que se obtiene, en qué unidad se expresa ese valor y el tipo de escala que se usa, para poder realizar posteriormente un correcto análisis matemático y de interpretación [60].

Tal como se expone en [61], Una *métrica* contiene la información del método y la escala de medición (y / o cálculo) definidos. Un *atributo* puede medirse utilizando diferentes métodos de medición y escalas, por lo tanto, una o más *métricas* pueden cuantificar el mismo atributo. La *métrica* contiene la información del método y la escala de medición (y / o cálculo) definidos. Un *atributo* puede medirse utilizando diferentes métodos de medición y escalas, por lo tanto, una o más *métricas* pueden cuantificar el mismo atributo.

En la definición de QuAM, la evaluación de cada componente se especifica a través del proceso de medición de las *métricas* asociadas a los atributos que lo componen, cuyo resultado ofrece evidencia empírica para poder comprender, de mejor manera, las diferentes dimensiones que engloban la calidad del proyecto ágil que se analice.

Asimismo, las métricas pueden ser *métricas directas* o *métricas indirectas*, las directas son aquellas que no dependen de la métrica de ningún otro atributo, mientras que las indirectas se derivan a partir de métricas de otros atributos [62][63]. Por ejemplo, en el caso del atributo “Complejidad de Requerimientos”, una métrica directa será el “Grado de inclusión de criterios de aceptación en Historias de Usuario” y una métrica indirecta será el “Porcentaje de inclusión de criterios de aceptación en Historias de Usuario” (pues intervienen en la medición más de una métrica)¹.

Teniendo en cuenta lo representado en la Figura 3.5, la Clase Métrica y sus clases heredadas se describen en función a los siguientes atributos:

- *Nombre*: Identificación de la métrica en el modelo
- *Objetivo*: Define el fin de la métrica
- *Abreviatura*: Nombre utilizado en las fórmulas
- *Representación de Escala*: La escala representa el conjunto de valores que puede tomar el resultado de la medición o cálculo de la métrica. Su representación puede ser discreta o continua, de acuerdo con su valor.
- *Valor de la Escala*: Tipo de dato entero o real que se obtiene como resultado de medición.
- Si es Métrica Directa, la clase se compone de los siguientes atributos:
 - *Especificación de Proceso*: Representa a la secuencia lógica particular de las operaciones y las posibles heurísticas especificadas para realizar la medición y obtener un valor.
 - *Tipo*: Indica si la medición es *subjetiva*, es decir, involucra el juicio humano o si es *objetiva*, es decir se basa en reglas matemáticas.
- Si es Métrica Indirecta, la clase se compone de los siguientes atributos:
 - *Fórmula de Cálculo*: Conjunto de operaciones matemáticas que permiten relacionar los valores obtenidos en una o más métricas directas.

Clase Criterio. Los posibles resultados que se obtengan en la medición de los atributos se interpretarán asociándose a determinados niveles o umbrales significativos representados por *criterios* en QuAM. Por ejemplo, si al ejecutarse la medición del atributo “Complejidad de los requerimientos” el valor obtenido es 4, el criterio asociado a esto indica que “Todas las historias de usuario se encuentran redactadas de manera completa”.

¹ Se describe cada métrica del modelo en la sección de detalle de los componentes en este trabajo

La clase Criterio se compone de los siguientes atributos, tal como se observa en la Figura 3.5:

- *Nombre*: Identifica al criterio en el modelo
- *Valor de medición*: Es un valor numérico que asocia el nivel o umbral al que se corresponde el criterio.

3.2.2.2 Diseño de componentes de QuAM

Los componentes de QuAM se han diseñado teniendo en cuenta:

- los *cuatro valores expresados en el Manifiesto Ágil*, pues resulta importante evaluar la calidad desde los aspectos que representan la filosofía de cualquier enfoque ágil, independientemente de cuál sea éste [64].
- las *conclusiones respecto a la revisión sistemática* que se llevó adelante previo al diseño del modelo y que constituyó un marco de referencia fundamental y
- finalmente, QuAM ha sido influenciado por las *características identificadas durante el relevamiento que se realizó sobre organizaciones* que actualmente trabajan en proyectos ágiles en la zona del NEA.

Ambas versiones del modelo, la que aquí se presenta y su predecesora [51], han sido validadas a través del estudio de casos en la industria, por lo que su evolución se fue dando paulatinamente como respuesta a cada una de estas experiencias. El esquema actual, graficado en la Figura 3.6, representa una instancia del modelo QuAM y su definición completa se describe a continuación.



Figura 3.6 Esquema de Componentes de QuAM

Fuente: De elaboración propia

Componente N° 1 “Capacidad de Producción de Entregables”. Este componente define la evaluación de factores que influyen en la obtención de entregables de valor para el cliente. Pues, un entregable que pueda ser instalado y utilizado por el Product Owner da visibilidad al avance del proyecto, haciendo que el trabajo y el avance del proyecto sean tangibles [65]. Originalmente, este componente consideraba cuestiones técnicas, tales como el uso de herramientas de gestión de cambios y testing automatizado, como factores que favorecen la producción de entregables. Sin embargo, y luego de la validación inicial expuesta en [51], se modificó su estructura en base a la observación de resultados obtenidos y propuesta de diversos equipos de proyectos ágiles evaluados.

El Componente N° 1 “Capacidad de Producción de Entregables” se evalúa en base a la medición de dos atributos:

- *Atributo Positivo: Valor a la estimación*
AQF considera positivo implementar estrategias que mejoren la calidad de la estimación en los proyectos ágiles, pues esto minimizará desfases en la obtención de entregables. Las acciones tendientes a lograr más precisión en la estimación de la entrega de artefactos en cada iteración, en detrimento de una visibilidad a largo plazo, aumentarán el nivel de calidad de la Producción de Entregables en proyectos ágiles.

Las métricas que componen el atributo positivo “Valor a la estimación” se describen a continuación en las Tablas 3.5, 3.6 Y 3.7

Tabla 3.5 Descripción de Métrica N° 1.1

Métrica N° 1.1	
Tipo	Directa
Nombre	Total de historias de Usuario
Abreviatura	#US
Objetivo	Obtener información de cantidad de historias de usuario del proyecto
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Se cuentan las historias de usuario revisando el product backlog y los sprints backlog
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.6 Descripción de Métrica N° 1.2

Métrica N° 1.2	
Tipo	Directa
Nombre	Historias de usuario estimadas
Abreviatura	#USE
Objetivo	Obtener el total de historias de usuario que han sido estimadas en función a la asignación de puntos de historia
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Se analizan todas las historias de usuario, si tiene asignado puntos de historia se cuenta 1, sino se pasa a la siguiente. Al finalizar se contabiliza el total
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.7 Descripción de Métrica N° 1.3

Métrica N° 1.3	
Tipo	Indirecta
Nombre	Porcentaje de historias de usuario con puntos de historia asignados
Abreviatura	%USE
Objetivo	Obtener el porcentaje de historias de usuario que han sido estimadas en función a la asignación de puntos de historia
Fórmula de cálculo	$\%USE = \frac{\#USE \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

- *Atributo Negativo: Valor al lead time*

El lead time representa el tiempo que se tarda en terminar cada tarea, midiéndolo en base a lo percibido por el cliente [66]. Si bien evaluar este factor permitirá lograr mejoras a nivel de procesos de negocio, para el proceso ágil de desarrollo de software resultará negativo que el proyecto se enfoque en reducir matemáticamente el lead time en lugar de desarrollar prácticas que minimicen el tiempo transcurrido entre la consulta del cliente y la respuesta a la necesidad del cliente (cycle time).

Por ello en este atributo se evalúa el nivel de participación del Product Owner en la validación de los entregables obtenidos, tendiente a valorar el cycle time por sobre el lead time.

Las métricas que componen al atributo negativo “Valorar el Lead time” se describen en las Tablas 3.8, 3.9 y 3.10.

Tabla 3.8 Descripción de Métrica N° 1.4

Métrica N° 1.4	
Tipo	Directa
Nombre	Cantidad de Criterios de Aceptación
Abreviatura	#AC
Objetivo	Obtener el total de criterios de aceptación que se han definido por cada historia de usuario.
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Se analizan todas las historias de usuario, se cuentan cantidad de criterios de aceptación definidos y se acumulan.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.9 Descripción de Métrica N° 1.5

Métrica N° 1.5	
Tipo	Directa
Nombre	Cantidad de Criterios de Aceptación validados
Abreviatura	#ACSi
Objetivo	Proporcionar el número de criterios de aceptación que fueron definidos y validados como “Cumplidos” por el Product Owner al realizar testing de aceptación de una historia de usuario
Procedimiento de Medición	
Tipo	Objetivo
Especificación	En cada historia de usuario finalizada se verifica el estado de sus criterios de aceptación, se cuenta 1 si han sido validados por el Product Owner, 0 en caso contrario.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.10 Descripción de Métrica N° 1.6

Métrica N° 1.6	
Tipo	Indirecta
Nombre	Porcentaje de historias de usuario validadas
Abreviatura	%HUV
Objetivo	Obtener el porcentaje de historias de usuario que han sido validadas por el Product Owner con el 100% de los criterios de aceptación cumplidos.
Fórmula de cálculo	$\%HUT = \frac{\#ACSi}{\#AC} \times 100$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Para terminar de describir el Componente N° 1 “Capacidad de Producción de Entregables” a continuación se exponen en la Tabla 3.11 la relación entre criterios y los atributos, a partir de la cual se calcula el valor de calidad correspondiente al componente.

Tabla 3.11 Relación entre atributos y criterios del Componente N° 1

Componente 1: Capacidad de Producción de Entregables			
Atributo Positivo		Atributo Negativo	
A1.1 Valor a la estimación		A1.2 Valor al lead time	
<i>Criterios de medición para A1.1</i>	<i>Valor</i>	<i>Criterios de medición para A1.2</i>	<i>Valor</i>
1.1.1 Todas las historias de usuario de un sprint tienen asignado puntos historia	4	1.2.1 De las historias de usuario finalizadas se encuentra validadas menos del 10%	-4
1.1.2 Más del 80% de las historias de usuario tienen asignado puntos de historia.	3	1.2.2. De las historias de usuario finalizadas se encuentra validadas menos del 40% y más del 10%.	-3
1.1.3 Entre el 40 % y el 80% de las historias de usuario tienen asignado puntos de historias.	2	1.2.3 De las historias de usuario finalizadas se encuentra validadas menos del 80% y más de 40% -	-2
1.1.4 Entre el 10% y el 40% de las historias de usuario tienen asignado puntos de historias.	1	1.2.4 De las historias de usuario finalizadas se encuentra validadas más del 80%.	-1
1.1.5 Menos del 10% de las historias de usuario tienen asignado puntos de historia	0	1.2.5 De las historias de usuario finalizadas se encuentra validadas todas	0

Fuente: de elaboración propia

Componente N° 2 “Evaluación del Equipo de Trabajo”. Los procesos de desarrollo de software utilizando prácticas ágiles dependen del funcionamiento de sus equipos de trabajo en forma flexible y con poder de decisión [67], deben estar integrados multi funcionalmente, colaborando en ofrecer conocimiento y enriquecer al cliente. Son dinámicos y rápidamente reconfigurables, dando flexibilidad a la organización; son cooperativos, en y entre las compañías, permitiendo la cooperación intra y extra organizacional necesaria para aumentar la competitividad; y son virtuales, lo cual permite a la compañía combinar los recursos necesarios para cumplir los objetivos empresariales [68]. Tal como lo expresa Beck en los principios del Manifiesto de desarrollo ágil de software [69], un equipo de trabajo ágil puede mejorar la velocidad y la productividad logrando obtener las mejores arquitecturas, requisitos y diseño de software.

En base a ello, QuAM incluye entre sus componentes a la evaluación de la calidad del proceso de desarrollo ágil teniendo en cuenta el flujo de comunicación entre los miembros del equipo a partir de la división de tareas y la transparencia del trabajo realizado en base a la definición de roles. De hecho, tal como se mostró con anterioridad al estudiar las organizaciones, muchas han planteado necesario evaluar aspectos de calidad que contribuyan a fortalecer un equipo ágil y multifuncional en el proyecto desde el punto de vista de la organización del trabajo del mismo equipo, desde la definición de roles hasta la estructuración de cada sprint.

A tal fin se diseñó el Componente N° 2 “Evaluación del equipo de trabajo” el cual considera como impacto positivo en la calidad a la definición clara de roles, y de impacto negativo, a

la falta de organización del proyecto restándole importancia a la definición de sprints. Este componente, entonces, se evalúa en base a la medición de dos atributos:

- *Atributo Positivo: Valor a la definición de roles y tareas*

En primer lugar, forman parte del atributo positivo los criterios que permiten evaluar la manera en la que se definen y se asignan las tareas, buscando promover que los proyectos ágiles definan roles para cada uno de los integrantes del equipo y valoren la transparencia del proceso, de forma tal de implementar estrategias que permitan que todos estén informados de las tareas que realiza cada uno. Para lograrlo QuAM establece que se evaluará, por cada historia de usuario, que las tareas definidas tengan asignado un responsable dentro del equipo y que además la tarea esté clasificada (por ejemplo, frontend, backend, testing, etc.), de modo tal de ofrecer información al equipo sobre quién está realizando qué tipo de tarea en todo momento. Entonces, por ejemplo, si todas las tareas del proyecto han sido completamente definidas, es decir tienen asignado un responsable y han sido tipificadas, el atributo positivo tomará el máximo valor posible. En el caso que ninguna de las tareas haya sido completamente definida, entonces QuAM establece que esta situación representa el valor más bajo de la escala y asigna el peor nivel de calidad para el atributo positivo.

En este sentido, las métricas que componen el atributo positivo “Valor a la definición de roles y tareas” de este componente se describen a continuación en las Tablas 3.12, 3.13, 3.14, 3.15 y 3.16

Tabla 3.12 Descripción de Métrica N° 2.1

Métrica N° 2.1	
Tipo	Directa
Nombre	Cantidad de Tareas del Proyecto
Abreviatura	#Tar
Objetivo	Obtener el total de tareas definidas por historia de usuario en el proyecto ágil a evaluar.
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Por cada historia de usuario se cuentan las tareas que la componen. Estas cantidades se suman al finalizar el análisis de todas las historias de usuario del proyecto.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.13 Descripción de Métrica N° 2.2

Métrica N° 2.2	
Tipo	Directa
Nombre	Cantidad de Tareas del Proyecto con Responsable asignado
Abreviatura	#TR
Objetivo	Obtener el total de tareas por historia de usuario que tengan responsable asignado.
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Por cada tarea se analiza si tiene asignado responsable. Si es así se cuenta 1, sino 0. Al finalizar se obtiene el total de tareas del proyecto.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.14 Descripción de Métrica N° 2.3

Métrica N° 2.3	
Tipo	Directa
Nombre	Cantidad de Tareas del Proyecto tipificadas
Abreviatura	#TT
Objetivo	Obtener el total de tareas definidas por historia de usuario que hayan sido clasificadas en UI (User Interface), Front (Front End) , Back (Back End) o Testing.
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Por cada una de las tareas se analiza: Si fue tipificada, se cuenta 1; si el tipo de tarea ha sido clasificado como "Otro" entonces se suma 0. Al finalizar se obtiene el total de tareas del proyecto.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.15 Descripción de Métrica N° 2.4

Métrica N° 2.4	
Tipo	Indirecta
Nombre	Porcentaje de tareas que tienen responsable asignado
Abreviatura	%TR
Objetivo	Obtener el porcentaje de tareas que tengan un responsable asignado
Fórmula de cálculo	$\%TR = \frac{\#TR \times 100}{\#Tar}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Tabla 3.16 Descripción de Métrica N° 2.5

Métrica N° 2.5	
Tipo	Indirecta
Nombre	Porcentaje de tareas que han sido tipificadas
Abreviatura	%TT
Objetivo	Obtener el porcentaje de tareas que han sido tipificadas teniendo en cuenta el total de tareas en el proyecto
Fórmula de cálculo	$\%TT = \frac{\#TT \times 100}{\#Tar}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

- *Atributo Negativo: Valor al proceso por sobre el equipo*

La evaluación del Componente #2 considera como factor de impacto negativo definir historias de usuario “islas solitarias”¹, es decir que no pueden incluirse en ningún sprint ni se componen de tareas que permitan mejorar su significado. Este atributo surge a partir de la experiencia empírica en ambientes reales donde fue posible observar que, al restringir la representación de la necesidad del cliente a la mera definición de la historia de usuario, sin descomposición en tareas, ni agrupación en algún sprint, se otorga más valor al proceso de elicitación que a la organización del equipo que en definitiva trabajará sobre la necesidad. En estos casos, la priorización en el proyecto se torna difícil, debido a la escasa información con la que se cuenta, y a su vez se quita valor al equipo, pues pierde el foco de trabajo y se centra en tratar de descifrar el problema. Así, la calidad del proyecto sufre un impacto negativo el seguimiento del proceso no podrá realizarse correctamente sin tener la información clara ni del problema ni del equipo.

En función a lo antes expuesto, se describen a continuación, en las Tablas 3.17, 3.18, 3.19 y 3.20 las métricas que permitirán obtener el valor asociado a este atributo:

Tabla 3.17 Descripción de Métrica N° 2.6

Métrica N° 2.6	
Tipo	Directa
Nombre	Historias de usuario que forman parte de algún sprint
Abreviatura	#USSp
Objetivo	Determinar la cantidad de historias de usuario que forman parte de algún sprint
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Se analiza la historia de usuario. Si forma parte de algún sprint se cuenta 1, en caso contrario se cuenta 0. Y al finalizar el análisis de las historias de usuario se obtiene el total de historias que forman parte de algún sprint.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

¹ El término “islas solitarias” fue acuñado por el equipo que colaboró en el desarrollo del software que forma parte del framework AQF y hace referencia a historias de usuario que no pueden agruparse en ningún sprint y para las que además no se han definido tareas.

Tabla 3.18 Descripción de Métrica N° 2.7

Métrica N° 2.7	
Tipo	Directa
Nombre	Historias de usuario que se componen de tareas
Abreviatura	#USTar
Objetivo	Obtener qué cantidad de historias de usuario han sido descompuestas en tareas
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Si la historia de usuario tiene al menos definida una tarea se suma 1, sino se cuenta 0. Al finalizar se obtiene el total de historias que tienen definida al menos una tarea.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.19 Descripción de Métrica N° 2.8

Métrica N° 2.8	
Tipo	Indirecta
Nombre	Porcentaje de historias de usuario que pertenecen a algún sprint
Abreviatura	%USSp
Objetivo	Obtener el porcentaje de historias de usuario del proyecto que han sido asignadas a algún sprint.
Fórmula de cálculo	$\%USSp = \frac{\#USSp \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Tabla 3.20 Descripción de Métrica N° 2.9

Métrica N° 2.9	
Tipo	Indirecta
Nombre	Porcentaje de historias de usuario que tienen tareas definidas
Abreviatura	%USTar
Objetivo	Obtener el porcentaje de historias de usuario del proyecto compuestas por tareas.
Fórmula de cálculo	$\%USTar = \frac{\#USTar \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Finalmente, en la Tabla 3.21 se muestra cómo el Componente N° 2 “Evaluación de Equipo de Trabajo” ha sido conformado teniendo en cuenta atributos, criterios y valores asociados.

Tabla 3.21 Relación entre atributos y criterios del Componente N° 2

Componente 2: Evaluación del Equipo de Trabajo			
Atributo Positivo		Atributo Negativo	
A2.1 Valor a la definición de roles		A2.2 Valor al proceso por sobre el equipo	
<i>Criterios de medición para A2.1</i>	<i>Valor</i>	<i>Criterios de medición para A2.2</i>	<i>Valor</i>
2.1.1 Al asignar tareas se indica el tipo y responsable, distinguiendo rol de los miembros del equipo	4	2.2.1 Se define las historias de usuario sin agruparlas en sprints ni especificar tareas	-4
2.1.2 Al asignar tareas se indica solo el responsable pero no el tipo	2	2.2.2. Se define las historias de usuario, agrupándolas en sprints pero no se especifican tareas	-2
2.1.3 No se indica responsable de la tarea	0	2.2.3 Se define las historias de usuario, agrupándolas en sprints y especificando tareas	0

Fuente: de elaboración propia

Componente N° 3 “Comunicación con el Cliente”. Los proyectos de software no son estáticos, todo lo contrario, deben adaptarse constantemente al dinamismo del entorno y las tecnologías, pero fundamentalmente, a los cambios y nuevos requerimientos que el cliente propone [70]. Por ello, surgen los enfoques ágiles como una alternativa que permita a los equipos de software responder a los cambios, acortando el tiempo de desarrollo y mejorando la comunicación y la colaboración con el cliente, sobre todo cuando el tiempo de entrega es una ventaja competitiva para una organización [71].

QuAM propone evaluar cómo se realiza la comunicación entre el cliente y el equipo, propiciando su participación en el proyecto. En el análisis inicial, presentado en este capítulo, los equipos han coincidido que, cuanto mayor es el grado de colaboración del Product Owner, mejor será la calidad del proceso, pues el sesgo de error en la relación “*Expectativa versus Realidad*” disminuye notoriamente y mejora la identificación temprana de errores. Ocurre lo contrario cuando solo se considera al Product Owner desde un punto de vista contractual, pues se pierden las ventajas del contacto y la comunicación constante. Así, el Componente N° 3 “Comunicación con el cliente” se define en función a dos atributos:

- *Atributo Positivo: Valor a la colaboración con el Cliente*

QuAM considera positivo la colaboración del Product Owner en la definición de historias de usuario y su participación en los sprints planning. Cabe destacar que, este rol ha sido tenido en cuenta, también, en la evaluación de otros componentes del modelo. Se describen a continuación las métricas que componen el atributo positivo “Valor a la colaboración con el cliente” del Componente N° 3 en las Tablas 3.22, 3.23, 3.24, 3.25 y 3.26

Tabla 3.22 Descripción de Métrica N° 3.1

Métrica N° 3.1	
Tipo	Directa
Nombre	Cantidad de sprints planning
Abreviatura	#SPI
Objetivo	Obtener el total de sprints que han sido categorizados como Sprint Planning
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Por cada sprint se verifica si es Sprint Planning y se acumula 1, sino no se acumula.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.23 Descripción de Métrica N° 3.2

Métrica N° 3.2	
Tipo	Directa
Nombre	Participación del Product Owner en el Sprint Planning
Abreviatura	#POSP
Objetivo	Obtener el total de sprints planning en los cuales el Product Owner ha participado
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Por cada sprint se verifica si el Product Owner ha participado y se cuenta 1, sino 0. Al finalizar se obtiene el total de sprints planning en los cuales ha participado el Product Owner
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.24 Descripción de Métrica N° 3.3

Métrica N° 3.3	
Tipo	Indirecta
Nombre	Grado de participación del Product Owner en Sprints Planning
Abreviatura	%POSP
Objetivo	Obtener el porcentaje de sprints planning en los cuales ha contado la participación del Product Owner sobre el total de sprints planning del proyecto
Fórmula de cálculo	$\%POSP = \frac{\#POSP \times 100}{\#SPL}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Tabla 3.25 Descripción de Métrica N° 3.4

Métrica N° 3.4	
Tipo	Directa
Nombre	Historias de Usuario definidas por el Product Owner
Abreviatura	#USPO
Objetivo	Determinar cantidad de historias de usuario que ha definido el Product Owner
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Por cada una de las historias de usuario del proyecto se verifica si el Product Owner ha participado en su definición. Si es así se cuenta 1, sino no se cuenta. Al finalizar, se obtendrá la cantidad de historias de usuario que hayan sido definidas por el Product Owner.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.26 Descripción de Métrica N° 3.5

Métrica N° 3.5	
Tipo	Indirecta
Nombre	Porcentaje de historias de usuario definidas por el Product Owner
Abreviatura	%USPO
Objetivo	Obtener el porcentaje de historias de usuario que han sido definidas por el Product Owner sobre el total de historias de usuario del proyecto
Fórmula de cálculo	$\%USPO = \frac{\#USPO \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

- *Atributo Negativo: Valor a la negociación contractual*

Sin embargo, no basta con aplicar enfoques o prácticas ágiles, sino que resulta necesario evaluar el impacto de su implementación en los proyectos. En el caso de la comunicación con el cliente, aproximadamente el 85% de las organizaciones y equipos de proyectos ágiles entrevistados en el relevamiento de datos en el contexto actual, han manifestado que de los factores que peor impactan en la calidad del software se destacan, las necesidades mal definidas y la poca participación del cliente. La mayoría coincide en que cuando la relación del cliente se reduce solo a definición del contrato, esto conlleva a aumento de costos asociados y problemas de incomodidad entre el Cliente y el equipo generando, muchas veces, suspensión de proyectos sin lograr su implementación.

Entonces, se describen a continuación, en la Tabla 3.27 la métrica directa que permitirá obtener el valor asociado a este atributo:

Tabla 3.27 Descripción de Métrica N° 3.6

Métrica N° 3.6	
Tipo	Directa
Nombre	Importancia a la relación con el cliente
Abreviatura	#Rel
Objetivo	Determinar qué tipo de relación se establece entre el equipo y el cliente
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Se mide el tipo de relación que se ha definido entre el equipo y el cliente: a) Solo por contrato b) Contrato, relevamiento de necesidades y aceptación del producto c) Contrato, relevamiento de necesidades, sprint review y aceptación del producto d) Forma parte activa de todo el proyecto
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Finalmente, en la Tabla 3.28 se consignan las correspondencias entre criterios y valores asociados a los atributos del Componente N° 3.

Tabla 3.28 Relación entre atributos y criterios del Componente N° 3

Componente 3: Comunicación con el Cliente			
Atributo Positivo		Atributo Negativo	
A3.1 Valor a la colaboración con el cliente		A3.2 Valor a la negociación contractual	
<i>Criterios de medición para A3.1</i>	<i>Valor</i>	<i>Criterios de medición para A3.2</i>	<i>Valor</i>
3.1.1 El Cliente es parte del equipo, participa de los sprints planning y redacta las historias de usuario	4	3.2.1 El Cliente solo es considerado desde el punto de vista contractual y no forma parte del equipo	-4
3.1.2 El Cliente es parte del equipo, redacta la mayoría de las historias de usuario y participa en la mayoría de los sprints planning	3	3.2.2 El Cliente ha tomado contacto con el equipo únicamente al iniciar y al finalizar el proyecto	-3
3.1.3 El Cliente es parte del equipo pero solamente participa en la mayoría de los sprint planning	2	3.2.3 El Cliente define el Contrato pero además contacta con el equipo al finalizar cada sprint	-1
3.1.4 El Cliente colabora únicamente demanda del equipo	1	3.2.3 El Cliente define el Contrato y forma parte activa del equipo de proyecto	0
3.1.5 El Cliente no forma parte del equipo	0		

Fuente: de elaboración propia

Componente N° 4 “Gestión de Requerimientos”. Entre los factores que afectan a la calidad de software y que han sido identificados como relevantes, tanto en la revisión sistemática de la literatura como en el relevamiento de datos en organizaciones y equipos ágiles, se destacan los referidos a la gestión de requerimientos y requisitos, los cuales son expresados a través de historias de usuario, la notación más utilizada en Ingeniería de Requerimientos ágiles [72]. En este sentido para QuAM, la calidad en la definición de las historias de usuario se evalúa basándose en las características propuestas por el modelo INVEST [73], donde se establece que cada historia de usuario debe ser: Independiente, Negociable, Valiosa, Estimable, Pequeña y Testeable. La evaluación de la redacción de historias de usuario se ha enfocado en el uso de template, teniendo en cuenta las recomendaciones de la comunidad *agile* y que permite colaborar en la determinación del nivel de construcción del entendimiento acerca del problema, describiendo la persona que detalla la necesidad junto a la salida esperada.

Particularmente, el objetivo del Componente N°4 “Gestión de Requerimientos” busca enfatizar la práctica ágil de definir necesidades como historias de usuario, de forma tal de lograr, luego, su conversión a pruebas de aceptación automatizadas, conduciendo al equipo a la utilización de técnicas como ATDD (Desarrollo guiado por pruebas de aceptación) [74] 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, valorando el software que funciona, por encima de la documentación exhaustiva y la respuesta al cambio, por encima del seguimiento de un plan algo que las organizaciones entrevistadas entienden como

fundamental para el proceso de mejora continua promoviendo la calidad de los proyectos ágiles.

Para estudiar al Componente N° 4 “Gestión de Requerimientos” se han definido dos atributos que se describen a continuación:

- *Atributo Positivo: Valor a la completitud de Requerimientos*

El framework AQF establece que los niveles de calidad de los proyectos ágiles mejoran si se promueve la práctica de contar con requerimientos bien definidos y completos, de forma tal de ofrecer la mayor información posible sobre la necesidad a satisfacer.

En este contexto, las métricas que se han diseñado para la evaluación de este atributo se describen en las Tablas 3.29, 3.30, 3.31, 3.32, 3.33 y 3.34.

Tabla 3.29 Descripción de Métrica N° 4.1

Métrica N° 4.1	
Tipo	Directa
Nombre	Cantidad de historias de usuario que se adecúan al template
Abreviatura	#UST
Objetivo	Medir el grado de adecuación del proyecto a la correctitud de historias de usuarios en su definición.
<i>Procedimiento de Medición</i>	
Tipo	Objetivo
Especificación	Por cada una de las historias de usuario se verifica que la redacción se ajuste con la estructura predefinida preferentemente en voz activa “Como [rol] Quiero [qué] Para [objetivo]”. Si es positivo se cuenta 1, sino no se cuenta. Así finalmente se obtiene el total de historias de usuario que se adecúan a ello
<i>Escala Numérica</i>	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.30 Descripción de Métrica N° 4.2

Métrica N° 4.2	
Tipo	Indirecta
Nombre	Adecuación de historias de usuario al template
Abreviatura	%UST
Objetivo	Obtener el porcentaje de historias de usuario que se adecúan al template propuesto por la comunidad.
Fórmula de cálculo	$\%UST = \frac{\#UST \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Tabla 3.31 Descripción de Métrica N° 4.3

Métrica N° 4.3	
Tipo	Directa
Nombre	Grado de inclusión de criterios de aceptación en historias de usuario
Abreviatura	#ACUS
Objetivo	Obtener el total de historias de usuario que incluyen criterios de aceptación
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Del total de historias de usuario del Product Backlog contar aquellas que incluyan criterios de aceptación. Si la historia de usuario incluye criterios de aceptación se cuenta "1", por el contrario no se cuenta.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.32 Descripción de Métrica N° 4.4

Métrica N° 4.4	
Tipo	Indirecta
Nombre	Porcentaje de inclusión de criterios de aceptación en Historias de Usuario
Abreviatura	%ACUS
Objetivo	Obtener el porcentaje de historias de usuario del Product Backlog que incluyen criterios de aceptación.
Fórmula de cálculo	$\%ACUS = \frac{\#ACUS \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Tabla 3.33 Descripción de Métrica N° 4.5

Métrica N° 4.5	
Tipo	Directa
Nombre	Cantidad de historias de usuario priorizadas
Abreviatura	#USP
Objetivo	Obtener el número total de historias de usuario a las que se ha asignado prioridad en base al método MOSCOW [75]
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Se observan las historias de usuario que conforman el Product Backlog y se verifica que hayan sido priorizadas. En ese caso se cuenta "1", caso contrario no se cuenta.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.34 Descripción de Métrica N° 4.6

Métrica N° 4.6	
Tipo	Indirecta
Nombre	Porcentaje de historias de usuario que han sido priorizadas
Abreviatura	%USP
Objetivo	Obtener el porcentaje de historias de usuario del Product Backlog que han sido priorizadas.
Fórmula de cálculo	$\%USP = \frac{\#USP \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

- *Atributo Negativo: Valor a la dependencia entre requerimientos*

En función a lo relevado en las organizaciones y a la observación de equipos trabajando en procesos ágiles de software, se concluye que los proyectos se ven seriamente retrasados en los tiempos cuando se definen historias de usuario dependientes entre sí, es decir historias que para estar finalizadas dependen de la finalización de otras dentro del mismo sprint. De hecho, resulta razonable evitarlo, pues la descentendencia añade un punto más de complejidad a la planificación de realización de tareas, obligando a estar muy pendiente de estas interrelaciones y definir una correcta priorización en la realización de las mismas. Asimismo, se pudo determinar que la calidad se ve disminuida cuando aparecen requerimientos dependientes dentro de una misma iteración porque no solo dificulta la estimación del proyecto, sino que añade disconformidades a nivel de equipo, pues muchas veces conduce a subocupación de participantes del proyecto.

Para cuantificar el efecto de este atributo sobre la calidad del proyecto ágil se han diseñado las métricas que se describen a continuación en las Tablas 3.35, 3.36 y 3.37.

Tabla 3.35 Descripción de Métrica N° 4.7

Métrica N° 4.7	
Tipo	Directa
Nombre	Total de sprints
Abreviatura	#SP
Objetivo	Obtener el número de sprints que componen al proyecto en estudio.
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Se cuentan la cantidad de sprints definidos hasta el momento en el proyecto.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.36 Descripción de Métrica N° 4.8

Métrica N° 4.8	
Tipo	Directa
Nombre	Total de sprints con definición de historias de usuario dependiente
Abreviatura	#SPDep
Objetivo	Obtener información respecto a cantidad de sprints que incluyen dependencias en la definición de sus historias de usuario.
Procedimiento de Medición	
Tipo	Objetivo
Especificación	Si el inicio de la historia de usuario dentro de un sprint depende de la finalización de otra historia de usuario dentro del mismo sprint se cuenta 1, sino no se cuenta.
Escala Numérica	
Representación	Discreta
Valor	Entero
Tipo	Absoluto

Fuente: de elaboración propia

Tabla 3.37 Descripción de Métrica N° 4.9

Métrica N° 4.9	
Tipo	Indirecta
Nombre	Porcentaje de sprints con dependencias
Abreviatura	%SPDep
Objetivo	Obtener el porcentaje de sprints que hayan definido historias de usuario con dependencia en todo el proyecto
Fórmula de cálculo	$\%USDep = \frac{\#USDep \times 100}{\#US}$
Escala Numérica	
Representación	Continua
Valor	Real
Tipo	Proporción

Fuente: de elaboración propia

Finalmente, en función al proceso de medición que se realice utilizando las métricas antes definidas, QuAM establece la correspondencia de valores para los criterios en su Componente N° 4 de acuerdo con lo que se muestra en la Tabla 3.38

Tabla 3.38 Relación entre atributos y criterios del Componente N° 4

Componente 4: Gestión de Requerimientos y Requisitos			
Atributo Positivo		Atributo Negativo	
A4.1 Valor a la completitud de requerimientos		A4.2 Valor a la dependencia entre requerimientos	
<i>Criterios de medición para A4.1</i>	<i>Valor</i>	<i>Criterios de medición para A4.2</i>	<i>Valor</i>
4.1.1 Todas las historias de usuario se encuentran redactadas de manera completa teniendo en cuenta que su redacción se realiza de acuerdo con el template, incluye criterios de aceptación y priorización.	4	4.2.1 Todos los Sprints incluyen al menos una dependencia entre las historias de usuario que lo integran.	-4
4.1.2 La redacción de historias de usuario se realiza de acuerdo con el template e incluye criterios de aceptación y priorización en al menos el 90% del total de historias de usuario definidas.	2	4.2.2. Más del 60% de los sprints incluyen historias de usuario con dependencias	-2
4.1.3 La redacción de historias de usuario se realiza de acuerdo con el template, incluyen criterios de aceptación y se encuentran priorizadas en hasta un 60% del total de historias de usuario.	1	4.2.3 Se definen dependencias en menos del 60% de los sprints	-1
4.1.4 La redacción de ninguna historia de usuario hace uso del Template, no se encuentran priorizadas y tampoco incluyen criterios de aceptación	0	4.2.4 Ningún sprint incluye dependencias entre sus historias de usuario	0

Fuente: de elaboración propia

3.2.3 Procedimiento de evaluación en QuAM

QuAM define un procedimiento de evaluación que se aplica a cada uno de los 4 *Componentes* del modelo, y se caracteriza por las fases que se muestran en la Figura 3.7.

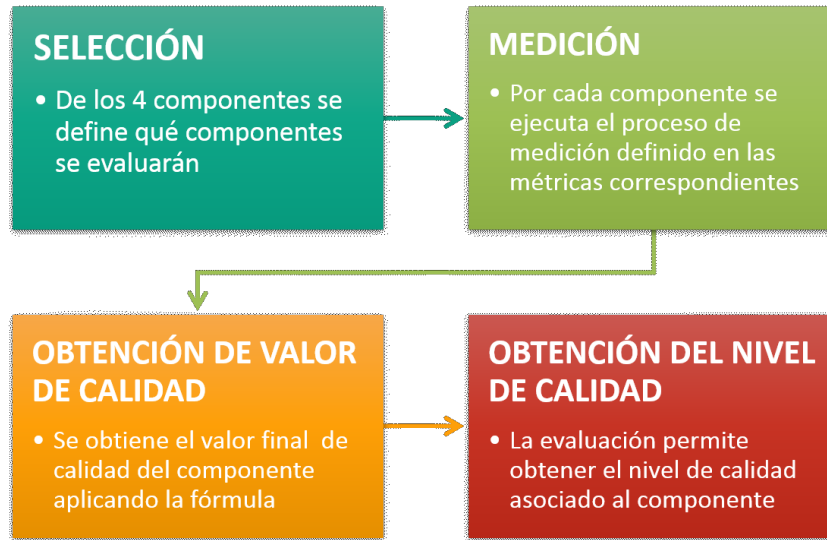


Figura 3.7 Fases del Procedimiento de evaluación de QuAM

Fuente: de elaboración propia

Fase 1) Selección

En esta fase se debe seleccionar el/los *componentes/s* que serán evaluados en el proyecto ágil.

Fase 2) Medición

De acuerdo con el *componente*, se ejecutan los procedimientos de medición correspondientes a cada *métrica* y se obtienen valores para los *atributos*.

Luego de la *medición*, los *atributos* podrán obtener valores, que serán asociados a *criterios*, correspondientes a una escala de 0 a 4 (si se trata de *atributos positivos*) o de -4 a 0 (si se trata de *atributos negativos*). Estas escalas se definieron en función a la cantidad promedio de *criterios* definidos por cada atributo, y, su signo se corresponde a si representan umbrales que afectan de manera positiva o negativa a la evaluación de calidad.

Fase 3) Obtención del valor de calidad

Según lo especificado en (1), una vez que se realiza la medición de ambos atributos, se calcula el *valor final de la evaluación de calidad* del componente (C_i) realizando la suma de los valores obtenidos en los atributos (A_i):

$$\text{Calidad de } C_i = \sum A_{i_n} \text{ donde } A_i \text{ son valores de atributos de } C_i, \\ \text{con } i = 1..4 \text{ y } n = 1..2 \quad (1)$$

Cada *componente* podría obtener, entonces, 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, significa que en ese componente la calidad del proceso no valora significativamente el atributo positivo por sobre el negativo ni viceversa.

Fase 4) Nivel de Calidad

Con el fin de complementar la evaluación y ofrecer información más comprensible por los equipos de cada proyecto ágil, el procedimiento involucra una fase de interpretación de los valores de calidad obtenidos por cada componente.

Para esto, de acuerdo con lo que se presenta en [61], se definen *indicadores globales*, los cuales permiten evaluar cada componente a un nivel de abstracción más alto, a través de la interpretación de los valores obtenidos por la medición previa. El uso de *indicadores globales* contribuye a conocer el nivel de satisfacción alcanzado por los *atributos*, caracterizando, de este modo, a cada componente de QuAM.

De acuerdo con el valor obtenido en cada componente, se lo categoriza en función a la definición de los siguientes *indicadores globales*:

- MUY MALO: -4 a -3
- MALO: -2 a -1
- BUENO: 0 a 1
- MUY BUENO: 2 a 3
- EXCELENTE: 4

Teniendo en cuenta la definición de componentes, atributos y criterios expuestos en la sección 3.1.2.2 de este capítulo, a continuación, se presenta un ejemplo en la tabla 3.39.

Tabla 3.39 Ejemplo de evaluación de calidad según QuAM

Evaluación del Componente N° 3 “Comunicación con el Cliente”	
Criterio asignado al atributo A4.1 <i>Valor a la completitud de requerimientos</i>	3
Criterio asignado al atributo A4.2 <i>Valor a la dependencia entre requerimientos</i>	-1
Nivel de Calidad obtenido	MUY BUENO 2

Fuente: de elaboración propia

3.2.4 Discusión

Hasta aquí el Capítulo ha presentado el diseño y discusión de una instancia del modelo QuAM, describiendo cada uno de sus componentes, atributos y métricas asociados, junto al procedimiento de evaluación definido.

Si bien, como se presentó en el Capítulo 2, existen en la literatura varios trabajos que, con el objetivo de mejorar la calidad de los procesos de desarrollo de software, presentan propuestas para adaptar normas y estándares a la filosofía ágil; no se enfocan específicamente en la evaluación de la calidad de procesos bajo enfoques ágiles. Así, en principio, la presentación de QuAM como aproximación a un nuevo modelo de calidad permitirá iniciar la evaluación de calidad en proyectos reales guiados por enfoques ágiles de desarrollo de software.

La definición de QuAM, tal y como se presentan en esta sección, contribuirá a obtener una mirada en detalle respecto a qué cuestiones podrían afectar la calidad en el proceso ágil que llevan adelante las organizaciones para la obtención de sus productos de software.

Sin embargo, para la evaluación de calidad de proyectos ágiles que propone AQF no basta con la definición de un modelo, sino que resulta necesario automatizar la gestión de los componentes de QuAM a partir de una herramienta de software que permita el seguimiento de proyectos acompañado por la obtención de informes parciales respecto de la calidad del proceso ágil que lo caracteriza.

A continuación, se presenta el esquema y arquitectura de una de las aplicaciones que servirá de soporte a QuAM y formará parte de la suite de herramientas del framework que se propone en el marco de esta tesis.

3.3 - QuAGI: Aplicación Web para la automatización de QuAM

Contar con un modelo de calidad, que permita evaluar el nivel de calidad de un proceso ágil, no es suficiente si no se dispone de una herramienta que posibilite: gestionar los elementos del modelo de manera automatizada y presentar los resultados obtenidos de manera clara y precisa para facilitar su análisis. Por ello, a fin de complementar el framework AQF, se aborda el desarrollo de QuAGI, una aplicación web que permite la gestión de los componentes de QuAM a través de la medición y obtención automática de los valores correspondientes a cada una de las métricas a lo largo del proceso de desarrollo. El objetivo central de esta herramienta es permitir el seguimiento online 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.

El proceso de desarrollo de QuAGI está guiado por las siguientes premisas:

- Evaluación de calidad precisa y automática sin que los responsables de proyectos ágiles deban recordar la configuración de cada componente de QuAM.
- Automatización del seguimiento de proyectos proporcionando informes de calidad completos.
- Definición de un mecanismo confiable e integrado a los proyectos ágiles para poder validar el modelo QuAM.

Además, QuAGI fue diseñada como una aplicación web escalable de forma tal que, permita añadir nuevas funcionalidades de manera rápida y sencilla a partir de reportes de feedback que surjan luego de escenarios de validación que se describirán en el capítulo siguiente de esta tesis.

Por lo tanto, QuAGI permite administrar proyectos ágiles, realizar el seguimiento de cada uno de ellos, obtener diversos informes y constituye una herramienta de comunicación interna entre los integrantes del equipo. Por último, da soporte a los procesos de toma de decisiones asistiendo a los responsables mediante reportes que informen sobre evaluación de calidad del proyecto y recomendaciones de ajustes para la mejora continua [76].

Para una mejor comprensión, se exponen a continuación en la sección 3.3.1 características del proceso de desarrollo de QuAGI y se incluye la especificación de requerimientos funcionales y no funcionales que representan la base sobre la cual se realiza la implementación posterior; luego en la sección 3.3.2 se presenta la arquitectura subyacente de QuAGI; para luego, en la sección 3.3.3, describir aspectos técnicos junto a detalles de su implementación. Y, finalmente en la sección 3.3.4 se presentan conclusiones de esta sección.

3.3.1 Características del proceso de Desarrollo

QuAGI, como se ve en la Figura 3.8, es una aplicación web que ha sido diseñada siguiendo el enfoque de *Diseño Centrado en el Usuario* (DCU) [77], por lo cual desde su inicio se han tenido en cuenta aspectos relacionados a la usabilidad del producto pues resulta necesario que el equipo trabaje de forma sencilla, práctica y amigable. Por ello, QuAGI ha sido validada en primer lugar por el equipo de desarrollo de la herramienta y, en segundo lugar, por las organizaciones que forman parte de los estudios de caso que se presentarán más adelante en esta Tesis.

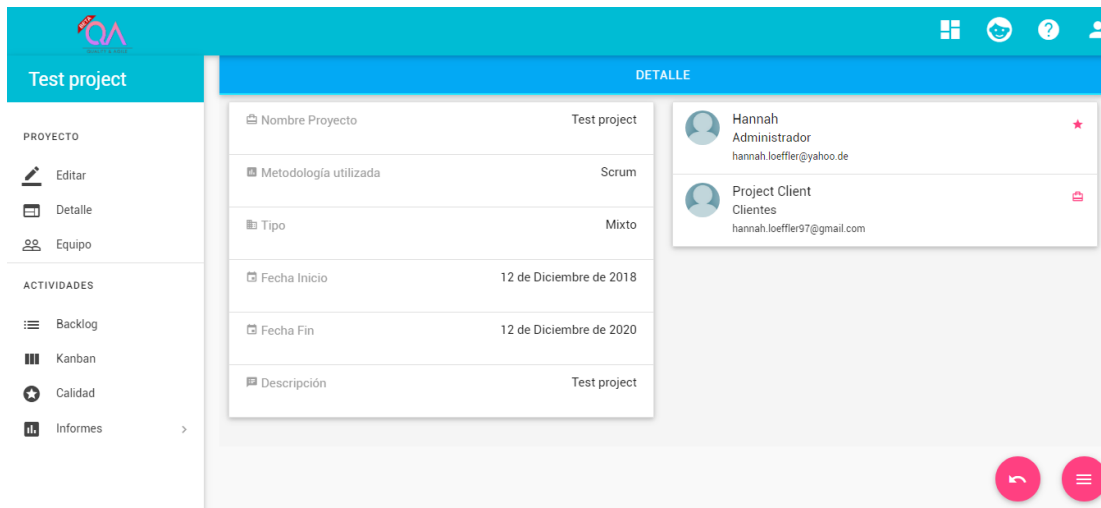


Figura 3.8 Aplicación web QuAGI

Fuente: de elaboración propia

Asimismo, el proceso de desarrollo de QuAGI se caracterizó por la combinación de prácticas ágiles que caracterizan a los enfoques de SCRUM y Kanban. De esta manera el proyecto, que tuvo una duración aproximada de 11 meses, contempló el diseño, desarrollo e implementación de la aplicación web y estuvo compuesto por 22 *sprints* con una duración de 15 días cada uno.

Debido a que se trabajó con un equipo distribuido, el proyecto de desarrollo de QuAGI no se caracterizó en función a un único enfoque ágil, sino que se fusionaron prácticas de SCRUM y Kanban, pero a su vez se diseñaron nuevas prácticas adaptables a la realidad del equipo. Por ejemplo, si bien se llevaron a cabo *sprints planning* su ejecución se realizó aproximadamente cada 60 días, de forma tal que cada instancia configuraba entre 3 y 4 sprints. Además, debido a dificultades para que el equipo se reúna cada día, en reemplazo de las daily meetings se realizaban *weekly meetings* en las cuales al inicio de cada semana los miembros del equipo comentaban sobre en qué venían trabajando, qué harían luego y, en caso de que hubiera, se planteaban dificultades que podrían afectar al proyecto.

El ciclo de desarrollo, tal como se observa en la Figura 3.9, se compone de 4 *épicas* que guiaron el proceso de desarrollo e implementación de las funcionalidades que caracterizan a QuAGI, independientemente de las tecnologías y su arquitectura, más bien desde un punto de vista de lógica de negocio.



Figura 3.9 Épicas que componen el proyecto QuAGI.

Fuente: de elaboración propia

El conjunto de épicas se descompone en un subconjunto de historias de usuario que describen los requerimientos funcionales y cada una de las cuales incluye en su definición criterios de aceptación, especificando requerimientos no funcionales, junto a las tareas que guían su implementación. Para más información, en el *Anexo D* de esta Tesis se describen cada una de las funcionalidades implementadas en QuAGI.

3.3.2 Diseño de la arquitectura

El objetivo del enfoque integral de QuAGI es optimizar la gestión de proyectos de software basados en prácticas ágiles teniendo en cuenta la calidad del proceso y del producto final, según lo indicado por QuAM. La arquitectura propuesta consiste en una plataforma basada en componentes reutilizables, permitiendo así que las aplicaciones que se integren a la plataforma hagan uso de dichos componentes.

Tal como se observa en la Figura 3.10, la propuesta de la arquitectura conceptual para QuAGI se ha diseñado siguiendo los principios de separación en 3 capas: la capa de presentación, la capa lógica y la capa de persistencia [78]. Cada una de las capas representa una vista particular de la arquitectura de QuAGI, y esta configuración propuesta permite asegurar el mantenimiento de la trazabilidad entre los diferentes artefactos desarrollados, la reutilización de estos y el mejor control de su evolución a la hora de incorporar nuevas funcionalidades. Además, se presenta en esta arquitectura un flujo constante de información entre cada una de las capas produciéndose una *retroalimentación bidireccional*, por ejemplo, la información que se envía desde la capa lógica respecto a valores obtenidos en los cálculos de las métricas permitirá que la capa de presentación muestre a través de las interfaces el resultado de Evaluación de calidad final asociado al proyecto ágil que se gestiona.

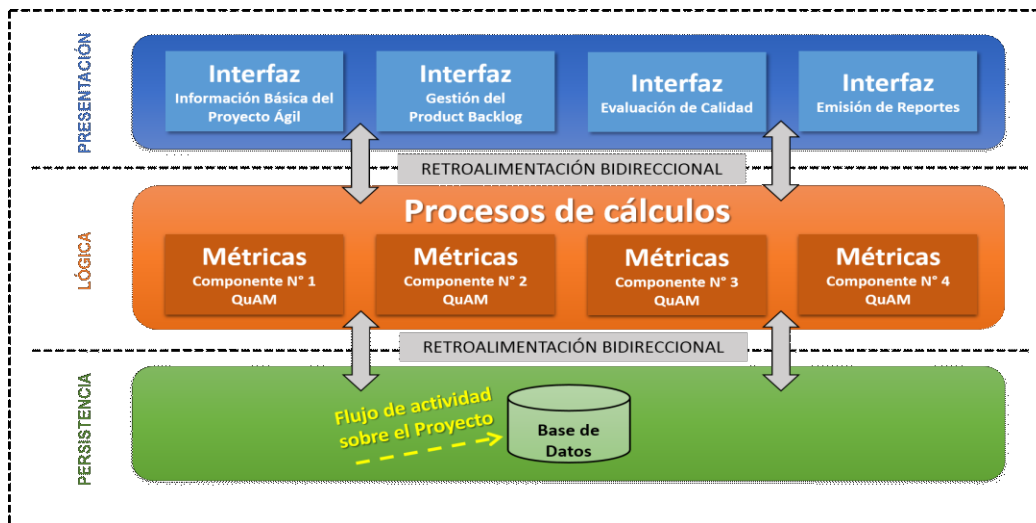


Figura 3.10. Arquitectura conceptual de QuAGI

Fuente: de elaboración propia

La **capa de presentación** es la capa con la cual el usuario interactúa directamente. Por ello, en esta capa se incluyen todas las interfaces que componen a QuAGI y a través de las cuales el equipo de desarrollo hace uso de las funcionalidades de la herramienta.

En la **capa lógica** se ubica el proceso de cálculo definido por las métricas del modelo y que se automatizan en QuAGI a partir de la información generada en la interacción con los usuarios. En esta capa es, además, donde se incluye el diseño del proceso de obtención del valor de calidad por cada componente.

Por último, la **capa de persistencia** desde donde se manipulan los datos, tanto su almacenamiento como su recuperación y acceso. A efectos de no dificultar la lectura de este trabajo, se muestra en la Figura 3.11, la representación gráfica de un diagrama que resume el conjunto de entidades que participan en los procesos diseñados en la capa de persistencia y que permitirán la gestión de información relativa a los proyectos ágiles que se evalúen.

Claramente, en la Figura 3.11, se distinguen dos grupos de entidades que apoyan la arquitectura de QuAGI. Por un lado, se encuentran agrupadas las entidades referentes a la *Gestión del Proyecto ágil* cuyos datos serán gestionados y tratados a través de QuAGI. Y, por otro lado, toda la información que se genere como flujo de actividades en el Proyecto será utilizada por el grupo de entidades referidas a *Evaluación de Calidad* del proyecto en cuestión para generar los cálculos y mediciones necesarias. En la representación gráfica de la Figura 3.11 se observa que desde cada grupo de entidades se solicita o se envía información en función al rol que éstas representan en el proceso de seguimiento y medición de acuerdo con lo establecido en el modelo QuAM.

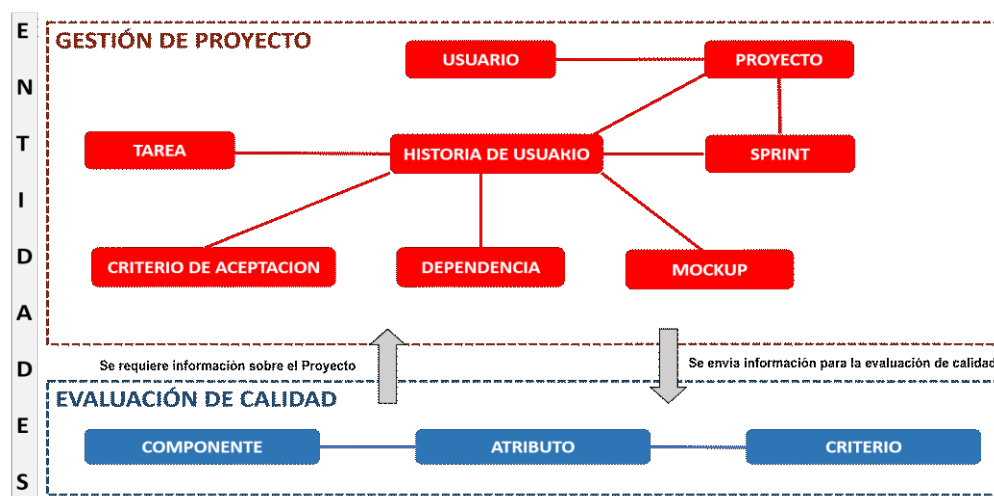


Figura 3.11 Relación entre entidades del modelo en la capa de persistencia de QuAGI

Fuente: de elaboración propia

En el caso del grupo correspondiente a *Gestión de Proyectos*, la entidad **USUARIO** representa cada uno de los perfiles que interactuarán con proyectos ágiles a través de QuAGI. Esta entidad se relaciona con **PROYECTO**, que representa el conjunto de datos que identifica a un proyecto ágil, y que está asociado con la entidad **SPRINT**, que representa información referida a cada una de las iteraciones que conformen a ese proyecto. La entidad **HISTORIA DE USUARIO**, representa el conjunto de requerimientos funcionales definidos para el proyecto y se relaciona con:

- la entidad **TAREA**, que representa actividades necesarias para llevar a cabo la realización de la HISTORIA DE USUARIO.
- la entidad **CRITERIO DE ACEPTACIÓN**, que representa condiciones establecidas por el Product Owner para cada HISTORIA DE USUARIO.
- la entidad **DEPENDENCIA**, que representa relaciones entre instancias de la entidad HISTORIA DE USUARIO de forma que la realización de una dependa de la realización de otra en un mismo SPRINT.
- la entidad **MOCKUP**, que representa información que identifica imágenes de prototipo de interfaces relacionadas a cada HISTORIA DE USUARIO.

En el grupo de entidades correspondientes a *Evaluación de Calidad* se observa a la entidad **COMPONENTE** que representa a cada aspecto a evaluar de acuerdo con lo establecido por QuAM. Esta entidad se relaciona directamente con la entidad **ATRIBUTO**, que representa la información necesaria para que puedan realizarse las mediciones correspondientes a cada **COMPONENTE**. El resultado de cada medición será gestionado desde la entidad **CRITERIO**.

Cabe aclarar que este modelo conceptual sirve de base para que luego, a nivel físico, se realice la implementación de la lógica necesaria y el diseño de tablas en la base de datos que utiliza QuAGI para su funcionamiento. En el *Anexo E* de esta Tesis se incluye cómo se efectiviza la integración entre el modelo conceptual QuAM y su implementación técnica en QuAGI.

3.3.3 Discusión

Durante la sección 3.3, se presenta a QuAGI, una aplicación web que automatiza la medición de los componentes definidos en el modelo QuAM y que permite, no solo el seguimiento de proyectos ágiles, sino también que proporciona información relativa a evaluación de calidad asociado a éste.

Considerando que existen actualmente en el mercado muchas herramientas que permiten gestionar proyectos de software (por ejemplo, Trello, Taiga.io, Jira, entre otras), QuAGI, a diferencia de las existentes, además forma parte del framework AQF, por lo que no solo permite el seguimiento online de los proyectos de software, sino que también da soporte a lo establecido en un modelo de calidad propio y permite, entonces, realizar evaluación de calidad del proyecto en cuestión. Así, actualmente QuAGI presenta a sus usuarios una herramienta web que permite realizar:

- *Configuración inicial del proyecto ágil,*
- *Gestión del Product Backlog,*
- *Evaluación de calidad y*
- *Emisión de reportes.*

Cabe destacar que QuAGI almacena constantemente la interacción de los usuarios con la herramienta; por lo que, cuando se requiere conocer sobre el nivel de calidad del proyecto, QuAGI recupera dicha información, realiza las mediciones de todos los componentes, definidos por QuAM, y obtiene la evaluación de forma automática.

Una de las contribuciones más importantes derivadas de esta tesis se enfoca en la posibilidad de incorporar, al seguimiento de proyectos ágiles, el proceso de evaluación de calidad asociado al ciclo de desarrollo que se pretende gestionar. Y con la presentación de QuAGI en esta sección ha sido posible mostrar, a partir de un enfoque práctico y técnico, cómo subyace en la plataforma el marco conceptual establecido por QuAM.

Finalmente, resulta importante destacar que, como resultado del desarrollo de esta plataforma web QuAGI la misma ha sido registrada, durante el desarrollo de la tesis, como Obra Inédita de Software ante la Dirección Nacional de Derecho de Autor dependiente del Ministerio de Justicia y Derechos Humanos de la República Argentina¹.

¹ El Registro se ha publicado en el Boletín Oficial del Poder Ejecutivo Nacional el 25 de junio de 2019, con el Número RE-2019-57162220-APN-DNDA-#MJ

3.4 – El valor agregado de AQF

Como se analizó en el Capítulo 2 de esta tesis, existe en la literatura disponible una diversidad de aproximaciones relacionadas a la evaluación de calidad en proyectos de software guiados por prácticas ágiles; sin embargo, estas propuestas no analizan factores que impactan positiva o negativamente sobre la calidad del proceso, ni tampoco ofrecen una visión integradora que permita definir estrategias transversales para mejorar el proceso de la evaluación de calidad que se estudia

Por ello, y luego de la descripción de la propuesta de modelo de calidad QuAM junto a QuAGI, la herramienta que automatiza la gestión y evaluación de calidad de los proyectos ágiles, a continuación, se destacarán cuáles son las características que permiten diferenciar a AQF del resto de las iniciativas existentes. Estas características no solo benefician a la calidad de los procesos de desarrollo, sino que, asimismo, ayudan a tomar decisiones que añaden valor agregado a los proyectos de software.

- **Propuesta integral**

Un aspecto clave de AQF, es que representa una propuesta que integra el *seguimiento de un proyecto de software y la evaluación de calidad del proceso ágil* asociado, en una única plataforma.

A diferencia de trabajos relacionados, que se enfocan unilateralmente en algunos aspectos del ciclo de desarrollo de software, AQF observa el proceso ágil desde 4 factores: *producción de entregables, equipo de trabajo, comunicación con el cliente y gestión de requerimientos*, coincidiendo con los valores de la filosofía ágil en la gestión de proyectos. AQF, obtiene una visión integral de la implementación del proceso ágil, a lo largo de todo el ciclo de desarrollo, reforzando la información de la evaluación de calidad obtenida con informes útiles para la toma de decisiones.

Asimismo, en los trabajos estudiados, a través de la revisión sistemática presentada en el Capítulo 2, se observó que se centraban, solamente, en la evaluación de la implementación del enfoque ágil (y de un único enfoque) sin aportar valor al proceso de desarrollo. De forma contrapuesta, AQF, se independiza del enfoque utilizado, y se focaliza en las prácticas ágiles asociadas al proyecto de software, resaltando aspectos que contribuyen a mejorar la calidad del proceso fortaleciendo la implementación del enfoque ágil en la organización y potenciando la obtención del producto de software.

En simples términos, AQF se presenta como una propuesta integral que, dando soporte a las organizaciones que utilizan algún enfoque ágil para el desarrollo de software, les ofrece el seguimiento de sus proyectos y la evaluación de la calidad de sus procesos, analizando cómo impactan, sobre esto, las prácticas ágiles implementadas.

- **Procedimiento de Evaluación**

De los estudios analizados en el Capítulo 2, una de las limitaciones destacadas tenía que ver con la evaluación que proponen, dado que no especifican, en su mayoría, cómo se compone el proceso de medición y se restringen, solamente, a indicar si el resultado es o no de calidad.

Por ello, y teniendo en cuenta que AQF representa una propuesta integral, se desprende otras de sus características de valor: la definición detallada de un *procedimiento de evaluación integral*, que basándose en QuAM, permite comprender el significado del nivel de calidad del proceso ágil asociado a su capacidad de producción de entregables del equipo, a la evaluación del equipo de trabajo, a la comunicación con el cliente y a la gestión de requerimientos. De hecho, el aspecto diferenciador de este procedimiento de evaluación es que la calidad de un proyecto ágil no es un valor binario, sino que puede darse en un espectro de valores que seguramente estén condicionados por condiciones del contexto donde el proyecto se lleva a cabo. Para identificar estas condiciones, los equipos de desarrollo de software, al utilizar AQF, podrán comparar los niveles de calidad entre diferentes proyectos ejecutados en contextos similares a partir de informes de calidad proporcionados por la aplicación web QuAGI.

- **Visibilidad del trabajo de equipo**

Otro beneficio de AQF, frente a otras propuestas disponibles en la literatura, es que ofrece mayor *visibilidad del trabajo* asignado al equipo. Esta característica exalta uno de los principios ágiles: la transparencia, haciendo hincapié en la importancia del acceso y conocimiento sobre la información que se maneja de cada proyecto. QuAGI propone, entonces, una visión holista de cada proyecto, mostrando, no solamente la actividad y quién tiene asignado cada tarea, sino promoviendo que cada integrante del equipo conozca todo el trabajo no finalizado en el que está participando, aun cuando todavía no tiene que realizar ninguna acción sobre este (por ejemplo, conocer, con antelación, las dependencias e impedimentos que puedan llegar a ocurrir) y acceder a información específica, respecto a cómo impacta cada uno de estos aspectos a la calidad del proyecto. Entre los trabajos analizados en la revisión sistemática del Capítulo 2, se identifican debilidades relacionadas a la falta de información que se proporciona a la organización, sobre factores que afectan a la calidad del proceso ágil de desarrollo y, por ende, del proyecto de software que se obtenga.

- **Gestión del conocimiento**

Por último, de las evidencias que existen en la literatura, hasta el momento, ninguna ofrece la posibilidad de gestionar el proyecto de software, evaluar la calidad del proceso ágil y tomar decisiones a nivel de la organización para la mejora continua. Como contraparte, AQF, a partir de la interacción de usuarios con QuAGI, permite generar conocimiento útil que se transformarán, luego, en lecciones aprendidas para toma de decisiones en proyectos actuales y futuros proyectos ágiles. A partir de los datos y de las evidencias generadas por el uso de QuAGI, se podrían analizar indicadores que permitan exponer las bondades y debilidades de los elementos intervinientes en los procesos ágiles de desarrollo de software de los proyectos evaluados. Este aporte de conocimiento permite el monitoreo y evaluación de los logros obtenidos por la ejecución de determinadas prácticas ágiles; fomentando la creación de una cultura orientada al autoaprendizaje y socialización, para permitir que las buenas prácticas y las lecciones aprendidas sean de total conocimiento de los participantes en la mejora continua de los procesos de desarrollo de software.

3.5 - Conclusiones

En este capítulo se han presentado y detallado todos los componentes, que actualmente forman parte, del framework AQF, cuyo objetivo se centra en la evaluación de calidad de proyectos ágiles. Tanto el modelo QuAM (sección 3.2) como la herramienta web QuAGI (sección 3.3), integran un nuevo enfoque de evaluación de calidad, basando la estrategia de medición en el seguimiento de actividades de proyectos ágiles¹.

En primer término, la sección 3.2 formaliza los términos y conceptos que definen al modelo de calidad QuAM que integra el framework AQF, junto a la descripción de cada uno de sus componentes en función a atributos, métricas y criterios de calidad.

Luego, en la sección 3.3 se ha desarrollado la configuración de la herramienta de software QuAGI, una plataforma web que a partir de la definición de una arquitectura conceptual implementa la gestión de los componentes del modelo QuAM posibilitando la automatización del seguimiento y evaluación de proyectos ágiles. Y, además, explica cómo se logra la integración entre QuAM y QuAGI de forma tal de aprovechar mecanismos de flujo de información que proporcionen a los usuarios del framework contar tanto con informes de Evaluación de Calidad como con reportes estadísticos respecto al seguimiento del proyecto.

Finalmente, en la sección 3.4 se exponen, beneficios que presenta AQF; destacándose los aspectos que otorgan valor agregado al framework y dan respuesta a limitaciones que se presentan en otras propuestas evaluadas.

¹ Como ya se hizo mención al inicio de este capítulo, se denomina Proyecto Ágil a aquel proyecto de desarrollo de software guiado por prácticas de enfoques ágiles.

Capítulo 4 - Validación

En el capítulo 3 de esta tesis se ha presentado el framework AQF a través de la especificación del modelo y de la herramienta web que lo compone. AQF permite realizar el seguimiento de proyectos ágiles para posteriormente procesar esa información y emplearla en la evaluación de calidad del proceso que lo caracteriza, independientemente del enfoque ágil utilizado.

A partir del enfoque metodológico presentado en el capítulo 1, y tal como lo proponen los autores Marcos & Marcos en [24], en este capítulo se describe la validación de la propuesta en base al resultado de los estudios de casos llevados a cabo a través de la implementación del framework AQF.

Retomando lo planteado en la sección 1.3 en donde se expone que el objetivo general de esta tesis doctoral es proponer un framework que facilite el seguimiento de proyectos y la evaluación de calidad de procesos en empresas PYMES que implementen prácticas ágiles en el desarrollo de software, se busca, a partir de estudios de casos, validar el modelo QuAM utilizando QuAGI como herramienta de seguimiento y evaluación de calidad, constituyendo en sí, un medio para la validación del modelo propuesto.

La validación se realiza, teniendo en cuenta cuatro aspectos: a) La información obtenida al utilizar AQF, a través de los resultados procesados por QuAGI y la interpretación provista por QuAM; b) La aplicabilidad del framework en diferentes dominios; c) El nivel de comprensión de los usuarios al obtener resultados de evaluación; d) La obtención de conclusiones de las experiencias a partir del feedback de los actores involucrados en cada caso.

Este capítulo se estructura como sigue: primero, la sección 4.1, se presentan 3 estudios de casos junto con los resultados obtenidos; seguidamente en la sección 4.2 se presentan conclusiones de este capítulo, discusiones generales y lecciones aprendidas que se derivan de la validación.

4.1 - Estudios de Casos

Como ya se ha mencionado en el Capítulo 1 de esta tesis, y según proponen Marcos & Marcos en [24], en esta sección se describe el proceso de validación de AQF. Realizando el estudio, análisis y observación de la implementación del framework sobre proyectos ágiles reales.

El proceso contempla el uso de estudios de casos, un enfoque adecuado para llevar adelante actividades de validación en ingeniería de software, ya que los objetos de estudio son *fenómenos contemporáneos*, que resultan difíciles de estudiar de forma aislada [79], como resulta ser el caso de procesos de desarrollo de software llevado a cabo por equipos que implementan, por ejemplo, prácticas ágiles en sus ciclos.

Se ha elegido trabajar con estudios de caso para que la validación de la propuesta se realice sobre entornos reales de proyectos ágiles en ejecución, de forma tal de obtener retroalimentación de los potenciales usuarios de QuAGI y al mismo tiempo poner en práctica toda la definición del método de evaluación propuesto por QuAM, lo que en sí mismo ya es un mecanismo de mejora. Además, los estudios de casos son una herramienta empírica que provee de cierto grado de libertad para definir contextos que puedan adecuarse a las necesidades de validación que fueran surgiendo.

La experiencia de validación abarcó el análisis de 2 variables independientes y subjetivas:

- *Utilidad del framework*: se refiere al grado en que los participantes consideran que el uso del proceso de evaluación establecido por QuAM y soportado tecnológicamente por QuAGI mejorará la calidad de sus proyectos ágiles.
- *Intención de uso*: se refiere a la medida en que un determinado participante tiene la intención de utilizar, en el futuro, QuAGI para el seguimiento de sus proyectos ágiles.

Por lo tanto, en este capítulo, se definen 3 estudios de casos diferentes, que se presentarán de acuerdo con el enfoque metodológico descrito en la sección 1.4 de esta tesis, en función a los siguientes aspectos:

- 1) *Diseño del Estudio*: Esta actividad comprende la definición de componentes asociados al fin del estudio, para ello se establecen 2 tareas relevantes:
 - i) *Definición de Objetivos*
 - ii) *Definición de Hipótesis*
- 2) *Preparación para la recolección de datos*: Esta actividad comprende la planificación y disposición de elementos necesarios para la ejecución de la experiencia, incluyendo:
 - i) *Especificación de la población y de la muestra*
 - ii) *Planificación de la experiencia*
- 3) *Recolección de evidencias*: Esta actividad permite generar inputs para su posterior análisis. Para la recolección de evidencias se llevará a cabo observación directa y entrevistas en cada uno de los escenarios.
- 4) *Análisis de los datos*: Esta actividad tiene por objetivo describir los resultados obtenidos desde un punto de vista cuantitativo y cualitativo.

- 5) *Conclusiones*: La última actividad en los estudios de casos se refiere a la determinación de conclusiones, esto permite relacionar los resultados obtenidos con los objetivos e hipótesis planteados en cada uno de los estudios de casos.

A continuación, se exponen, cada uno de los estudios de casos realizados para validar QuAGI a partir del seguimiento de proyectos ágiles reales y su evaluación de calidad en base a los componentes establecidos por QuAM.

4.1.1 Estudio de Caso N° 1: Validación del framework AQF

Debido a que los componentes de AQF, QuAGI y QuAM, atravesaron diversas experiencias de validación previas hasta lograr la versión actual, se hizo necesario realizar un nuevo estudio que permitiese analizar el impacto las modificaciones del framework mejorado en casos reales [80][81]. Se presenta a partir de aquí el *Estudio de Caso N° 1* con las siguientes características:

a) Diseño del Estudio

- i) *Objetivo del Estudio*: Validar la propuesta definida por QuAM en empresas PYMES de Desarrollo de Software y áreas de Sistemas de la región Nordeste de Argentina (NEA) a través de la utilización de QuAGI.
- ii) *Hipótesis*: La estructura de componentes de QuAM, automatizada a través de QuAGI, contribuye a la evaluación de calidad de procesos ágiles.

b) Preparación de la experiencia

- i) *Población y Muestra*: La *población* objeto de estudio se constituyó por equipos trabajando en proyectos ágiles dentro de empresas PYMES de Desarrollo de Software o áreas de Sistemas de organizaciones localizadas en el NEA. A partir de relevamiento en Polos IT, Clústers, Cámaras de Empresas, Organismos Estatales se obtuvo una *población total* conformada por **54 equipos de desarrollo de software**. Sobre esta población objetivo se aplicó la *técnica de muestreo aleatorio simple* definida por Otzen y Manterola en [57], realizándose entonces una evaluación proporcional acorde a la distribución por provincia en la población, obteniéndose entonces, tal como se ve en la Tabla 4.1, una *muestra* de **20 equipos**, identificando el número de elementos de acuerdo con la proporción asignada.

Tabla 4.1 Población y Muestra en el estudio de caso N° 1

Provincia	Número de Equipos	Proporción	Tamaño de Muestra
Chaco	29	54%	16
Corrientes	13	24%	3
Formosa	5	9%	0
Misiones	7	13%	1
Total	54	100%	20

Fuente: de elaboración propia

Cabe destacar que en promedio cada equipo contaba con aproximadamente 5 miembros, participando del proceso de validación un total exacto de 105 personas.

- ii) **Planificación de la experiencia:** En primera instancia se acordó, con los equipos participantes, la selección de un proyecto en ejecución y guiado por prácticas ágiles, cuyo avance e información se registró a partir del uso de la herramienta QuAGI completando una encuesta online, con un esquema de preguntas con valoración basadas en la escala de Likert [82] y conformadas tal como se presentan en la Tabla 4.2. El objetivo de la encuesta fue obtener resultados respecto a la interacción de los equipos con el framework de forma tal de validar, por un lado, los componentes del modelo QuAM y por otro, su automatización a través de QuAGI.

Resulta necesario destacar que el orden de las preguntas fue alterado de forma aleatoria con el fin de evitar el sesgo de respuesta sistemática y se reformularon algunas preguntas para convertirlas en enunciados negativos con el fin de evitar así las respuestas monótonas.

Tabla 4.2 Preguntas de la Encuesta de interacción

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

Fuente: de elaboración propia

La experiencia fue planificada para que su duración total se extendiera, como máximo, 20 días agrupándose las actividades tal y como se muestra en el Diagrama de Gantt representado en la Figura 4.1.

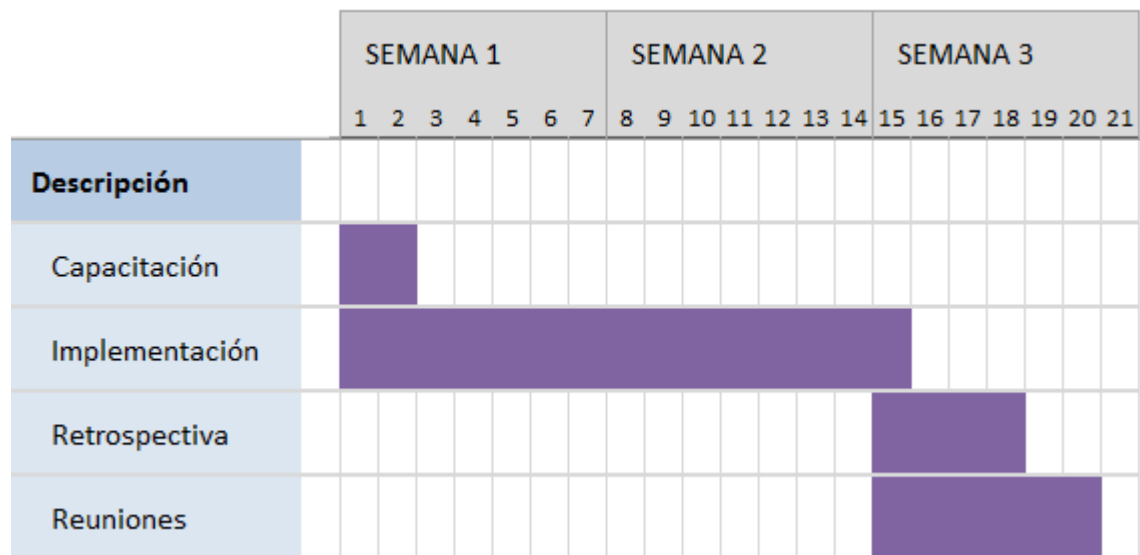


Figura 4.1 Actividades planificadas para la experiencia de validación
Fuente: de elaboración propia

De esta forma la experiencia de validación del framework en cada equipo se constituyó por las siguientes actividades:

- a) **Capacitación:** Durante 2 días, con los equipos distribuidos en ambas sesiones, se realizó la capacitación en el uso de la herramienta QuAGI en forma virtual aprovechando las ventajas que esto generaría en la vinculación con los usuarios, por ejemplo, evitando el traslado del equipo hacia otra oficina para las pruebas previas. Además, en simultáneo, se configuró el ambiente de trabajo de forma tal de ofrecer el acceso a la plataforma con la menor cantidad posible de inconvenientes técnicos.
- b) **Implementación:** Luego, se planificó que, aproximadamente durante 15 días, cada equipo debía implementar el uso del framework en sus procesos de desarrollo. En ese transcurso, cada equipo registró en su cuenta información de los proyectos ágiles (sprints, historias de usuario, mockups, etc) y realizó ajustes a través del seguimiento proporcionado por QuAGI. Se estableció, además, un acuerdo de confidencialidad de forma tal de resguardar la información correspondiente a los proyectos y considerar esto al momento de publicar o difundir datos.
- c) **Retrospectiva:** Una vez finalizado el sprint se definieron reuniones con los equipos en las cuales se realizaría el análisis de los informes

proporcionados por el framework y se registraron observaciones surgidas a partir del uso de QuAGI como herramienta de seguimiento de proyecto (estas observaciones incluso contemplaron cuestiones asociadas a la experiencia de usuario y usabilidad). Se planificaron como máximo 4 días, destinados a reuniones con los equipos, encuestas a los usuarios de QuAGI y observación in-situ del trabajo llevado a cabo.

- d) **Reuniones:** Los reportes de calidad asociados al proceso ágil evaluado fueron analizados en reuniones con los líderes de cada equipo (Project Manager, Scrum Master, Responsables de áreas) a fin de poder validar la información que el framework proporciona al equipo. Tal cual lo planificado, durante 6 días de trabajo, los líderes de equipo manifestaron observaciones especiales respecto a las experiencias y propusieron recomendaciones que contribuirían a la mejora continua de AQF.

c) Recolección de evidencias

Luego de la planificación e implementación de un sprint utilizando QuAGI, cada uno de los equipos, en función a su experiencia, completó una encuesta online al respecto y teniendo en cuenta lo consignado en la Tabla 4.2. Finalmente se analizó el impacto de AQF en la evaluación de calidad de *20 proyectos ágiles*, que implicaron el estudio de aproximadamente *110 historias de usuario* y *300 tareas asignadas* con un total de aproximadamente *105 usuarios* interactuando con el framework. Las evidencias fueron recabadas, no solo a partir de la interacción de los equipos con QuAGI, sino también a través de observación in situ del funcionamiento y de entrevistas posteriores a su uso.

d) Análisis de los datos

El estudio de los resultados de validación se enfoca en la evaluación respecto a cada uno de los componentes considerados en el modelo QuAM, ya descritos en el Capítulo 3. Por ello, luego de la implementación y del uso de la herramienta QuAGI, se llevó adelante el análisis de los datos que se habían logrado recolectar.

Se evalúa, entonces, en primer lugar el *Componente N°1 “Capacidad en la producción de entregables”*, para el cual, analizando el impacto del atributo *“Valor a la estimación”*, tal y como se observa en la Figura 4.2, el 70% de los equipos considera “Muy útil” evaluar el proceso de estimación en cada sprint, debido a que resulta ser la práctica que cuanto mejor se realiza mejor resulta el proceso de desarrollo, logrando disminuir notoriamente los desvíos en las entregas al cliente. De hecho, solo un 2% opinó que no sería relevante a la calidad del proceso ágil contemplar la estimación realizada en cada sprint, que a su vez coincide con el porcentaje de equipos que no tiene experiencia cierta con técnicas de estimación ágil.

¿Le resulta útil medir la calidad del proyecto asociado con la estimación

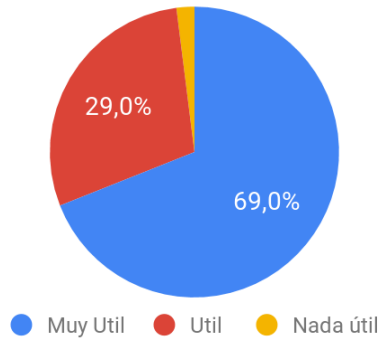


Figura 4.2 Validación del atributo positivo en el Componente N° 1 de QuAM
Fuente: de elaboración propia

Respecto al atributo “*Valor al Lead Time*” del Componente N° 1, como se ve en la Figura 4.3, se ha obtenido que el 60% de los participantes está de acuerdo en que resulta negativo desestimar el análisis de la validación que el cliente realiza sobre cada release para evaluar la calidad del proceso ágil, pues, según expresa la mayoría, la no interacción con el cliente o product owner resulta ser uno de los principales indicadores que influyen sobre los resultados de la presentación de entregables. Sin embargo, luego de la experiencia en el presente estudio del caso, un 10% considera no estar de acuerdo con evaluar negativamente este aspecto pues resulta dificultoso para los equipos lograr contar con la participación del cliente en la presentación de releases e intentar lograrlo añadiría complejidad al proceso afectando el funcionamiento del equipo.

¿Está de acuerdo en que resulta negativo no validar cada release con el cliente?

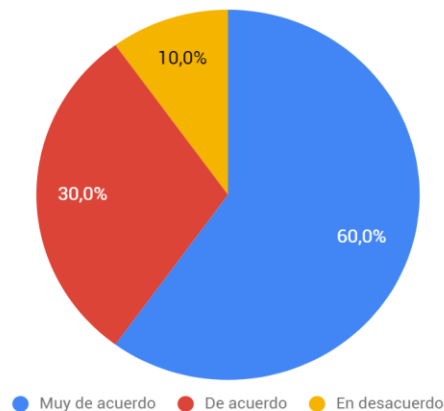


Figura 4.3 Validación del atributo negativo en el Componente N° 1 de QuAM
Fuente: de elaboración propia

Para el caso del *Componente N° 2 “Evaluación del Equipo de Trabajo”*, y respecto a su atributo positivo “*Valor a la definición de roles*”, y como se observa en la Figura 4.4, 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,

muchos 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, ocasionando problemas de comunicación y disminuyendo la productividad durante el proceso.

¿Le resulta útil medir la calidad del proyecto asociado con la definición de roles y asignación de tareas?

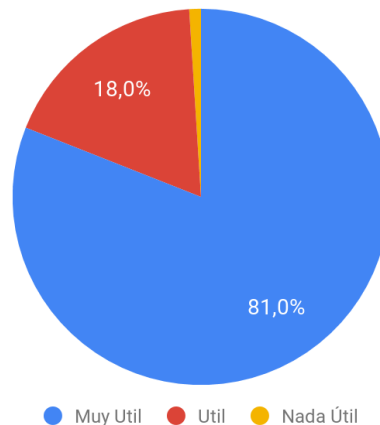


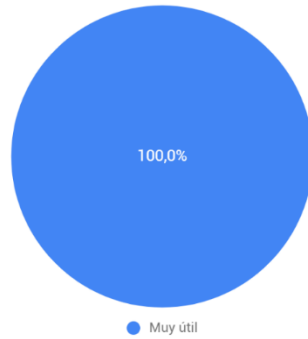
Figura 4.4 Validación del atributo positivo en el Componente N° 2 de QuAM

Fuente: de elaboración propia

Respecto al atributo negativo “*Valor al proceso por sobre el equipo*” del *Componente N° 2*, el 100% de los participantes de la experiencia han expresado a través de la encuesta que están de acuerdo en atender la configuración de sprints (historias de usuario incluidas, tareas asignadas, etc.) para la evaluación de calidad del proceso. Todos los encuestados coinciden en que limitar el proceso de desarrollo solamente a la definición de historias de usuario perjudica a la organización y autonomía del equipo, que pierde orientación al no contar con sprints completamente definidos. Sin embargo, han coincidido en que se debería ampliar el estudio a todo el proceso a fin de verificar si los ajustes introducidos en cada sprint o a partir de los informes de Calidad logran impactar en mejoras a nivel del proceso de desarrollo ágil.

En cuanto al *Componente N° 3 “Comunicación con el Cliente”*, tal como se ve en la Figura 4.5, el 100% de los participantes, tanto en la validación del atributo positivo como del atributo 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 ha sido posible validar, con este estudio de caso, su impacto al usar QuAGI, pues esta métrica solo obtiene un valor asociado a la calidad al finalizar el proyecto y el presente estudio se realizó al finalizar el primer sprint del proceso.

¿Le resulta útil considerar la participación del cliente en cada uno de los proyectos ágiles?



¿Le resulta útil considerar la participación del cliente durante todo el proceso de desarrollo de software ágil?

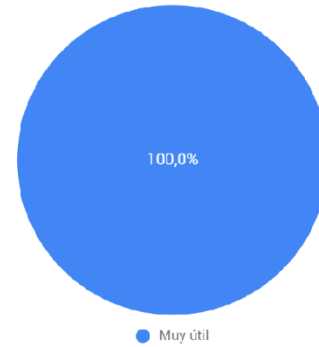


Figura 4.5 Validación del atributo positivo en el Componente N° 3 de QuAM

Fuente: de elaboración propia

Respecto del *Componente N° 4 “Gestión de requerimientos y requisitos”*; como se ve en la Figura 4.6, 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 “Nada 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.

¿Le resulta útil medir la calidad del proyecto asociado con la gestión de requisitos y requerimientos?

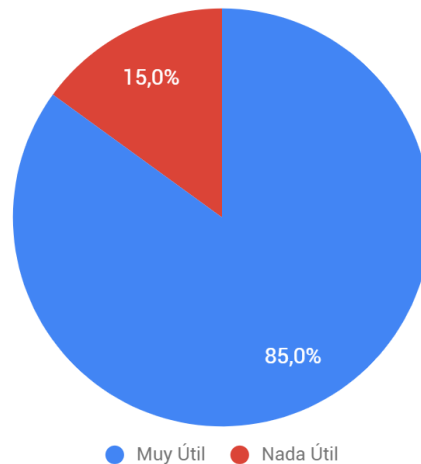


Figura 4.6 Validación del atributo positivo en el Componente N° 4 de QuAM

Fuente: de elaboración propia

En cuanto al atributo negativo “*Valor a la dependencia entre requerimientos*” del *Componente N° 4*, se observa en la Figura 4.7 que un 60% está de acuerdo que la evaluación de las dependencias en un sprint impacta negativamente en la calidad asociada al proyecto, pues explican que cada historia de usuario debería poder ser planificada e implementada en cualquier orden, por tanto, es altamente recomendable, que las historias de usuario no presenten dependencias entre si dentro del sprint. Sin embargo, el 40% manifestó no estar de acuerdo en considerar este aspecto, pues lograr que todas las historias sean independientes es una labor muy compleja que puede retrasar el proyecto innecesariamente si el equipo no tiene la experiencia suficiente para resolver las dependencias entre requerimientos.

¿Está de acuerdo en que resulta negativo definir dependencias excesivas entre requerimientos?

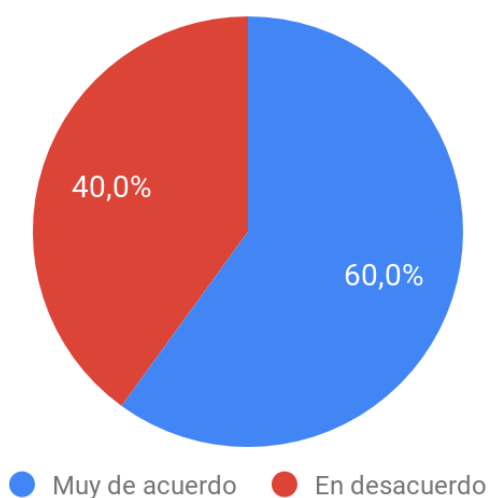


Figura 4.7 Validación del atributo negativo en el Componente N° 4 de QuAM

Fuente: de elaboración propia

Aprovechando el encuentro de retrospectiva con los equipos se llevó a cabo, además, el estudio de usabilidad de la herramienta QuAGI desde el análisis de Experiencia de Usuario (UX) en los equipos. Esto comprendió la interacción con 20 administradores de proyecto (o líderes) y 85 desarrolladores, lo que permitió obtener interesantes propuestas de mejoras, las cuales pudieron ser incorporadas a la última versión del software.

Se considera relevante, para los fines de esta tesis, destacar cuáles han sido los aspectos que influyen en la experiencia de los usuarios de QuAGI, por lo cual el resultado del estudio de usabilidad se ha tabulado teniendo en cuenta *aspectos positivos* y *aspectos negativos*, como se presenta en la tabla 4.3.

Tabla 4.3 Aspectos positivos y negativos del Estudio de usabilidad

ASPECTOS POSITIVOS	ASPECTOS NEGATIVOS
Fácil de usar	Navegación
Feedback	Vocabulario
Consistencia	Cantidad de clics

Fuente: de elaboración propia

En primer lugar, como se ve en la Figura 4.8, la usabilidad fue evaluada desde la *facilidad al usar* la interfaz de QuAGI, relacionado a que bastaron pocas horas de capacitación al equipo para presentarles las funcionalidades de la herramienta y no fue necesario ahondar en detalles técnicos para comenzar a usarla. Por el lado del Administrador del equipo o proyecto, por ejemplo, no se hizo necesario profundizar en los componentes del modelo QuAM pues la herramienta les ofrece claramente el significado del nivel de calidad alcanzado. De acuerdo a la Figura 4.8, el 95% de los líderes o administradores de proyecto coinciden en que la *facilidad de uso* es uno de los aspectos a destacar positivamente en QuAGI; por el lado de los desarrolladores, aproximadamente un 85% tiene esta misma opinión.

En segundo lugar, los participantes han destacado el *feedback* que reciben desde la plataforma web, es decir cómo QuAGI en todo momento presenta claridad respecto al proceso o etapa en la que está trabajando el usuario. De hecho, la Figura 4.7 muestra que un 93% aproximadamente de desarrolladores han coincidido en esta observación, destacando que las herramientas que ofrecen información como QuAGI se convierten en aliadas para el éxito de proyectos ágiles. En el caso de los usuarios que desempeñan rol de líder o administrador de proyecto hicieron hincapié en que si bien QuAGI ofrece información respecto al nivel de calidad alcanzado sería interesante incorporar, además, un sistema de recomendaciones para seguir mejorando el proceso ágil. Tal como se puede ver en la Figura 4.7, un 70% de los administradores han manifestado que el *feedback* de QuAGI resulta un aspecto positivo a destacar.

Por último, entre los aspectos positivos más relevante se observa, la *consistencia*, pues, de acuerdo con lo que manifiestan los usuarios, se valora mucho que la herramienta mantenga en todas sus funcionalidades la uniformidad en los términos que utiliza y que no se deban aprender conceptos similares a medida que se avance en el uso de QuAGI. Resulta importante destacar que, como se observa en la Figura 4.8, el 100% de los desarrolladores valora la *consistencia* que presenta QuAGI en todas sus interfaces, y aproximadamente un 80% de los administradores coincide en esta observación.

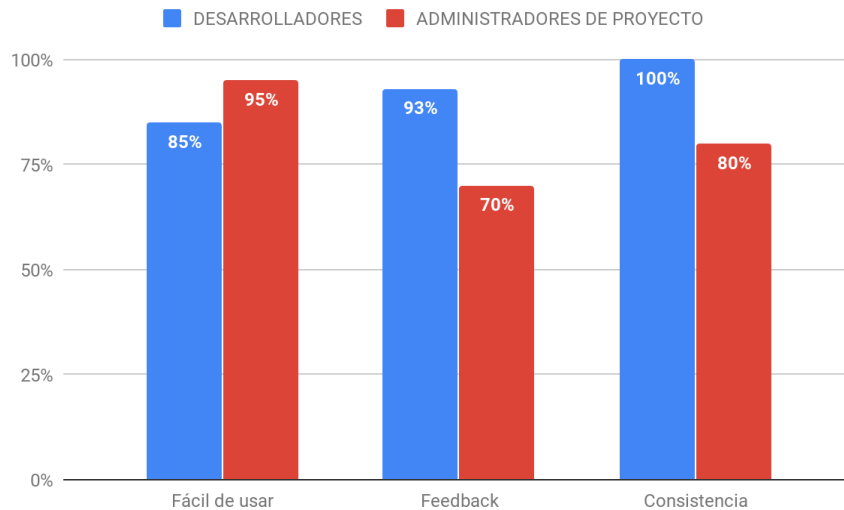


Figura 4.8 Aspectos positivos en la evaluación de usabilidad

Fuente: de elaboración propia

Respecto a los aspectos negativos, han sido difícil identificarlos y lograr determinar aquéllos en los que la mayoría de los participantes coincidieran. Aun así, los usuarios destacaron algunos aspectos a ser revisados en las próximas versiones para mejorar la experiencia de interacción entre el usuario y la herramienta de software.

El primer aspecto que destacaron ha sido la *navegación* que, en mayor medida, según lo expresan los administradores, dificultan el uso de QuAGI. Por ejemplo, cuesta mucho llegar desde una historia de usuario hasta sus criterios de aceptación debido a que se debe navegar por diferentes secciones de la plataforma para lograr el objetivo. Se observa, en la Figura 4.9, que un 78% de los desarrolladores ha considerado a la *navegación* como el aspecto negativo para revisar en QuAGI. Y, por otra parte, coinciden en esta recomendación, un 89% de los administradores de proyecto.

El segundo aspecto en el que, tanto desarrolladores como administradores del proyecto, coinciden es el *vocabulario* utilizado en QuAGI. Por ejemplo, al cargar historias de usuario, la opción “Usar Template” no queda claro a qué se refiere ni tampoco qué significa la opción “Datos opcionales”. Así, se ve en la Figura 4.9 cómo un 50% de los desarrolladores y un 42% de los administradores resaltaron esta característica para mejorar en la herramienta.

El tercer y último aspecto que varios usuarios indicaron debe revisarse para mejorar aún más la interacción, es la *cantidad de clics* para realizar algunas actividades o tareas dentro de QuAGI. Por ejemplo, cargar criterios de aceptación en las historias de usuario del backlog implica aproximadamente 5 clics desde que se inicia el proceso hasta que se logra el objetivo. En este caso, se observa en la Figura 4.9 que el 85% de los usuarios, en ambos roles, coincidieron en que el aspecto debe mejorarse para la próxima versión de QuAGI.

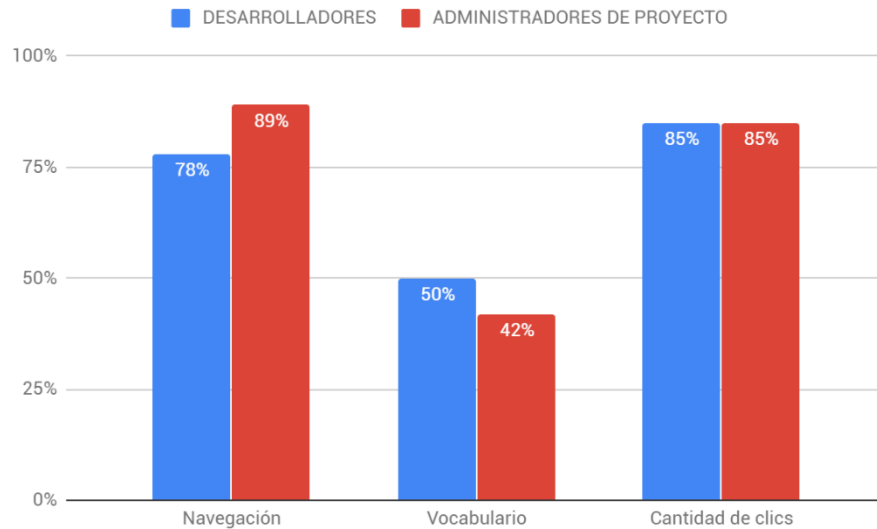


Figura 4.9 Aspectos negativos en la evaluación de usabilidad

Fuente: de elaboración propia

e) Conclusiones

Según proponen Runeson y Host en [83], el proceso de validación culmina con la presentación de informes que destacan ciertos aspectos de la experiencia. Debido a la característica de este estudio de caso y los que siguen, se reemplaza esta actividad con una más integral que, basada en informes estadísticos o cualitativos, presenta conclusiones que caracterizan a la experiencia.

Para el Estudio de Caso N° 1 se expondrán aquí, por un lado, conclusiones en base a informes cualitativos, haciendo referencia a aspectos que integren la relación entre los proyectos y el uso de QuAGI y, por otro lado, basadas en resultados cuantitativos a través de estadísticas que otorguen mayor claridad a los resultados cualitativos.

A partir de las reuniones post-implementación que se llevaron a cabo con cada uno de los Administradores de Proyecto, se obtuvieron conclusiones respecto al proceso de validación teniendo en cuenta los informes de Evaluación de Calidad del Proceso que ofreció QuAGI.

Tal como se detalló en el Capítulo 3 de esta tesis, QuAM establece una escala nominal que permite rotular el nivel de calidad obtenido en cada momento y asociado al componente que se evalúe. En la Figura 4.10 se muestran los resultados logrados por cada equipo participante del estudio en los informes de evaluación de calidad de todos los componentes. Como se ve, el Componente N° 3 “Comunicación con el Cliente” resulta ser el que mejor resultados obtuvo con un 50% de evaluaciones con nivel de calidad “*Muy Bueno*”. En cambio, el componente que obtuvo los peores resultados fue el Componente N° 1 “Capacidad de Producción de Entregables”, con un 30% de evaluaciones con nivel de calidad “*Regular*”.

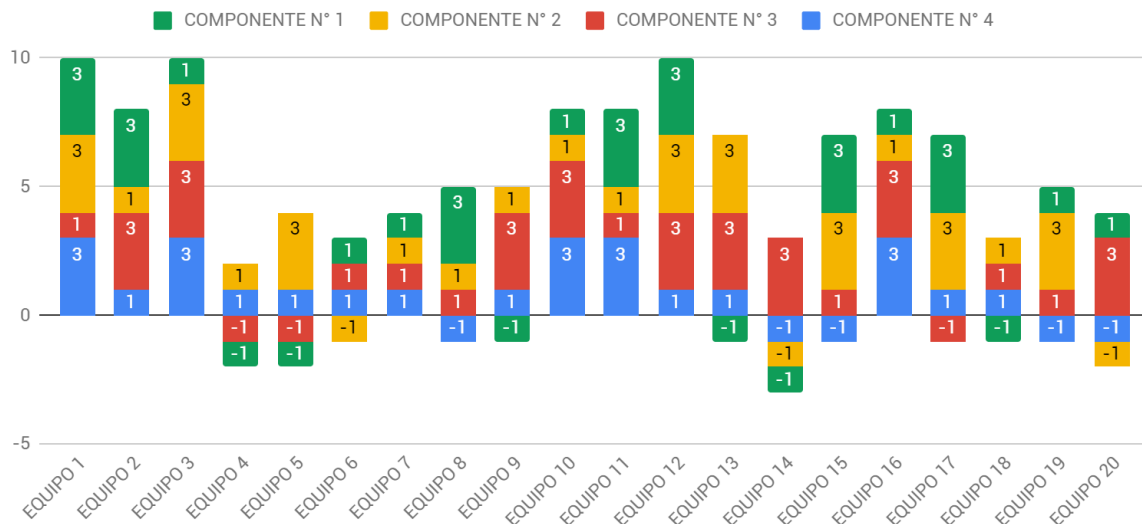


Figura 4.10 Resultados de Informe de Calidad por componentes de los 20 equipos participantes
Fuente: de elaboración propia

Estos resultados fueron analizados durante reuniones posteriores, algunas en forma presencial y otras a través de videollamadas. Durante dichos encuentros se ha registrado información proporcionada por cada uno de los administradores, de forma tal de poder realizar un análisis que permita validar la versión del framework AQF, incluyendo modelo y la plataforma web.

Finalmente, teniendo en cuenta la hipótesis planteada, se observa que la estructura de componentes definida por QuAM y automatizada a través de QuAGI ha resultado adecuada para evaluar la calidad del proceso de desarrollo asociado al proyecto ágil en cuestión. De hecho, al momento de la validación, como se observa en la Figura 4.11, utilizar QuAGI como herramienta de seguimiento había resultado de mucha ayuda para el 70% de quienes toman el rol de administración del proyecto, pues los informes de calidad habían permitido detectar desviaciones en el registro de requerimientos del cliente, identificar falencias en la distribución de los roles del equipo y completar información valiosa para el seguimiento del proyecto.

Cabe destacar que un 30% de los administradores manifestó que usar el framework resultó de poca o nula ayuda para mejorar el nivel de calidad del proyecto ágil pues la herramienta y los reportes simplemente ayudaron a que los equipos organizaran el trabajo de mejor manera sin ahondar en otras cuestiones tales como por ejemplo la comunicación dentro del equipo. Asimismo, manifestaron la necesidad de revisar el resultado final validando el uso de QuAGI durante todo el proceso de desarrollo que comprende el proyecto ágil estudiado.

Solo uno de los 20 equipos participantes del estudio manifestó, a través de su administrador, que basar la evaluación de calidad del proceso en QuAM soportado en la herramienta QuAGI no le fue de ayuda para tomar decisiones que mejoren la calidad del

proyecto. Esto debido fundamentalmente a aspectos relacionados con el equipo, entre las que se destacan:

- Falta de conocimiento en prácticas ágiles
- Poca experiencia en estimación de proyectos
- Problemas en el trabajo en equipo
- Dificultad del equipo para adaptarse a una nueva herramienta

¿Qué nivel de ayuda le significó AQF para tomar decisiones respecto al Proyecto Ágil que administra?

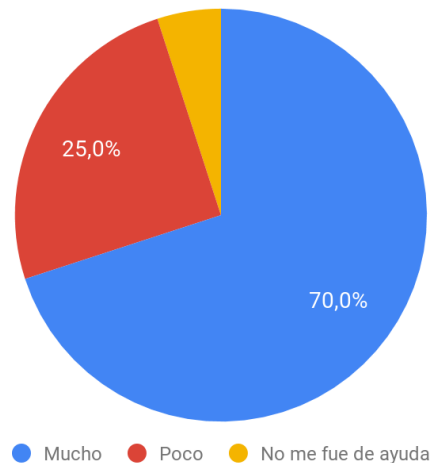


Figura 4.11 Utilidad de informes de Calidad de QuAGI en proyectos ágiles

Fuente: de elaboración propia

Como conclusión, resulta interesante destacar que los 20 equipos, a través de sus administradores, han manifestado que desearían utilizar la herramienta y, por tanto, incorporar AQF, para mejorar el trabajo en equipo cuando trabajan en proyectos ágiles. En concordancia a esto, han propuesto que se realicen experiencias en otros entornos a fin de validar el desempeño de los equipos utilizando QuAGI desde el inicio del proyecto y a lo largo de su ejecución.

A partir de esta experiencia, y notando la relevancia del equipo para el éxito de la implementación de QuAGI en entornos reales, se diseñó un nuevo estudio de caso que permita validar la herramienta teniendo en cuenta el proceso de evaluación de calidad en función a los equipos de trabajo en entornos ágiles añadiendo, además, nuevos informes que faciliten el seguimiento de los proyectos.

4.1.2 Estudio de Caso N° 2: Evaluación de calidad en procesos ágiles desde la perspectiva del trabajo en equipo

El Manifiesto Ágil [64] reconoce entre sus valores “...a los individuos y su interacción, por encima de los procesos y las herramientas...”. En base a ello, AQF incluye entre los componentes de QuAM a la evaluación de la calidad del proceso teniendo en cuenta el flujo

de comunicación entre los miembros del equipo a partir de la división de tareas y la transparencia del trabajo realizado en base a la definición de roles.

Debido a que durante la experiencia del estudio de caso N° 1 se observó la relevancia del trabajo en equipo en el proceso ágil y por tal motivo resulta necesario evaluar los niveles de calidad logrados, se diseña el *Estudio de Caso N° 2* que se enfoca, en el componente N° 2 para llevar a cabo el análisis de los equipos de trabajo como factor que impacta directamente en la calidad de procesos ágiles de desarrollo de software.

a) Diseño del Estudio

- i) **Objetivo del Estudio:** Validar la información que QuAGI proporciona respecto a la calidad lograda a nivel de equipo de trabajo en cada proyecto, y cómo esto impacta en la mejora de los procesos asociados.
- ii) **Hipótesis:** El uso de QuAGI, teniendo en cuenta lo definido por QuAM, favorecerá el trabajo en equipo en proyectos ágiles incrementando sus niveles de calidad durante el proceso.

b) Preparación de la experiencia

- i) **Población y Muestra:** La experiencia que aquí se describe contempla la participación de 20 equipos de proyectos de software guiados por algún enfoque ágil (SCRUM, XP, Lean Kanban, entre otros), localizados geográficamente en Chaco y Corrientes, los cuales constituirán la muestra que forma parte de este estudio de caso.

En total han formado parte del estudio aproximadamente 90 personas de acuerdo con la siguiente conformación de roles:

- 20 Product Owners
- 20 Responsables de equipos
- 50 Desarrolladores

- ii) **Planificación de la experiencia:** La metodología de evaluación implicó la implementación de QuAGI como herramienta de seguimiento de proyecto, para lo cual se solicitó a cada equipo que se registre en la plataforma antes de iniciar su uso. Luego desde el equipo técnico de QuAGI estableció permisos, indicando quién tendría el rol de Administrador del proyecto, que a su vez tenía la responsabilidad de dar de alta al cliente o Product Owner¹.

Una vez configurado el equipo, se continuó con la carga de información relevante al proyecto (Fecha de Inicio, Tipo de enfoque ágil, Nombre). Luego se solicitó se registren datos del sprint en ejecución, fundamentalmente, el

¹ Cabe destacar que de la experiencia participaron todos los integrantes del equipo, incluyendo al cliente o Product Owner. Para facilitar la lectura se utilizará el término cliente de aquí en adelante.

conjunto de historias de usuario y sus tareas correspondientes. En el caso de los equipos que debían trabajar sobre información cargada en otras herramientas de seguimiento, el equipo técnico de QuAGI realizó una importación de datos de todo el proyecto.

Para iniciar la primera iteración, se indicó al administrador que solicite obtener la calidad correspondiente al Componente N° 2, permitiendo que esa información quede registrada y pueda ser analizada a posteriori¹.

c) Recolección de evidencias

Con la información inicial registrada en QuAGI, los equipos de trabajo comenzaron la ejecución y uso de la herramienta a través del seguimiento y gestión de proyectos ágiles en un primer sprint. De forma preliminar, y teniendo en cuenta la metodología de trabajo actual, los equipos obtuvieron los niveles de calidad en el Componente N° 2 “Evaluación del Equipo de Trabajo”, registrando la información obtenida.

Luego, a medida que se avanzaba en el proyecto, los equipos incorporaron mejoras en base a recomendaciones que la herramienta proporciona, y nuevamente cada equipo solicitó la emisión de informes de evaluación de calidad.

Para cada caso se analizó el impacto del uso de QuAGI y la automatización de recomendaciones basadas en la configuración de QuAM, sobre el Componente N° 2 “Evaluación del Equipo de Trabajo”. Teniendo en cuenta el análisis de *20 equipos* trabajando en aproximadamente *300 historias de usuario* agrupadas en *30 sprints* y conformadas por casi *850 tareas*, se han recolectado evidencias que fueron analizadas una vez de finalizada la experiencia.

d) Análisis de los datos

El estudio de los resultados de validación en este estudio de caso se enfoca en el análisis del Componente N° 2 “Evaluación del equipo de trabajo”. Por ello, luego de la implementación y del uso de la herramienta QuAGI, se llevó adelante el análisis de los datos que se habían logrado recolectar.

En primer lugar, tal como se ve en la Figura 4.12, la mayoría de los proyectos participantes, aproximadamente un 60%, utilizan Scrum como framework de desarrollo ágil de software. Cerca de un 32% utiliza una metodología propia que combina prácticas ágiles de Scrum, Kanban Scrum, Lean, etc. Y finalmente, en menos de un 10%, se observan otros enfoques tales como XP o Lean.

¹ Respetando acuerdos de confidencialidad no es posible publicar información referente a cada empresa, ni a los proyectos considerados para este estudio.

Enfoques utilizados en los Proyectos de Software

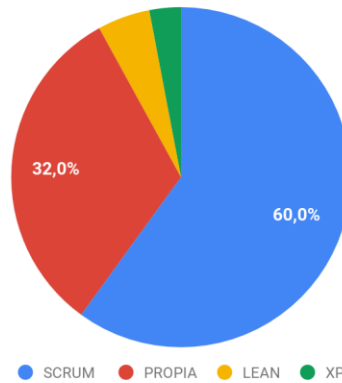


Figura 4.12 Distribución de enfoques entre los equipos participantes.
Fuente: De elaboración propia

Como se explicó anteriormente, los equipos trabajaron en sus proyectos y cargaron datos en QuAGI tal como lo hacían antes de utilizar AQF. Al consultar el nivel de calidad asociado al Componente de “Evaluación de Equipo de Trabajo”, como se aprecia en la Figura 4.13, el 75% de los equipos obtuvo una evaluación cercana a 1 que se corresponde con un nivel de calidad *Regular*.

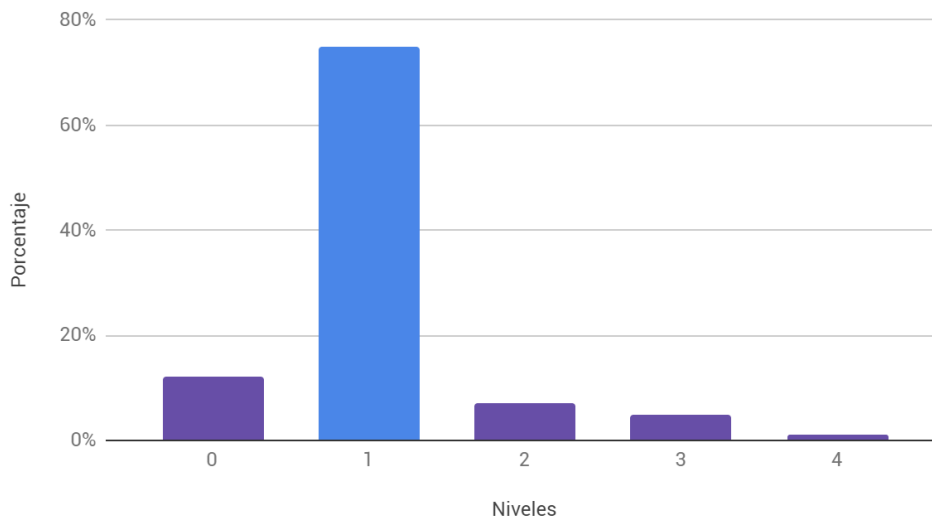


Figura 4.13 Niveles de calidad por equipo en el Componente N° 2. Primera ejecución.
Fuente: de elaboración propia

Como se destaca en la Figura 4.14, los niveles de calidad por equipo fueron resultado de la composición de los valores del atributo positivo y del atributo negativo para el Componente N° 2.

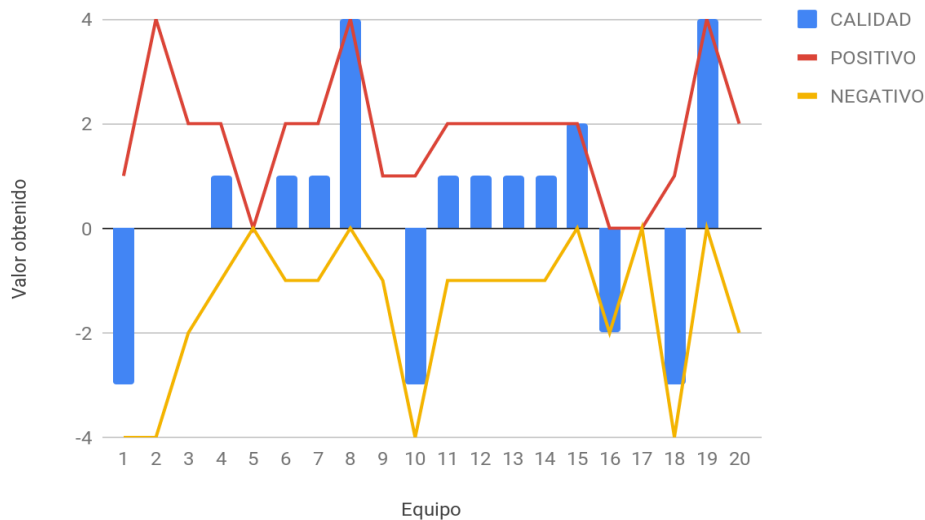


Figura 4.14 Distribución de valores en atributos negativo y positivo. Primera ejecución.
Fuente: de elaboración propia

Teniendo en cuenta lo que se observa en la Figura 4.15, en el caso del atributo positivo “Valorar la definición de roles”, se obtiene en promedio, de acuerdo con lo establecido por QuAM, un valor 2, lo que significa que en la mayoría de los proyectos analizados más del 80% de las tareas tienen definido responsable pero no se identifica qué tipo de tarea es, dificultando el seguimiento del proyecto. Para mejorar este aspecto, desde cada proyecto se ha podido acceder a informes, provistos por QuAGI, que han permitido identificar datos incompletos a nivel de tareas y a nivel de equipos.

Respecto al atributo negativo, se obtuvo en promedio un valor de -1 lo que, según QuAM, significa que si bien todas las historias de usuario pertenecen a algún sprint solo el 80% tienen sus tareas definidas. Esto dificultaría la priorización y la estimación para cada historia de usuario, lo que impacta directamente sobre la comunicación dentro del equipo de trabajo respecto al seguimiento del proyecto. Así, y si bien no forma parte del alcance de AQF resolver cómo definir tareas en cada historia de usuario, se recomendó a los equipos llevar a cabo los siguientes pasos: (1) realizar un primer esquema y análisis técnico del proceso; (2) determinar el “Definition of Done” (DoD), de aquí surgirá la necesidad de diversas tareas (por ejemplo revisiones de código, preparación de la demo, entre otras); (3) realizar un mapa de historias de usuario que permita visualizar vertical y horizontalmente el proyecto completo. De esta forma se ofreció una alternativa para organizar tareas y optimizar el valor del trabajo del equipo de desarrollo.

Luego de esta primera iteración, se generaron recomendaciones que fueron implementadas como mejoras en la segunda iteración del proceso de validación de AQF. Como se ve en la Figura 4.15, los niveles de calidad asociados a la “Evaluación de Equipos de Trabajo” mejoraron notoriamente completando datos en las tareas y en cada sprint backlog.

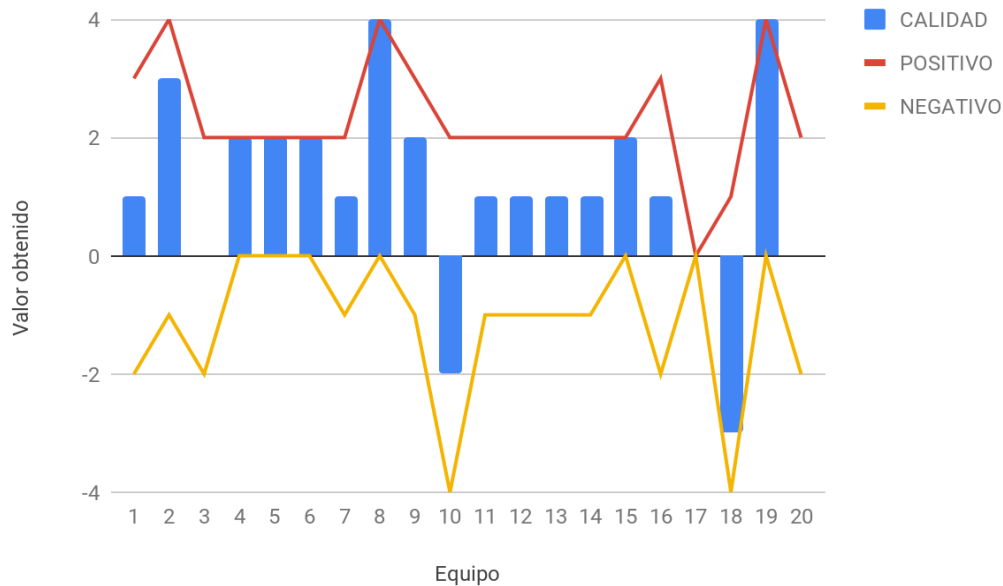


Figura 4.15 Valores de atributos positivo/negativo y niveles de calidad luego de la 2da ejecución
Fuente: de elaboración propia

Complementando los resultados cuantitativos obtenidos a partir del uso de QuAGI, se realizó una reunión con cada equipo para obtener feedback de aspectos cualitativos. Para ello, a través de reuniones se logró obtener más información respecto a cómo se modificó el trabajo dentro de los equipos para mejorar la evaluación de calidad de sus proyectos ágiles.

En consecuencia, varios de los Administradores de Proyecto han manifestado que tras la experiencia, la primera acción llevada adelante para mejorar el rendimiento fue establecer una clara definición de roles, distribuyendo las tareas entre cada miembro del equipo de acuerdo con su perfil. Esto contribuyó, incluso, a una mejor comunicación e interacción con el cliente, y brindó mayor claridad al seguimiento de cada proyecto.

Otra acción que han destacado es que fue posible identificar aquellas historias de usuario aisladas, que no formaban parte de ningún sprint, y que, por lo tanto, dificultaban la distribución de tareas impactando negativamente en el seguimiento del proyecto ágil. Después de identificarlas, se asignó cada una de ellas al sprint correspondiente, de acuerdo con su prioridad y a la necesidad que describían. En algunos casos la historia de usuario identificada fue incluida como tarea en la realización de otra historia y en otros casos simplemente fue eliminada. Algunos equipos han explicado que la posibilidad de observar, a través de QuAGI, la composición general de los proyectos en función a sprints, historias de usuario y tareas, permitió aplicar la práctica ágil de elaborar un Story Map, favoreciendo una visión integral y más ordenada del proyecto.

e) Conclusiones

A través de este estudio de caso que permitió evaluar el impacto del uso de QuAGI sobre el equipo de trabajo, se ha observado, por un lado, que las empresas han incorporado fácilmente a QuAGI como herramienta de soporte al proceso de desarrollo y gestión de sus proyectos ágiles. Esto debido a, según han expresado las mismas empresas en las reuniones posteriores a los resultados obtenidos, la plataforma proporciona a sus usuarios una interfaz amigable y con flujos de trabajo claramente definidos que alivianan los procesos asociados a la gestión.

Y, por otro lado, respecto a los resultados obtenidos a través de la evaluación de calidad del *Componente N° 2 “Evaluación de Equipos de Trabajo”*, las empresas han manifestado que reflejan en gran medida lo esperado de acuerdo con las características particulares de cada uno de sus equipos y se pudo concluir que, resulta necesario establecer nuevas estrategias para mejorar la interacción de sus miembros y por ende su productividad. El factor más crítico, de acuerdo con lo que manifiestan los participantes, está relacionado a la falta de claridad y organización en la distribución de tareas, según los perfiles que participan en cada equipo. Esta distribución, se puede ver claramente, a través de los informes de calidad presentados en QuAGI.

Verificándose la hipótesis inicial del estudio, el uso de esta herramienta ha facilitado el trabajo de cada equipo colaborado, fuertemente en la identificación de aquellos aspectos que afectaban negativamente la calidad del proceso ágil.

Al igual que en el estudio de caso N° 1, durante esta experiencia se lograron obtener datos que contribuyeron al análisis de QuAGI desde la perspectiva de la Experiencia de Usuario (UX), a través de diálogos con los usuarios y teniendo en cuenta las modificaciones realizadas en la interfaz producto del análisis anterior¹.

De esta forma, los equipos coincidieron en su mayoría que entre los factores que aportan beneficios a la UX se destacan:

- *Fácil de Usar*: Pues la versión de QuAGI, en este estudio de caso, contó con una interfaz intuitiva que facilitó el proceso de carga de información y modificaciones posteriores. El uso de la herramienta no resulta engorroso, con pocos clics se puede realizar todas las tareas necesarias simplificando su navegación.
- *Aprendizaje rápido*: Sumado a lo anterior la amigabilidad de la herramienta permite agilizar su uso, y lograr una curva rápida para aprender a trabajar con ella.
- *Feedback valioso para el equipo*: Debido a las funcionalidades que determinan el trabajo ordenado y organizado en equipo, QuAGI representa una herramienta que aporta valor especialmente por el feedback constante que ofrece durante su uso.

¹ Para este estudio de caso se mejoró la navegabilidad de QuAGI, revisando mejoras en la interfaz para disminuir la cantidad de pasos necesarios en cada función. Además, se añadieron mensajes de ayuda para el uso de las funciones de QuAGI.

Además, si bien se mejoraron algunas cuestiones a partir de la experiencia del estudio de caso N° 1, durante esta experiencia los equipos han identificado nuevos factores que deberán revisarse. Entre estos se destacan:

- *Necesidad de usar otras herramientas:* Por el momento los informes que reporta QuAGI se limitan a resultados de la evaluación de calidad del proyecto ágil, para otros informes los equipos debieron recurrir a herramientas adicionales. Es necesario incorporar reportes que aporten a la gestión del proyecto y reduzcan la incertidumbre respecto a los resultados obtenidos.
- *Soporte Técnico:* Si bien la herramienta presenta una interfaz, los equipos han manifestado que existen cuestiones procedimentales que han requerido de soporte y asesoramiento para llevarse a cabo.

Luego de esta experiencia y gracias a la recolección de evidencias relacionadas tanto a la herramienta en uso, como al análisis de la interacción con el usuario, se definió un nuevo estudio de caso, en función a todos los componentes establecidos en QuAM e incluyendo las mejoras a nivel de interfaz sobre QuAGI.

Finalmente, los equipos participantes, en este estudio de caso, propusieron evaluar el funcionamiento de QuAGI comparando su uso frente a otras herramientas de gestión de proyectos ágiles. La propuesta de los equipos se basó en la necesidad de contar con un informe que destaque los beneficios de contar con QuAGI como software de gestión de proyectos ágiles. Por ello, a continuación, se describe otro estudio de caso, cuya experiencia tiene por objetivo comparar el resultado de evaluación de calidad de procesos ágiles en proyectos cuyo seguimiento se realice usando QuAGI versus el uso de otras herramientas.

4.1.3 Estudio de Caso N° 3: Experiencia de validación de QuAGI como herramienta de gestión de proyectos ágiles

Este estudio de caso contó con la participación de equipos de estudiantes en la cohorte 2019 del Informatario, el Laboratorio Informático de la provincia del Chaco.

Antes de describir el estudio, resulta necesario hacer referencia al ámbito en el cual se llevó a cabo la experiencia:

El Informatario surge a partir del trabajo en el año 2009 durante el Foro de Competitividad de la Industria Software y Servicios Informáticos (SSI) de la Provincia del Chaco. En ese momento se detectó como uno de los principales problemas la falta de capacitaciones específicas orientadas a formar técnicos en programación aptos para trabajar en las empresas del medio. Como respuesta a esta problemática, el Ministerio de Economía, Industria y Empleo junto con la UTN-Facultad Regional Resistencia (UTN-FRRe), el Polo Tecnológico Chaco y la Empresa Globant montaron en 2012 un Laboratorio de Informática (Informatario) [84].

Desde el año 2012 y hasta la actualidad, como parte del Informatario, se lleva adelante el *Curso de Desarrollo Avanzado en Aplicaciones Informáticas*. El mismo tiene como destinatarios a personas entre 18 y 23 años (no excluyente), con o sin estudios secundarios

completos (máximo 2 materias adeudadas). El curso exige una demanda de trabajo de 1200 horas reloj de manera presencial, distribuidas entre 30 y 40 horas semanales durante 10 meses (de abril a diciembre), en el que se articulan contenidos entre 5 ejes temáticos (o módulos): Fundamentos de la programación, Ingeniería de Software, Gestión de la Información, Programación avanzada (Tecnologías .NET., Tecnologías Java. y/o Tecnologías PHP) e inglés.

Durante el año 2019 formaron parte de la octava edición del curso aproximadamente 30 estudiantes, quienes debieron afrontar el desafío, durante el primer semestre, de la gestión de proyectos ágiles simples, que duraron en total 2 sprints. A partir de esta breve presentación, se describe a continuación el Estudio de Caso N° 3.

a) Diseño del Estudio

i) *Objetivo del Estudio:*

- 1) Validar la gestión de calidad de proyectos ágiles utilizando el esquema propuesto por AQF, incluyendo la estructura definida por QuAM y su automatización en QuAGI.
- 2) Comparar los resultados entre equipos, utilizando QuAGI frente a otras herramientas similares.

ii) ***Hipótesis:*** El uso de QuAGI, como herramienta de seguimiento de proyectos, mejora la calidad de procesos ágiles de desarrollo de software frente al uso de otras herramientas.

b) Preparación de la experiencia

i) ***Población y Muestra:*** El estudio del caso abarca la participación de 30 personas, distribuidas en 6 equipos trabajando en diferentes proyectos ágiles de desarrollo web en el Curso Avanzado de Desarrollo de Aplicaciones Informáticas del Informatorio, cohorte 2019. En este caso dado que la *Población* está representada por un número bajo de personas, se realizará el estudio sobre el 100% de la población sin ser necesario la selección de una muestra representativa.

ii) ***Planificación de la experiencia:*** La experiencia implica la comparación del funcionamiento y productividad en la gestión de la calidad de proyectos ágiles utilizando QuAGI por 3 equipos y Taiga¹ por el resto de los equipos. Cabe destacar que los equipos participantes no habían tenido experiencia previa en el uso de otras herramientas para el seguimiento de proyectos de software guiados por prácticas ágiles.

Todos los equipos debieron iniciar con la carga de información de cada uno de sus proyectos, y del Product Owner que les correspondía de acuerdo con el escenario

¹ Se eligió Taiga entre 4 opciones de herramientas para gestión de proyectos ágiles, pues es de fácil uso, acceso libre, cuenta con la opción de kanban board, organización de sprints y además permite la emisión de diversos reportes. Para más información se puede consultar en <http://taiga.io>

asignado. El t3pico elegido para el desarrollo del producto web fue la obtenci3n de una Landing Page¹ de presentaci3n para una ONG.

La duraci3n total de los proyectos abarc3 la ejecuci3n de 2 sprints, de 10 d3as cada uno. El ciclo de desarrollo de cada uno de los productos estuvo guiado por pr3cticas 3giles correspondientes a Scrum. En cada sprint se establecieron fechas de control a modo de puntos de encuentro que permitieron monitorear el avance de los proyectos, la evaluaci3n de calidad del proceso 3gil y el uso de las herramientas.

Los equipos solo contaban con el conocimiento referido a cada pr3ctica 3gil, pero no hab3an sido capacitados en el uso de software para su gesti3n. Por lo cual, cada equipo debi3 aprender a trabajar de forma aut3noma con cada herramienta.

c) Recolecci3n de evidencias

Habi3ndose definido la primera versi3n del product backlog de cada proyecto 3gil, se procedi3 a la organizaci3n de sprints en cada una de las plataformas por equipo.

La experiencia cont3 con el an3lisis de *6 proyectos 3giles*, que implicaron la evaluaci3n de *12 sprints* con aproximadamente *95 historias de usuario* con un total de *31 participantes* (incluyendo el Product Owner).

Como ocurri3 con los estudios de casos explicados anteriormente, en este escenario tambi3n se realiz3 la recolecci3n de evidencias durante el tiempo de uso de las herramientas QuAGI y Taiga. Pero a diferencia de los primeros casos, en el Estudio de Caso N3 3 la evaluaci3n se realiz3 de principio a fin del proyecto 3gil. La experiencia implic3 el an3lisis de 2 grupos, por un lado, los equipos utilizando QuAGI y por otro los equipos que usaban Taiga.

Para la recolecci3n de evidencias, no solo se tuvo en cuenta lo registrado en cada herramienta, sino que adem3s se elabor3, por un lado, un formulario para plasmar los resultados parciales y finales y, por otro lado, una encuesta que fue completada por los participantes al terminar la experiencia. Finalmente se realiz3 una entrevista con el Product Owner pues su experiencia resultar3a de vital importancia para la obtenci3n de conclusiones respecto a la utilidad de QuAGI en la calidad de procesos 3giles de acuerdo con lo establecido por QuAM.

A continuaci3n, se exponen los resultados obtenidos y el an3lisis correspondiente.

d) An3lisis de los datos

La validaci3n en este estudio comprendi3, por un lado, la observaci3n y an3lisis de ambos grupos con las herramientas de gesti3n correspondientes. Y, por otro lado, se enfoc3 en el estudio del uso de QuAGI de forma completa, incluyendo el funcionamiento del an3lisis para los 4 componentes definidos en QuAM.

¹ Una landing page, es una p3gina web dise3ada espec3ficamente para convertir visitantes en leads. Un lead es un usuario que ha descargado uno de los contenidos de la web y, a cambio, ha dejado sus datos de contacto.

Para el caso de los equipos que han trabajado con QuAGI, se analizaron, en primer lugar, los resultados de evaluación de calidad de todos los componentes establecidos en QuAM, ya descritos en el Capítulo 3 de esta tesis. Y, en segundo lugar, se realizó además el Test de Usabilidad a partir de la interacción de los miembros de cada uno de los equipos participantes, al igual que se hizo en los estudios de casos antes descritos.

Para el caso de los equipos que han trabajado con Taiga, y debido a que esta herramienta no ofrece evaluación de calidad, se realizó el análisis de resultados en base a los mismos componentes que define QuAM.

Antes de iniciar con la experiencia, se realizó una encuesta para conocer el grado de conocimiento y experiencia de los participantes respecto a proyectos ágiles de software. Tal como se observa en la Figura 4.16, la mayoría de los usuarios, exactamente más del 45%, no cuenta con experiencia práctica en el uso de herramientas para gestión de proyectos de software. Idénticos resultados se obtienen al consultar sobre la gestión de proyectos ágiles, observándose que casi el 62% de los participantes no ha tenido nada de experiencia en esto.

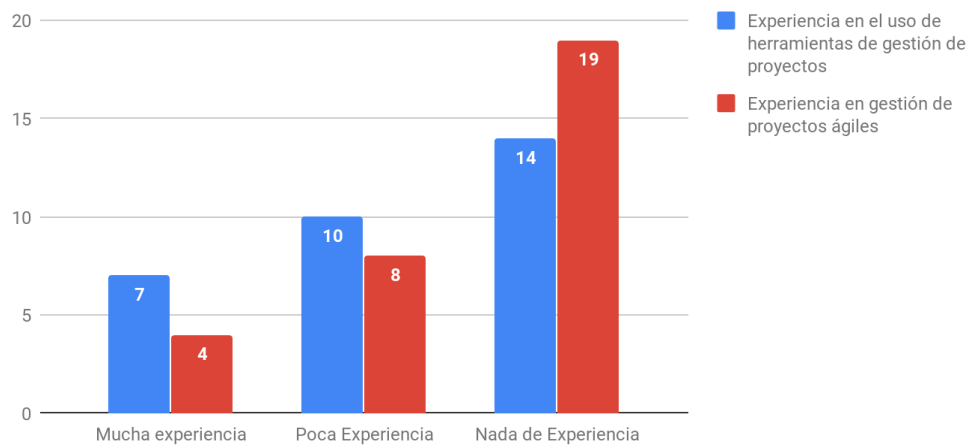


Figura 4.16 Resultados de Encuesta inicial a participantes sobre su experiencia
Fuente: de elaboración propia

Como se explicó anteriormente, los 6 proyectos ágiles a evaluar estuvieron compuestos por 2 sprints de 10 días cada uno. A fin de realizar seguimiento continuo a la ejecución de cada caso, durante la experiencia se establecieron dos puntos de control para verificar los niveles de Calidad de cada componente: un punto de control luego de transcurridos los primeros cinco días de iniciado el proyecto, y un segundo control al finalizar el sprint. Y cabe destacar que la información correspondiente a cada punto de control fue obtenida a partir de encuestas que respondieron los administradores de proyecto y cuya estructura se incluye en el *Anexo F* de esta tesis.

e.1 Análisis de datos: Primer control

Para los equipos que utilizaron QuAGI, el primer punto de control implicó la observación de los niveles de calidad logrados por cada proyecto junto a la utilidad de los informes de seguimiento que ofrece QuAGI en esta experiencia. Se presenta en la Figura 4.17 los resultados de niveles de calidad obtenidos en cada proyecto, observándose que predomina el nivel MALO, seguido del nivel BUENO en algunos casos y con pocas evaluaciones del tipo nivel MUY MALO.

EQUIPO	COMPONENTE 1 CAPACIDAD DE PRODUCCIÓN DE ENTREGABLES	COMPONENTE 2 EVALUACIÓN DEL EQUIPO DE TRABAJO	COMPONENTE 3 COMUNICACIÓN CON EL CLIENTE	COMPONENTE 4 GESTIÓN DE REQUERIMIENTOS
Equipo 1	MUY MALO	MALO	MUY MALO	BUENO
Equipo 2	BUENO	MALO	MALO	BUENO
Equipo 3	MALO	MALO	MALO	MALO

Figura 4.17 Niveles de calidad obtenidos por los equipos del grupo QuAGI al inicio de la experiencia
Fuente: de elaboración propia

A partir de estos resultados, los administradores y equipos, expresaron en qué medida están de acuerdo con la evaluación arrojada por QuAGI respecto a cada componente, y contrastándolo con su propia visión en cuanto a ello.

Tal como muestra la Figura 4.18, el 100% de los administradores está *Muy de acuerdo* con la evaluación de calidad correspondiente al componente 1. En el caso del componente 2, casi el 70% de los administradores considera estar *De acuerdo* con la evaluación, y en el mismo porcentaje, expresan estar *Muy de acuerdo* con la evaluación arrojada para el componente 3. Por último, el 100% considera estar *De acuerdo* con la evaluación correspondiente al componente 4. Por lo tanto, resulta destacable que, en todos los casos, se observa gran porcentaje de aceptación respecto a la evaluación informada por QuAGI.

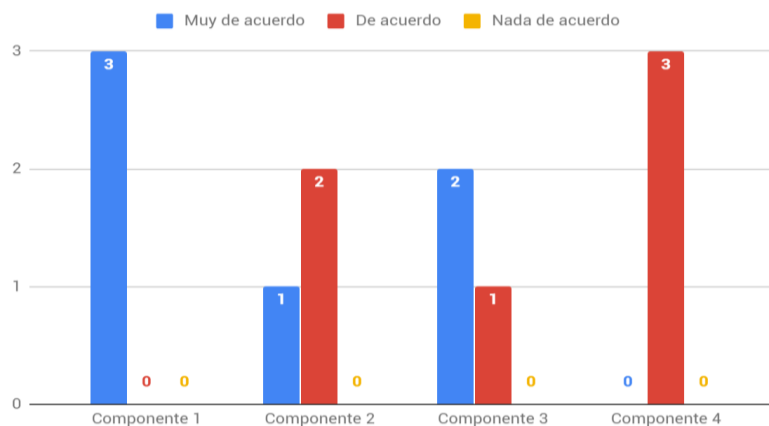


Figura 4.18 Evaluación de los administradores de proyecto respecto a lo informado por QuAGI
Fuente: de elaboración propia

El análisis de los resultados presentados en la Figura 4.18 se complementa con la identificación de los factores que han causado estos valores a partir de lo que manifiestan los administradores de cada proyecto y que se describen a continuación.

En primer lugar, para el caso del componente *Capacidad de Producción de Entregables* la mayoría de los equipos asignan puntos de historia a un grupo de historias de usuario que representan entre un 40% y un 80%, pero que en ningún caso llegan a 100%. Además, se verifica que, en todos los equipos, las historias de usuario finalizan, pero sus criterios de aceptación no se validan con el product owner.

En el segundo lugar, para el componente *Evaluación del Equipo de trabajo* se ha podido detectar que, si bien los equipos definen las historias de usuario menos de un 10% de ellos define tareas en cada caso y asigna responsables de su ejecución. Al mismo tiempo, muchas veces el product backlog acumula historias de usuario aisladas, pues son creadas, pero no se asignan a ningún sprint, ocasionando confusiones en el ciclo de desarrollo.

Los peores niveles de calidad se obtuvieron para el componente *Comunicación con el Cliente*, pues en todos los casos se observa que, si bien el cliente o Product Owner forma parte del equipo, no ocurren ingresos a la plataforma ni participación en el proyecto. Esto ocurre por dos razones principales: la primera tiene que ver con que, en esta experiencia los equipos dependen de un intermediario que los conecta con cada cliente, y se ha observado que en todos los casos han olvidado enviar datos de credenciales para el acceso a la plataforma. Y la segunda razón se relaciona a que, mayoritariamente, los perfiles de clientes involucrados en este estudio¹, en su mayoría, cuentan con poca experiencia en tecnología y que no consideran importante formar parte del proceso de obtención del producto de software.

Por último, en el caso del componente *Gestión de los Requerimientos* se han observado los mejores niveles de calidad. Casi un 70% ha obtenido el valor BUENO, debido a que 2 de los 3 equipos al redactar las historias de usuario, si bien incluyen criterios de aceptación, no utilizan el template recomendado ni las prioriza. Pero por otro lado ninguna de las historias de usuario presenta dependencias en un mismo sprint, lo que atenúa el impacto del atributo negativo. El equipo restante ha obtenido un valor MALO pues si bien sus historias de usuario han sido redactadas en función al template recomendado e incluyen criterios de aceptación, la mayoría de ellas no ha sido priorizada y, enfatizando el atributo negativo, se identificaron dependencias dentro del sprint.

En función a estos resultados los equipos consultaron los informes online complementarios que QuAGI les proporciona de acuerdo con el avance de sus proyectos y los que, tal como se explicó en el Capítulo 3 de esta tesis, permiten acceder a información útil para definir estrategias que permitan mejorar los niveles de calidad del proyecto ágil. Cabe destacar que esta característica fue añadida en la herramienta web

¹ En este caso los 6 Product Owners representan a fundadores de ONGs

luego del análisis de los resultados obtenidos al finalizar el estudio de caso N° 2 presentado anteriormente en esta sección.

Así, haciendo uso del *Informe de Avance* se obtuvo que, tal y como se resume en la Figura 4.19, habiendo transcurrido más de la mitad del Sprint, ningún equipo lograba completar siquiera el 25% de lo previsto. Ayudados con este informe, cada equipo pudo identificar cuáles eran las historias de usuario que tenían retrasos en su desarrollo y el porcentaje de avance del sprint.

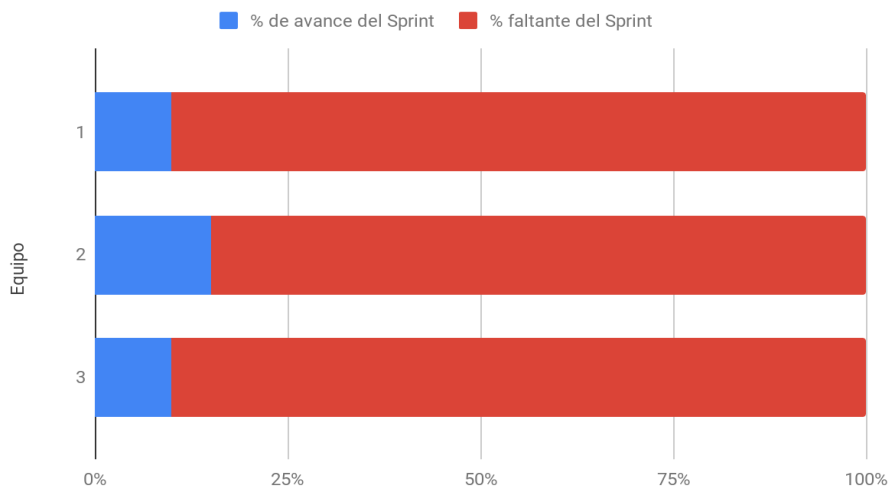


Figura 4.19 Resultados obtenidos por cada equipo respecto al *Informe de Avance del Sprint*
Fuente: de elaboración propia

Además, los equipos pudieron detectar las historias de usuario que no habían sido priorizadas, de forma tal de poder actualizarlas y favorecer la producción de entregables por sprint. La Figura 4.20 compara los resultados obtenidos respecto a porcentaje de historias de usuario priorizadas y sin priorizar, siendo este último el caso que predomina.

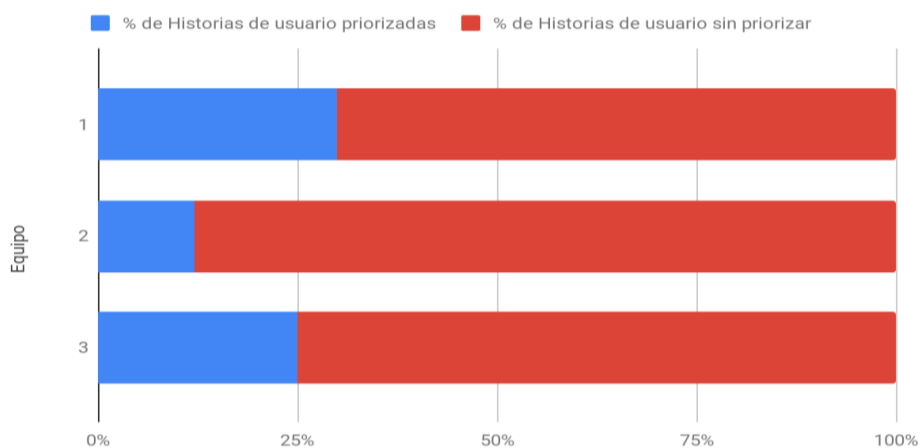


Figura 4.20 Resultados obtenidos por cada equipo respecto al informe de *Priorización*
Fuente: de elaboración propia

Luego, los equipos se abocaron a realizar los ajustes necesarios para lograr mejorar los niveles de calidad en cada componente. Para ello optaron por diversas estrategias, por ejemplo:

- Corregir cuestiones que mejoren los valores por componente, teniendo en cuenta los valores arrojados por QuAGI en la evaluación y en los informes de calidad.
- Completar información relacionada a las Historias de Usuario para mejorar la organización de los sprints, tales como criterios de aceptación, descomposición en tareas, estimación y priorización.
- Ajustar todos los detalles identificados por QuAGI que mejorarían los niveles de calidad del proceso.

En el caso de los equipos que trabajaron con Taiga, las encuestas realizadas a los administradores de los proyectos arrojaron los siguientes resultados.

En primer lugar, de los 3 equipos, solo en uno de ellos sus integrantes comprendían más del 80% de las historias de usuario especificadas, el resto lograron comprender requerimientos en menos de un 20%. En principio, para el primer sprint se identifican 13 historias de usuario para el Equipo 1, 15 para el Equipo 2, y 12 para el Equipo 3.

Como se aprecia en la Figura 4.21, respecto a la gestión de requerimientos en los 3 equipos la cantidad de historias de usuario completas¹ fueron muy pocas en contraposición a las historias de usuario incompletas. Pues en todos los equipos, estas últimas representaron, en promedio, un 85% del total de historias de usuario.

En cuanto a historias de usuario estimadas también se observa muy poco porcentaje, de hecho, solo 1 equipo ha logrado estimar al menos el 50% de las historias de usuario. Finalmente, la priorización tampoco logró buenos resultados, como se ve en la Figura 4.20 los números de historias de usuario priorizadas no llegan, en promedio, al 10% del total de historias de usuario por equipo.

¹ En esta tesis son catalogadas como *completas* las historias de usuario que fueron redactadas de acuerdo al template recomendado por la comunidad (Como [rol] quiero [función] para [objetivo]), además incluye tareas y criterios de aceptación.

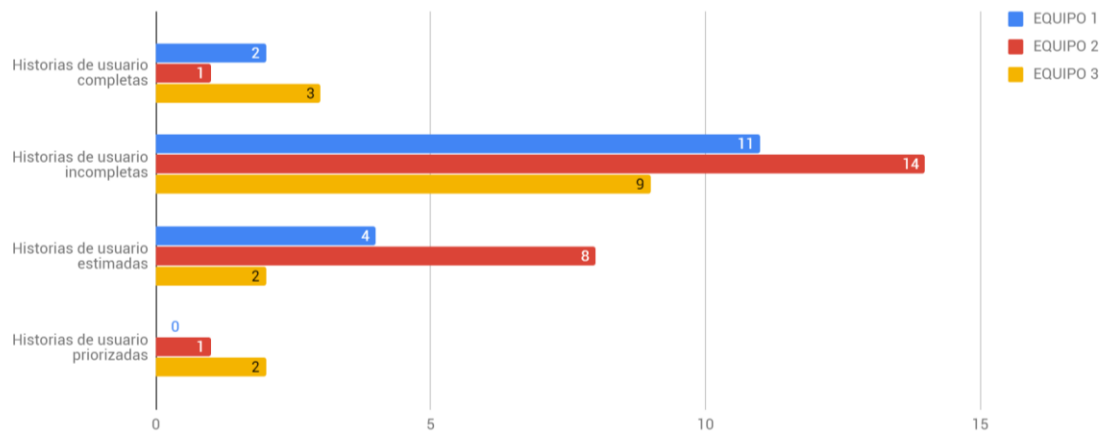


Figura 4.21 Producción de historias de usuario por equipos en Taiga
Fuente: de elaboración propia

Asimismo, los 3 equipos manifestaron, a través de sus administradores de proyecto, que presentaban dependencias de historias de usuario dentro del sprint, por lo que debieron ser bloqueadas en Taiga. Sin embargo, luego de bloquearlas resultó difícil identificar cuáles eran las historias de usuario dependientes entre sí, dificultando la planificación de cada proyecto.

En la Figura 4.22 se incluye el porcentaje de tareas definidas pero que no tienen responsable asignado al momento del control; esta información fue enviada por los administradores de proyecto al consultar con sus equipos.

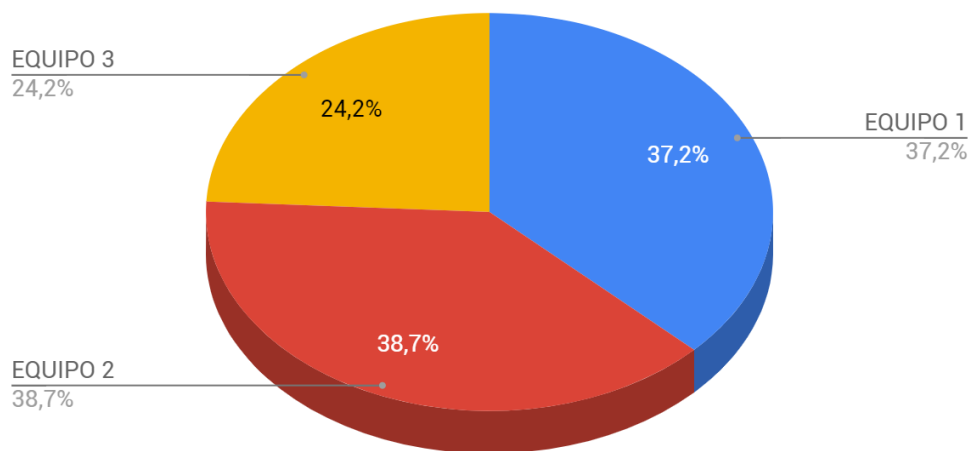


Figura 4.22 Miembros del equipo con tareas asignadas sin finalizar
Fuente: de elaboración propia

Por último, en todos los proyectos se destaca la participación del Product Owner tanto en las reuniones del tipo sprint planning como al momento de definir historias de usuario. Sin embargo, esta evidencia no puede ser visualizada desde la herramienta pues los equipos no consideraron estrategias para su seguimiento, tales como comentarios en la definición de historias de usuario o novedades en la sección “issues” que ofrece Taiga.

Cabe destacar que los administradores indicaron que fue muy difícil dar respuesta a la encuesta solicitada dado que Taiga no proporciona dicha información de forma directa, y en la mayoría de los casos debió recurrirse al equipo o a control manual de lo realizado hasta el momento.

De igual manera, los administradores se basaron en estas evidencias para detectar factores que pudieran alterar la calidad del proceso ágil, como por ejemplo la definición incompleta de historias de usuario que dificultan la comprensibilidad del proyecto, la dependencia entre historias de usuario que obstaculiza la planificación, la falta de seguimiento de la participación del Product Owner, entre otros. Por ello, los equipos se abocaron a continuar en cada proyecto teniendo en cuenta los siguientes ajustes:

- Control de historias de usuario y redefinición en caso de ser necesario
- Estimación y priorización del 100% de historias de usuario planificadas
- Monitorización del trabajo conjunto con el Product Owner
- Seguimiento en la asignación de tareas a miembros de cada equipo

Debido a que Taiga no proporciona esta información, los equipos dispusieron de planillas de cálculo compartidas a fin de complementar el seguimiento utilizando herramientas adicionales como Google Drive para documentos y Github para el control de producción en el equipo. Luego del primer punto de control, los equipos continuaron con sus respectivos ciclos de desarrollo y ajustando detalles correspondientes a cada proyecto.

e.2 Análisis de datos: Segundo control

Al finalizar el sprint se retomaron las actividades de recolección de datos desde el funcionamiento de cada equipo a través de la misma encuesta utilizada durante el primer control.

En primer lugar, para el caso de los equipos que utilizaron QuAGI se realizó el monitoreo, nuevamente, de los niveles de calidad por componente y, como se observa en la Figura 4.23, todos los valores han mejorado considerablemente, sobre todo en el Componente N° 3 “Comunicación con el cliente”.

EQUIPO	COMPONENTE 1 CAPACIDAD DE PRODUCCIÓN DE ENTREGABLES		COMPONENTE 2 EVALUACIÓN DEL EQUIPO DE TRABAJO		COMPONENTE 3 COMUNICACIÓN CON EL CLIENTE		COMPONENTE 4 GESTIÓN DE REQUERIMIENTOS	
	1° PUNTO DE CONTROL	2° PUNTO DE CONTROL	1° PUNTO DE CONTROL	2° PUNTO DE CONTROL	1° PUNTO DE CONTROL	2° PUNTO DE CONTROL	1° PUNTO DE CONTROL	2° PUNTO DE CONTROL
Equipo 1	MUY MALO	BUENO	MALO	MUY BUENO	MUY MALO	MUY BUENO	BUENO	EXCELENTE
Equipo 2	BUENO	MUY BUENO	MALO	BUENO	MALO	MUY BUENO	BUENO	EXCELENTE
Equipo 3	MALO	BUENO	MALO	EXCELENTE	MALO	EXCELENTE	MALO	EXCELENTE

Figura 4.23 Evolución de los niveles de calidad en equipos usando QuAGI

Fuente: de elaboración propia

Iniciando el análisis con el Componente N° 1 “*Capacidad de Producción de Entregables*”, en la Figura 4.24 se resumen los datos que muestran los porcentajes de avance respecto a estimación de historias de usuario y validación del sprint. Como se observa, en la mayoría de los equipos, al finalizar el primer sprint, lograron altos porcentajes de cumplimiento tanto para la estimación de historias de usuario como para la validación de los criterios de aceptación al finalizar historias de usuario. En promedio se destaca aproximadamente un 90% de cumplimiento, en promedio, tanto para estimación como para sprints validados con el cliente.

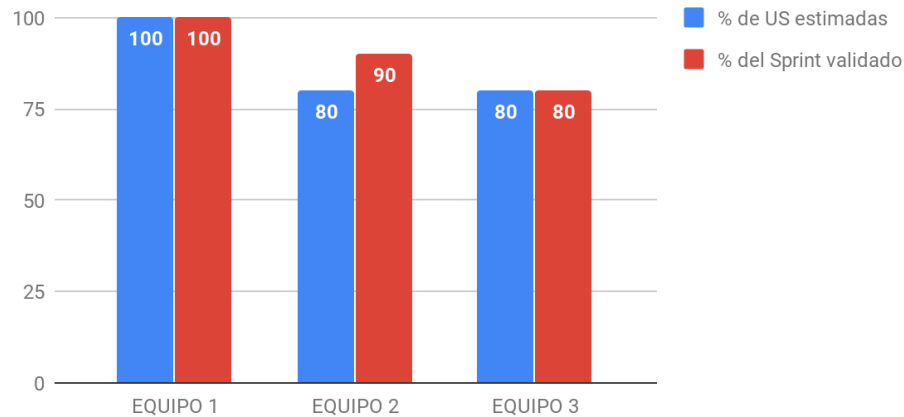


Figura 4.24 Análisis de atributos del Componente N° 1 de QuAM al usar QuAGI
Fuente: de elaboración propia

En el caso del componente N° 2 “*Evaluación del Equipo de Trabajo*”, y como se destaca en la Figura 4.25, se evidencia un alto porcentaje en la definición de tareas por historia de usuario junto a la asignación de responsables al definir el tipo de tarea (UI, Front, Back o Testing). Respecto a la descomposición de historias de usuario, se observa que la mayoría de los equipos aún no logran completar el 100%, por lo que debe mejorarse. Y, finalmente, se ha logrado que los equipos eliminen historias de usuario aisladas, es decir que no puedan agruparse en ninguno de los 2 sprints, facilitando la organización y distribución de trabajo.

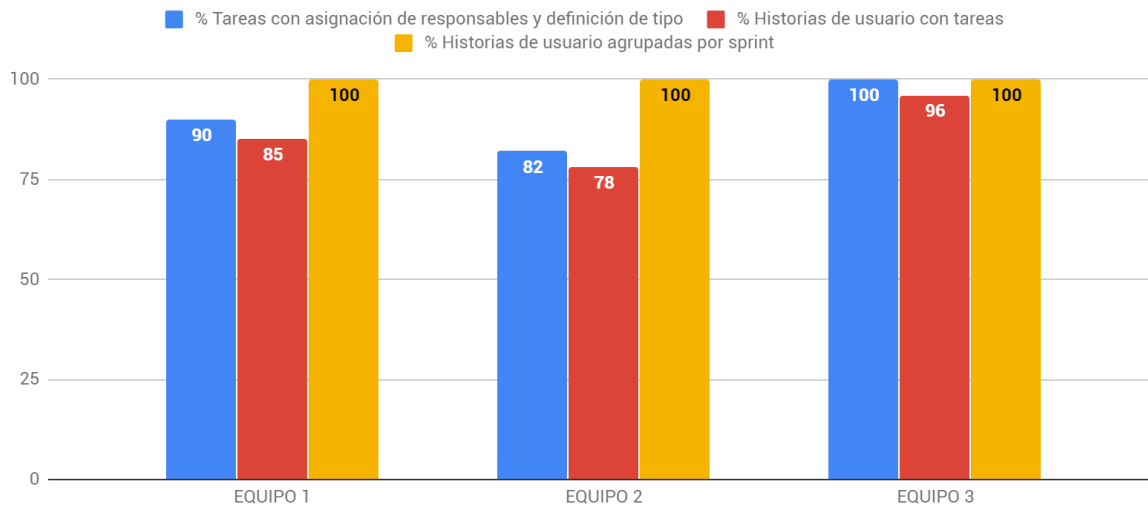


Figura 4.25 Análisis de atributos del Componente N° 2 de QuAM al usar QuAGI
Fuente: de elaboración propia

A continuación, en la Figura 4.26 se presentan los resultados al analizar los atributos correspondientes al componente N° 3 “*Comunicación con el cliente*”, como se puede apreciar en todos los casos los valores se acercan al 100% luego del ajuste que cada equipo realizó sobre sus proyectos desde el primer control. Según se observa en la Figura 4.26 se destaca un alto porcentaje en la participación del Product Owner en todo el proyecto, incluyendo el uso de QuAGI como plataforma de seguimiento.

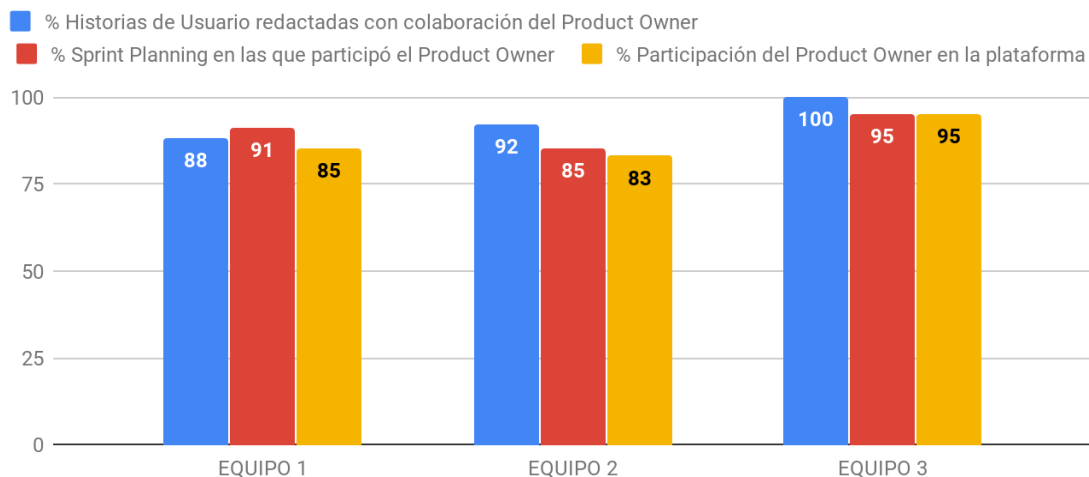


Figura 4.26 Análisis de atributos del Componente N° 3 de QuAM al usar QuAGI
Fuente: de elaboración propia

Finalmente, ha sido en el componente N° 4 “*Gestión de Requerimientos*” donde se evidencia, de acuerdo con lo presentado anteriormente en la Figura 4.23, los mejores índices de niveles de calidad obtenidos. Para mayor claridad se expone en la Figura 4.27 el análisis de los indicadores necesarios para medir los atributos que forman parte del componente N° 4. Como se puede observar, en su mayoría los valores logran el 100%

de acuerdo con lo que se ha logrado recolectar a través del funcionamiento del proyecto en QuAGI.

En primer lugar, todos los equipos indican que sus historias de usuario han sido redactadas en función al template recomendado por la comunidad. En segundo lugar, se observa que, por equipo, en promedio casi el 98% de las historias de usuario se completaron incluyendo criterios de aceptación para facilitar su validación con el Product Owner. Y, además casi un 95% de historias de usuario en cada proyecto, han sido priorizadas utilizando la técnica de MOSCOW provista por QuAGI. Por último, pero no menos importante, en ninguno de los casos se observan dependencias entre historias de usuario dentro de un mismo sprint.

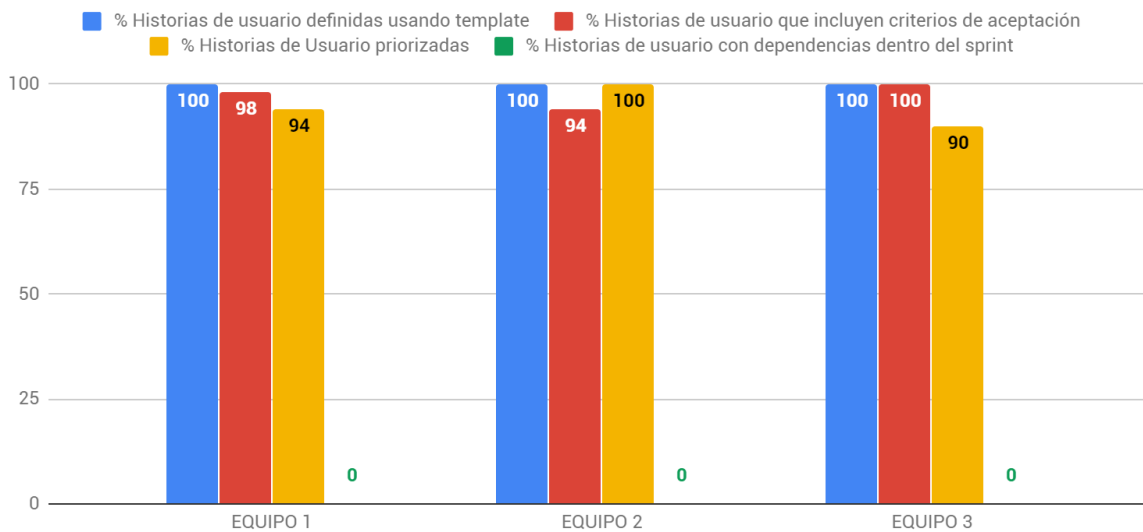


Figura 4.27 Análisis de atributos del Componente N° 4 de QuAM al usar QuAGI

Fuente: de elaboración propia

A continuación, se exponen los resultados obtenidos, luego del segundo control, por los equipos que trabajaron el seguimiento de sus proyectos ágiles con Taiga.

En primer lugar, se evidenció que la redacción de historias de usuario aún no seguía un patrón común dificultando la comprensión en cada proyecto. De hecho, como se ve en la Figura 4.28, y según los resultados de la encuesta en el segundo control, el 70% de los equipos representados por sus administradores aún no lograban mejorar la comprensión de sus historias de usuario.

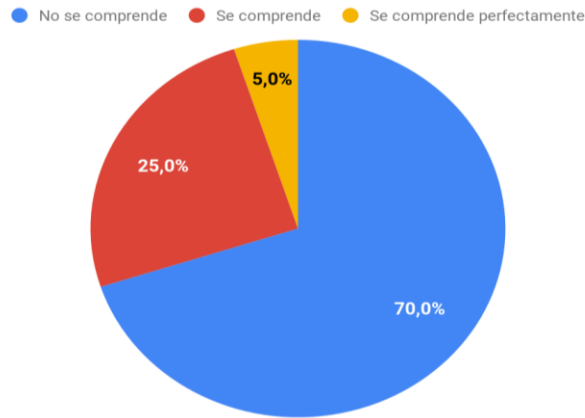


Figura 4.28 Grado de comprensión de historias de usuario en los equipos
Fuente: de elaboración propia

Luego, respecto al seguimiento del proyecto como se ve en la Figura 4.29, los valores han mejorado respecto a lo obtenido durante el primer punto de control y expuestos en la Figura 4.21, pero aun así no logran cubrir el 100% respecto a completitud, estimación y priorización de historias de usuario. Como se aprecia, se incrementó el número de historias de usuario completas cubriendo, aproximadamente, el 65% de historias de usuario que componían el sprint en ese momento. Luego, se observa en promedio un 34% de historias de usuario estimadas por equipo y se ve, también, un incremento en la cantidad de historias de usuario priorizadas con un total de 41% de cobertura de todos los proyectos analizados.

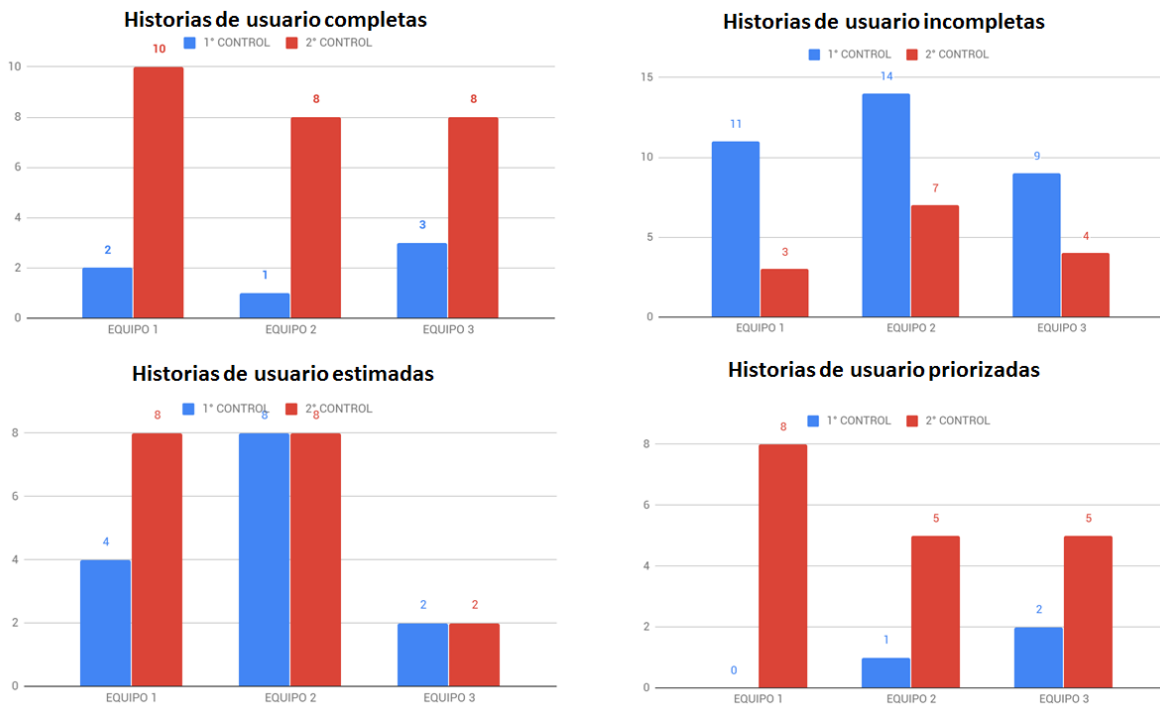


Figura 4.29 Avances en la producción de historias de usuario por equipo
Fuente: de elaboración propia

Por el lado de la distribución de tareas, se observan mejoras en los valores de acuerdo con lo que se presenta en la Figura 4.30, sin embargo, en todos los equipos más de un 20% de tareas sigue sin tener responsable asignado.

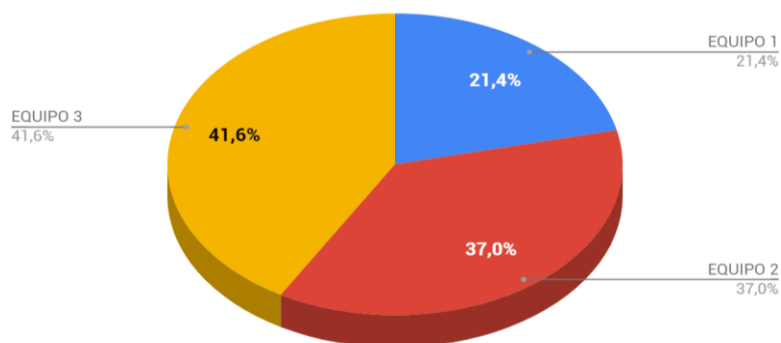


Figura 4.30 Porcentajes de tareas sin responsable asignado
Fuente: de elaboración propia

e) Conclusiones

Para concluir con este estudio de caso, teniendo en cuenta lo manifestado por los administradores de proyecto que han utilizado QuAGI, los informes de evaluación de calidad se han aproximado a la realidad de cada equipo y han reflejado correctamente las mejoras luego de los ajustes realizados en todos los casos. Los 3 administradores han coincidido en que la calidad de sus proyectos mejoró radicalmente respecto al componente N° 4 “*Gestión de Requerimientos*”, y desde allí se fueron logrando mejoras en los demás componentes impactando de manera positiva en la calidad de proceso ágil de cada proyecto.

Como se destaca en la Figura 4.31, al consultar a los administradores de los 6 equipos participantes que califiquen, en una escala del 1 (Muy malo) a 10 (Excelente), cómo han colaborado las herramientas de seguimiento utilizadas con la calidad del proceso ágil que caracteriza a cada proyecto, las mejores calificaciones se identifican en los equipos que han usado QuAGI (los cuales, como dato curioso, fueron los únicos en completar la implementación real de sus proyectos en las ONGs).

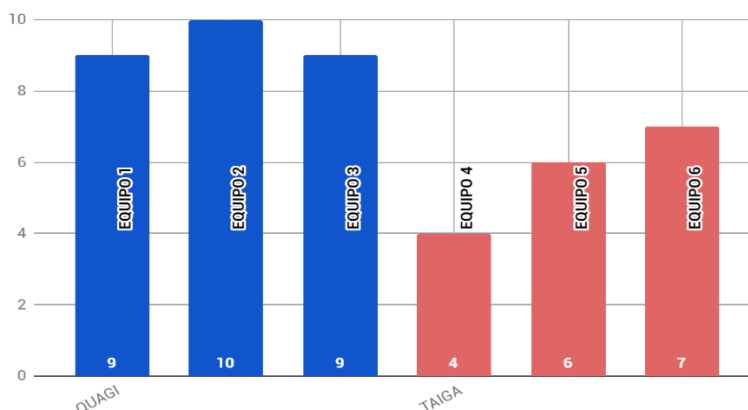


Figura 4.31 Grado de colaboración de la herramienta con la calidad del proceso
Fuente: de elaboración propia

De esta forma, se verifica el cumplimiento de la hipótesis planteada inicialmente demostrando que la utilización de QuAGI como herramienta de seguimiento mejora notoriamente el nivel de calidad del proceso asociado a los proyectos ágiles.

Tal como se realizó con los casos anteriores, y a fin de complementar el proceso de validación, se llevó adelante durante esta experiencia un nuevo estudio de usabilidad que tuvo por objetivo ampliar la información respecto al impacto del uso de ambas herramientas en proyectos ágiles. Fue así como al finalizar los proyectos se entregó a cada Administrador de Proyecto una encuesta para conocer por un lado cuestiones relacionadas a interfaz y por otro, muy importante, si el uso de la herramienta mejoró la calidad del proceso ágil desde la perspectiva del equipo.

Para la encuesta se utilizó la escala de usabilidad de un sistema (denotada en inglés por las siglas SUS) y presentada por Brooke en [85]. La medición de las respuestas respecto a una serie de afirmaciones presentadas utiliza un sistema de puntuación de 5 puntos (desde “completamente de acuerdo” hasta “completamente en desacuerdo”, y un algoritmo rápido de puntuación son componentes que resultan en una puntuación de usabilidad en una escala del 0 al 100. El compendio de afirmaciones (A1...A10) junto al sistema de procesamiento de respuestas se incluye en el Anexo G de esta tesis.

Habiéndose llevado a cabo la encuesta, y tal como se ve en la Tabla 4.2, los equipos 1 al 3, que han utilizado QuAGI como herramienta de gestión de sus procesos ágiles en cada proyecto, son los que mejor nivel han alcanzado obteniendo porcentajes superiores al 80%, calificados según Brooke como *De buena usabilidad*. En cambio, por el lado de los equipos 4 al 6, que han utilizado Taiga, los porcentajes de SUS no alcanzan el 40% en ningún caso calificando la usabilidad de la herramienta como *Ineficiente*.

Tabla 4.2 Valores obtenidos en la Encuesta de usabilidad a administradores de equipos

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	X	Y	SUS
Equipo 1	5	1	5	3	5	1	5	1	5	5	20	14	85
Equipo 2	4	1	5	1	5	2	5	1	5	4	19	16	87,5
Equipo 3	5	2	5	1	4	1	4	2	5	5	18	14	80
Equipo 4	3	3	2	4	2	4	3	3	1	2	6	9	37,5
Equipo 5	2	4	2	4	3	4	3	4	2	2	7	7	35
Equipo 6	2	4	1	4	3	3	2	4	1	1	4	9	32,5

Fuente: de elaboración propia

Como se puede observar en la Figura 4.32, el contraste es notorio entre los resultados obtenidos por la *encuesta de usabilidad* entre los equipos que han utilizado QuAGI como herramienta de seguimiento del proyecto ágil y los equipos que usaron Taiga. Observándose que en todos los casos las afirmaciones que destacan aspectos positivos tales como la facilidad de uso, la intención de utilizar la herramienta con frecuencia, la seguridad, etc. obtienen puntajes entre 4 (Muy de acuerdo) y 5 (Totalmente de acuerdo) en los equipos 1 al 3 (es decir los que han usado QuAGI durante el estudio de caso). En el caso de afirmaciones tales como A2 que resalta la complejidad innecesaria del sistema

como factor a tener en cuenta, son los equipos del 4 al 6 (equipos que han utilizado Taiga) los que expresan los peores resultados.

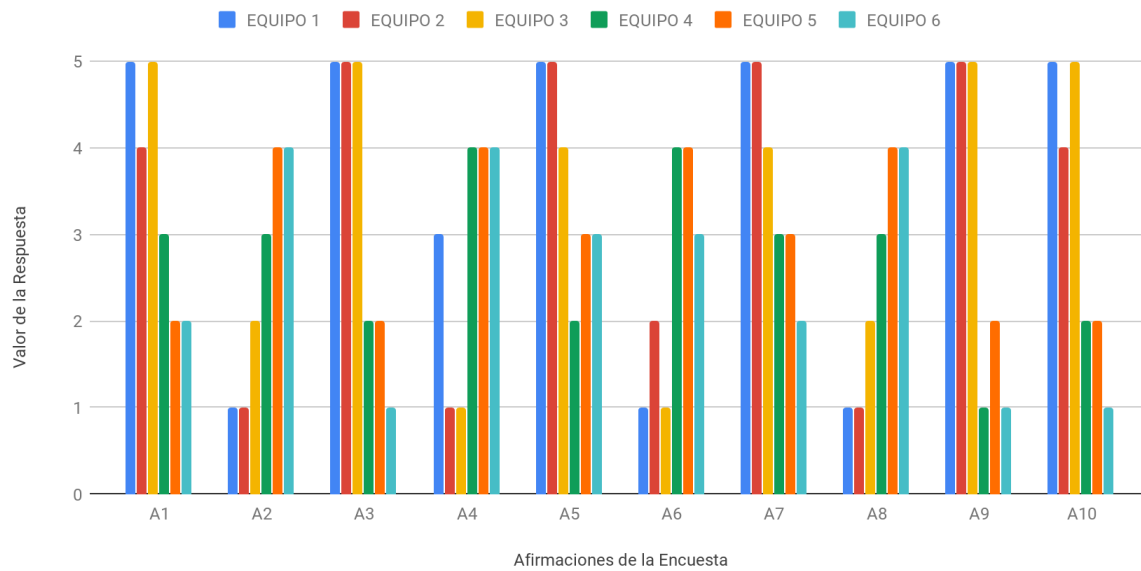


Figura 4.32 Resultados de la encuesta SUS
Fuente: de elaboración propia

A continuación, se exponen conclusiones de este capítulo, a las cuales se han arribado luego de llevarse a cabo cada uno de los estudios de casos presentados a lo largo de esta sección.

4.2 - Conclusiones

En este capítulo se ha presentado la evaluación formal del framework AQF, a través de estudios de casos, caracterizados por proyectos reales que han utilizado QuAGI como herramienta de seguimiento y como soporte para la evaluación de calidad del proceso de desarrollo basado en prácticas ágiles. Asimismo, a partir del análisis del funcionamiento de QuAGI, se han validado los componentes del modelo QuAM.

En base a los resultados obtenidos y presentados en la sección 4.1, mediante la evaluación detallada de los aspectos que se presentaron en cada caso, y gracias a la correlación entre los estudios de casos expuestos, se han logrado incorporar mejoras a la herramienta QuAGI a medida que se avanzó en la experiencia de validación.

En el estudio de caso N° 1 se llevó adelante la validación de la herramienta en un primer sprint de proyectos ágiles en 20 equipos de desarrollo de empresas PYMES de software. En función al uso de la herramienta QuAGI se realizó una encuesta a los equipos de forma tal de validar la estructura de QuAM analizando la utilidad de cada componente en la evaluación de calidad de procesos ágiles. Del estudio se concluyó que un 70% de los equipos participantes consideran útil la herramienta QuAGI para la evaluación de calidad de procesos ágiles gracias a que facilita el seguimiento de la estructura del modelo tal como se define en QuAM. En el caso del 30% que consideró de poca o nula ayuda utilizar QuAGI, se pudo relevar que se debió a la falta de informes representativos del proceso asociado al trabajo en equipo, factor fundamental para el éxito de proyectos ágiles de acuerdo con su perspectiva.

Los resultados del estudio de caso N° 1 dieron lugar al diseño del estudio de caso N° 2, el cual tuvo por objetivo validar la información que QuAGI ofrece para mejorar la calidad de trabajo en equipo en proyectos ágiles. El estudio de caso N° 2 estuvo constituido por 90 personas pertenecientes a 20 equipos, y se enfocó en los resultados proporcionados por el componente N° 2 de QuAM “Evaluación del trabajo en equipo”. Los resultados que se lograron, luego de analizar los informes proporcionados por QuAGI y recomendaciones realizadas durante reuniones posteriores a la implementación, fueron muy superiores a los obtenidos en un primer control sin aplicar las mejoras indicadas por la herramienta. No solo se observaron mejoras a nivel del trabajo en equipo, sino que esto impactó favorablemente a los demás componentes considerados al evaluar la calidad del proceso asociado al proyecto ágil correspondiente. La experiencia aportó además mejoras que eran necesarias a nivel de interfaz para continuar añadiendo valor a QuAGI como herramienta de seguimiento.

Finalmente, se diseñó un nuevo estudio de caso que facilite la validación integral de QuAGI teniendo en cuenta la evaluación de todos los componentes de QuAM. El estudio de caso N° 3 se caracterizó por analizar el rendimiento completo de QuAGI en equipos sin experiencia previa con otras herramientas de seguimiento en proyectos ágiles. Además, se incluyó en el estudio el análisis de la calidad de proyectos ágiles en equipos que utilizaron otra herramienta diferente con el objetivo de comparar resultados. Durante la experiencia se realizaron observaciones en forma gradual y en diferentes momentos acordados desde el inicio, registrándose cada caso para su posterior análisis. Concluyendo el estudio de caso, y confirmando la hipótesis definida, los resultados que se obtuvieron, han sido mucho mejores para la calidad de los procesos en los equipos que utilizaron QuAGI en sus proyectos ágiles

no solo a nivel de gestión sino también relacionados a curva de aprendizaje, experiencia de usuario y utilidad.

Además, el proceso de validación que se describió en este capítulo ha permitido obtener conclusiones y adquirir experiencias de las que se desprenden ciertas lecciones aprendidas. En primer lugar, luego de haberse ejecutado todos los estudios de caso se destaca cómo las recomendaciones ofrecidas por QuAGI en conjunto con el seguimiento de los componentes definidos en QuAM permite mejorar notoriamente la calidad de los procesos ágiles de cada proyecto. Del análisis se destaca también que los participantes subrayan entre los beneficios de QuAGI que sea una herramienta fácil de usar, útil y, sobre todo, manifiestan intención de usarla en el futuro pues la información obtenida resulta de vital importancia para el éxito de sus proyectos. Todo esto no hubiera sido posible sin analizar la aplicabilidad real de la evaluación de calidad de proyectos ágiles mediante el estudio de componentes definidos en QuAM y el seguimiento automático a través de QuAGI, logrado gracias al proceso de ejecución de estudios de casos.

En segundo lugar, la realización de esta experiencia de validación ha permitido, además, analizar el grado de adecuación, completitud y corrección de los resultados en los informes de calidad aportados por QuAGI; logrando modificaciones que aporten valor en cada instancia de validación y favorezca la obtención de la mejor versión de la herramienta.

Y, por último, pero no menos importante, el estudio de casos sobre proyectos ágiles reales ha posibilitado obtener realimentación de los participantes que aplicaron el modelo y usaron la herramienta. Esto no solo colaboró con mejoras a nivel de interfaz y usabilidad de la herramienta QuAGI, sino que también permitió añadir valor a través de mejoras en la configuración de los componentes de QuAM.

En el futuro será necesario replicar los diferentes estudios de casos en otros grupos para continuar analizando posibles factores de interacción que se deriven de cada experiencia.

Capítulo 5 - Conclusiones y Trabajos Futuros

***P**ara finalizar, en este capítulo se presentan las conclusiones, que responden a los objetivos e hipótesis propuestos para esta tesis doctoral y los hallazgos obtenidos como consecuencia del trabajo de investigación realizado.*

A estos fines, el capítulo se estructura de la siguiente manera: en la sección 5.1 se presentan conclusiones a las que se han arribado en base a un análisis del cumplimiento a los objetivos e hipótesis planteados en el capítulo 1 de esta tesis. Luego en la sección 5.2 se detallan los resultados de divulgación de esta tesis doctoral a través de las publicaciones científicas y comunicaciones a congresos realizadas durante su desarrollo. Y, finalmente, en la sección 5.3 se presentan futuras líneas de investigación que se consideran de interés a la problemática objetivo de esta tesis doctoral.

5.1 - Conclusiones

Al inicio de este trabajo, en el capítulo 1, se presentaron una serie de objetivos específicos cuyo cumplimiento implica la concreción del objetivo principal de esta tesis: *Proponer un framework que facilite el seguimiento de proyectos y la evaluación de calidad de procesos en empresas PYMES que implementen prácticas ágiles en el desarrollo de software*. Dicho framework deberá incluir un modelo de calidad y una herramienta de soporte que automatice los componentes de dicho modelo.

A continuación, se analizará el cumplimiento de cada uno de los objetivos específicos:

1. *Realizar un estudio de trabajos previos relacionados a modelos de calidad y herramientas de software aplicables a procesos ágiles en contextos de empresas PYMES.*

El estudio realizado se describe en el Capítulo 2 de esta tesis; el mismo incluyó el análisis de trabajos previos donde se evidenciaban propuestas e iniciativas relacionadas al proceso de evaluación de calidad de procesos ágiles y su aplicación en proyectos de software. En la sección 2.1 se ha presentado el análisis realizado de las propuestas existentes para la evaluación de calidad de proyectos de software guiados por prácticas ágiles. Para facilitar y unificar los criterios a tener en cuenta durante el análisis de las diferentes propuestas se llevó adelante un proceso de revisión sistemática aplicado a las principales fuentes de búsquedas del ámbito de la Ingeniería de Software. Las principales conclusiones a las que se llegaron son:

- Si bien existen en la literatura investigaciones recientes que denotan interés sobre la evaluación de calidad en proyectos de software desarrollados bajo enfoque ágil, muy pocos estudios han evaluado el impacto de aspectos de calidad sobre procesos ágiles en equipos, áreas o empresas de desarrollo de software con una visión integral.
- Además, ninguno de los trabajos previos analizados expone factores de impacto al implementarse la propuesta que presentan, ni tampoco ofrecen una visión integradora que permita definir estrategias transversales para mejorar el proceso de la evaluación de calidad que se estudia.
- También, es importante destacar que, si bien el 50% de los trabajos previos presentan modelos o herramientas para la evaluación de calidad de procesos ágiles de software, en ningún caso es posible identificar los componentes o factores que son considerados para la evaluación. Sin embargo, en esta tesis se ha presentado AQF, un framework que propone la integración de un modelo de calidad, un proceso de evaluación y un conjunto de herramientas que permitan automatizar la gestión de componentes en pos de mejorar la calidad de procesos de desarrollo de software guiados por prácticas ágiles.
- Por último, otra importante conclusión a la que se ha arribado radica en el hecho que, ningún trabajo previo, estudiado durante la revisión sistemática de la literatura, evidencia continuidad en el tiempo por parte de sus autores. A diferencia de la propuesta presentada en esta tesis, la cual al día de hoy se

encuentra disponible para su uso e implementación en los equipos que así lo requieran.

2. *Proponer un modelo de evaluación de calidad sobre procesos de desarrollo de software guiados por prácticas ágiles, en base a la definición de un conjunto de componentes.*

En el Capítulo 3 de esta tesis se presenta la configuración del framework AQF, como propuesta para la evaluación de calidad de procesos ágiles. De esta forma, en la sección 3.1 se explica el diseño y se exponen discusiones acerca de una instancia del modelo QuAM (Quality Agile Model), describiendo cada uno de sus componentes, atributos y métricas asociados.

3. *Diseñar e implementar herramientas de software que soporten la gestión de componentes del modelo de calidad e integren un framework que evaluar la calidad en procesos ágiles.*

Una vez definido el modelo QuAM, se describe en la sección 3.2 la implementación de QuAGI, una aplicación web desarrollada para proporcionar el seguimiento de proyectos ágiles y permitir la evaluación de calidad del proceso subyacente, a través de la automatización de los componentes del modelo. Se presentó QuAGI desde la arquitectura de la aplicación hasta sus funcionalidades, incluyendo la implementación del framework completo gracias a la integración del modelo junto con la herramienta de software.

4. *Analizar y estudiar el comportamiento del framework propuesto utilizando un método de validación sobre entornos reales de producción de software.*

Una vez finalizada la implementación tecnológica del framework, se ha realizado la validación de este sobre proyectos ágiles reales en producción.

Por un lado, se valida el funcionamiento de QuAGI por medio del diseño de 3 estudios de casos, los cuales fueron diseñados en función a objetivos propuestos de validación. En el Capítulo 4 se describen estos estudios, y se muestran los resultados obtenidos en cada caso, de modo tal que lo obtenido en una experiencia permitió diseñar el próximo estudio logrando sinergia durante todo el proceso de validación.

Del proceso de validación pudo concluirse que el uso de QuAGI y sus recomendaciones en los informes de calidad basados en la automatización de los componentes de QuAM, permiten mejorar notoriamente la calidad del proceso de desarrollo ágil en proyectos de software, lo cual, a su vez, impacta directamente en la calidad de su producción final. Y, además, se ha logrado determinar que QuAGI no solo es funcionalmente útil para los equipos de proyectos ágiles, sino que también les resulta atractiva y fácil de usar durante todo el proceso de desarrollo.

Finalmente, el trabajo de esta tesis ha permitido integrar en su propuesta de framework la presentación de un nuevo Modelo de Calidad, QuAM, describiendo de forma detallada cada uno de los componentes que se evalúan; y la implementación de una herramienta de software, QuAGI, que de soporte automático a lo establecido en QuAM. Y, mejor aún, la propuesta ha

sido adaptada, en base a estudios de casos de validación, para que pueda utilizarse en entornos reales de producción con proyectos de software cuyo desarrollo es guiado por prácticas ágiles.

A continuación, se exponen líneas futuras de investigación y aportes de divulgación científica, generadas en base al trabajo presentado en esta tesis.

5.2 - Trabajos Futuros

A partir del desarrollo de esta tesis, han surgido cuestiones que se relacionan a la calidad del software y que originarán líneas de investigación futuras. Algunas de ellas, simplemente no se habían considerado como objetivos de esta tesis; y otras, han surgido durante el desarrollo de la misma y ya se encuentran en marcha. A continuación, se resumen las principales líneas futuras de investigación:

I. Mejoras a la herramienta QuAGI

Si bien el framework propuesto en esta tesis aporta sistematicidad a la gestión integral de proyectos ágiles y permite evaluar la calidad del proceso de desarrollo que subyace a los mismos, queda como tema de trabajo pendiente, incorporar a AQF una API que permita a QuAGI, consumir datos desde otra herramienta de seguimiento de proyectos, sin necesidad que los equipos deban migrar lo realizado hasta el momento y, lograr igualmente, la obtención de reportes de calidad asociados al proceso.

II. Ampliación del framework AQF

En consonancia con lo anterior, se prevé el desarrollo de un agente inteligente que permita recomendar acciones al equipo de forma tal de mejorar los niveles de calidad del proceso ajustando los factores que sean necesarios. Esto surge como resultado de las interacciones logradas con los equipos, quienes manifiestan la necesidad de contar con herramientas que den soporte a su actividad diaria, a partir de recomendaciones automáticas que surjan del seguimiento del proyecto ágil y sus actividades, las cuales muchas veces son afectadas por acciones en segundo plano que pasan desapercibidas e impactan negativamente en los niveles de calidad del proceso de desarrollo asociado. Así, las mejoras a QuAGI y la incorporación de una nueva herramienta al framework AQF, formarán parte de la presentación de un nuevo proyecto de investigación que permita la obtención de una versión actualizada de AQF incluyendo ambos aspectos y nuevas mejoras que surjan de experiencias de validación futuras.

III. Impacto de las emociones del usuario en la percepción de la calidad del software

El trabajo de investigación llevado adelante durante esta tesis ha permitido conocer, además, que ya no solo se hace necesario considerar la calidad del software desde la perspectiva del producto y del proceso, sino también considerar a la experiencia del usuario como un indicador más a tener en cuenta. Profundizar en el estudio de la calidad del software significa, ahora, evaluar cómo las emociones pueden detectarse y comunicarse a través de la interacción del producto, y cuál es el impacto, que esto produce, en la percepción de calidad del software por parte de usuarios finales.

El desafío de futuras investigaciones será el diseño de una metodología de evaluación de calidad del software, que observe las emociones generadas producto de la interacción con usuarios. Actualmente, esto forma parte de las actividades del proyecto “Evaluación del impacto de las emociones en la calidad de software desde el punto de vista del usuario” (UTN-PID 5517), surgido a raíz del trabajo de investigación de esta tesis doctoral, el cual ha sido aprobado para su financiación por la Secretaría de Ciencia y Tecnología de UTN para el período 2019-2022.

5.3 -Publicaciones

Durante el desarrollo de esta tesis se han divulgado y comunicado resultados parciales de la investigación a través de las siguientes publicaciones:

1. *Capítulos de Libros*

a. **Título: “Quality Evaluation in Agile Process: A First Approach”**

Autores: Noelia Pinto, César Acuña, Liliana Cuenca Pletsch

COMPUTER SCIENCE & TECHNOLOGY SERIES. XXII Argentine Congress of Computer Science. Selected Papers. EdiT. EdULP. ISBN: 978-987-4127-28-0 / Capítulo: XIII Software Engineering Workshop - Año 2017 – Página 131

2. *Publicación con referato en revistas internacionales*

a. **Título: “Validation of the reengineering applied on the first version of Agile Quality Framework “**

Autores: Pinto, N., Tortosa, N., Cabas Geat, B., Ibáñez, L., & Acuña, C. (2019).

Electronic Journal of SADIO (EJS), 18(1), 93-109. Marzo 2019

Recuperado a partir de <https://www.cais.org.ar/index.php/EJS/article/view/87>

3. *Publicación con referato en revistas nacionales*

a. **Título: “Evaluación de calidad en procesos ágiles desde la perspectiva del trabajo en equipo”**

Autores: Noelia PINTO, César J. ACUÑA, Nicolás TORTOSA (2019)

Revista Digital del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de La Matanza (ReDDI). Universidad Nacional de La Matanza. ISSN: 2525-1333 | Vol.:4-Nro.1 (Julio-2019)

Recuperado a partir de

<https://reddi.unlam.edu.ar/index.php/ReDDi/article/view/78/156>

b. **Título: “WQF en acción: Una experiencia de evaluación de la calidad en aplicaciones web en la industria del software del NEA”**

Autores: Noelia Pinto, Nicolás Tortosa, Blas Cabas Geat, Maximiliano Ulibarrie, Liliana Cuenca Pletsch.

Revista Argentina de Ingeniería (RADI). Consejo Federal de Decanos de Ingeniería de Argentina (CONFEDI). AÑO 5 - VOLUMEN 9 - MAYO DE 2017

Recuperado a partir de <https://radi.org.ar/wp-content/uploads/2017/08/RADI-9-MAYO-DE-2017-WEB-21.pdf>

4. *Presentaciones en reuniones científicas internacionales*

- a. **Título: “Quality evaluation of agile processes: Measurement of requirements management using AQF v2”**

Autores: Noelia Pinto; Nicolás Tortosa; Blas Cabas Geat; Lucas Ibáñez; Verónica Bollati

Proceedings of 11th International Conference on the Quality of Information and Communications Technology (QUATIC), Coimbra, 2018, pp. 15-20.

- b. **Título: “Validación de la reingeniería aplicada sobre Agile Quality Framework”**

Autores: Noelia Pinto; Nicolás Tortosa; Blas Cabas Geat; Lucas Ibáñez; César Acuña

Proceedings of Argentine Symposium on Software Engineering (ASSE).
Septiembre 2018. ISSN: 2451-7593

- c. **Título: “QUAGI: Una propuesta para el seguimiento y evaluación de proyectos de Software Ágiles”**

Autores: Noelia Pinto, Gabriela Tomaselli, César J. Acuña, Liliana Cuenca Pletsch

Anais do V SABTIC, VIII STIN e XVIII Fórum , Três de Maio, Brasil, 25 e 26 de Maio de 2017. DOI: 10.5281/zenodo.583174

- d. **Título: "Validación del diseño de componentes de QuAM: un Modelo de Calidad para procesos Ágiles"**

Autores: Noelia Pinto, Gabriela Tomaselli, Liliana Cuenca Pletsch, Nicolás Tortosa, César J. Acuña

Anales de IV Seminario Argentina-Brasil de Tecnologías de la Comunicación y la Comunicación (SABTIC). Noviembre 2016, Corrientes, Argentina. ISBN 978-987-3619-15-1

5. *Presentaciones en reuniones científicas nacionales*

- a. **Título: “Evaluación de Calidad en Procesos Ágiles desde la Perspectiva del Trabajo en Equipo”**

Autores: Noelia Pinto; Blas Cabas Geat; César J. Acuña; Nicolás Tortosa

Publicación on line - ISSN 2347-0372

6to Congreso Nacional de Ingeniería Informática/Sistemas de Información (CONAIISI). Noviembre 2018, Mar del Plata, Argentina.

Se destaca, además, que el artículo ha obtenido el reconocimiento del Comité Académico al recibir el Premio “Mejor Trabajo” de esa edición del Congreso.

b. Título: “Evaluación de Calidad de Gestión de Requerimientos en proyectos ágiles”

Autores: Noelia Pinto; Blas Cabas Geat; Nicolás Tortosa; Lucas Ibañez; César Acuña

Anales del IV Congreso Argentino de Ingeniería. X Congreso Argentino de Enseñanza de la Ingeniería (CADI-CAEDI). Septiembre 2018, Córdoba, Argentina.

c. Título: “Evaluando la Calidad en Desarrollos Ágiles: Caso de Estudio con PYMES de Software de Chaco y Corrientes”

Autores: Noelia Pinto; César J. Acuña; Gabriela Tomaselli; Nicolás Tortosa; Blas Cabas Geat

Anales del 5to Congreso Nacional de Ingeniería Informática/Sistemas de Información (CONAIISI). Noviembre 2017, Santa Fe, Argentina. ISSN: 2347-0372

d. Título: “Evaluating Quality in Agile Developments. A first validation experience with NEA Software SMEs”

Autores: Noelia Pinto; César Acuña; Nicolás Tortosa; Blas Cabas Geat

Anales del XXIII Congreso Argentino de Ciencias de la Computación (CACIC). Octubre 2017, La Plata, Argentina. ISBN 978-950-34-1539-9

e. Título: “Evaluación de Calidad en Proyectos Ágiles: Una experiencia en pequeñas y medianas empresas del NEA”

Autores: Noelia Pinto; Gabriela Tomaselli; Analía Montero; Liliana Cuenca Pletsch; César Acuña

Anales del 1er. Congreso Latinoamericano de Ingeniería (CLADI). Septiembre 2017, Entre Ríos Argentina.

f. Título: “Una plataforma tecnológica para el seguimiento y evaluación de calidad de proyectos ágiles”

Autores: Nicolás Tortosa, Blas Cabas Geat, Noelia Pinto

Anales del IXX Workshop de Investigadores en Ciencias de la Computación (WICC). Abril 2017, CABA, Argentina. ISBN 978-987-42-5143-5

g. Título: “Quality Evaluation in Agile Process: A First Approach”

Autores: Noelia Soledad Pinto, Liliana Raquel Cuenca Pletsch, César Javier Acuña

Anales del XXII Congreso Argentino de Ciencias Informáticas y Computación (CACIC). Octubre 2016, San Luis, Argentina. ISBN 978-987-733-072-4

h. Título: “Hacia un modelo de Evaluación de Calidad de Procesos Ágiles”

Autores: Noelia Pinto, Gabriela Tomaselli, César Acuña, Liliana Cuenca Pletsch, Nicolás Tortosa, Blas Cabas Geat, Maximiliano Ulibarrie

Anales del 4º Congreso Nacional de Ingeniería Informática/Sistemas de Información(CONAIISI). Noviembre 2016, Salta, Argentina.

i. Título: “Evaluación de Calidad en entornos ágiles: un estudio comparativo de la literatura”

Autores: César Acuña, Noelia Pinto, Gabriela Tomaselli, Nicolás Tortosa

Anales del III Congreso Argentino de Ingeniería (CADI). Septiembre 2016, Resistencia, Argentina. ISBN 978-950-42-0173-1

6. Patentes y Propiedad Intelectual.

Por último, también como aporte científico, la herramienta QuAGI, resultado de esta tesis doctoral, ha logrado obtener el Registro como Obra Inédita de Software ante la Dirección Nacional de Derecho de Autor dependiente del Ministerio de Justicia y Derechos Humanos de la República Argentina, aprobado con fecha 25 de Junio de 2019 y publicado en Boletín Oficial del Poder Ejecutivo Nacional, con el Número RE-2019-57162220-APN-DNDA-#MJ

ANEXO A

Se incluye aquí la lista de estudios seleccionados resultado de la revisión sistemática realizada:

ID	G1
Título	Assisting the continuous improvement of scrum projects using metrics and bayesian networks
Año de Publicación	2017
Autores	Mirko Perkusich, Kyller Costa Gorgônio, Hyggo Almeida, Angelo Perkusich
Información relevante para el estudio	
Metodología	Elaborar un modelo basado en métricas, a partir de una red bayesiana basada en el conocimiento extraído de la literatura y los expertos. Dicho modelo proporciona datos para ayudar a Scrum Masters a liderar la mejora de la entrega de valor comercial de los equipos de Scrum. Realizaron caso de estudio de 10 escenarios simulados para validar el modelo y la inferencia de la red bayesiana.
Resultados obtenidos	Luego de definir la primer versión del modelo se validó exitosamente con 10 escenarios simulados y concluyó que el modelo representa adecuadamente los proyectos de Scrum desde el punto de vista del ScrumMaster

ID	G2
Título	Quality of software requirements specification in agile projects: A cross-case analysis of six companies.
Año de Publicación	2018
Autores	Juliana Medeiros ; Miguel Goulão ; Alexandre Vasconcelos ; Carla Silva
Información relevante para el estudio	
Metodología	Se realizaron seis casos de estudio en empresas del medio industria para caracterizar la actividad de especificación de requisitos.
Resultados obtenidos	Luego del análisis y la síntesis cruzada de los seis casos de estudio, se obtuvo como resultado un modelo que define la simplicidad y la objetividad como factores de calidad esenciales de especificación de requisitos utilizado practicas agiles de desarrollo de software.

ID	G3
Título	An Empirical Study on the Role of Requirement Engineering in Agile Method and Its Impact on Quality
Año de Publicación	2015
Autores	Anzira Rahman
Información relevante para el estudio	
Metodología	Caso de estudio de 16 proyectos para realizar un estudio comparativo para analizar el impacto en la calidad, en la implementación de ingeniería de requisitos utilizando prácticas de desarrollo de software agile, en contraste con metodología en cascada.
Resultados obtenidos	Primero, los proyectos que utilizan prácticas de desarrollo de software agile tenían un número significativamente mayor de defectos. En segundo lugar, cuando los defectos se categorizaron en requisitos no funcionales y funcionales, las comparaciones entre los dos grupos revelaron que los proyectos con prácticas de desarrollo de software agile tenían más defectos relacionados con requerimientos no funcionales que Waterfall expresivamente. Estos resultados sugieren que la elección de prácticas ágiles puede llevar a un mayor número de defectos y, por lo tanto, disminuir la calidad, posiblemente debido a la falta de comprensión de los requisitos

ID	S1
Título	Patterns of Collaboration Driven by Requirements in Agile Software Development Teams
Año de Publicación	2017
Autores	Irum Inayat, Sabrina MarczakSiti, Salwah Salim, Daniela Damian
Información relevante para el estudio	
Metodología	Se realizaron múltiples casos de estudio para analizar la colaboración basada en los requisitos traducida como comunicación y conocimiento entre equipos ágiles de cuatro proyectos distribuidos.
Resultados obtenidos	Los equipos ágiles confían en la comunicación ad hoc y en los patrones dinámicos de intercambio de conocimientos. En este estudio se identificaron algunos patrones en la colaboración de equipos ágiles centrados en los requisitos. Los siete patrones principales de colaboración se refieren a qué roles se comunican, así como qué temas predominan en las discusiones centradas en los requisitos y qué factores contribuyen a la conciencia en los equipos distribuidos.

ID	S2
Título	Agile Quality Requirements Management Best Practices Portfolio: A Situational Method Engineering Approach
Año de Publicación	2017
Autores	Lidia López, Woubshet Behutiye, Pertti Karhapää, Jolita Ralyté, Xavier Franch, Markku Oivo
Información relevante para el estudio	
Metodología	investiga qué prácticas industriales, a partir de los métodos ágiles, pueden utilizarse para una mejor gestión de los requisitos de calidad en el desarrollo de software ágil. Utilizamos la ingeniería de métodos situacionales para identificar, complementar y clasificar una cartera de mejores prácticas para la gestión de requisitos de calidad en entornos ágiles.
Resultados obtenidos	Los primeros resultados informados corresponden a la obtención de los requisitos del método. Las pautas incluyen las mejores prácticas para cumplir con cuatro intenciones diferentes: obtención de requisitos de calidad, especificación, comunicación y verificación y validación, y las tres estrategias para cumplir con la intención de especificación: por refinamiento, documentación y priorización. Hasta el momento, se identificaron dos estrategias concretas para la priorización: la priorización por urgencia (orientada a las issues) y la priorización basada en el valor (impulsada por el valor).

ID	S3
Título	Scrumban/XP: A New Approach to Cover the Third Level of CMMI Model
Año de Publicación	2016
Autores	Bougroun, Z., Zeaaraoui, A., & Bouchentouf, T.
Información relevante para el estudio	
Metodología	El artículo realiza una comparación entre CMMI Nivel 3 y Scrum. Enfocado en debilidades del enfoque Scrum y destacando prácticas de CMMI que mejorarían la implementación de Scrum. Además profundizan en prácticas de otros enfoques tales como XP y Kanban, haciendo referencia a una propuesta que combine los 3 enfoques ágiles para cubrir la implementación de CMMI Nivel 3.
Resultados obtenidos	Durante la implementación, el estudio recomienda la combinación de enfoques Scrum y Kanban, a través de prácticas de Scrumban. Y por otro lado propone también implementar prácticas de XP. Estos enfoques aplicados de forma conjunta en diferentes prácticas permiten cubrir el 66% del Nivel 3 de CMMI, mientras que aplicar solo Scrum lograría una cobertura de aproximadamente solo el 40%.

ID	S4
Título	The Disciplined Agile Process Decision Framework
Año de Publicación	2016
Autores	Scott W. Ambler, Mark Lines
Información relevante para el estudio	
Metodología	El artículo presenta una revisión de casos de implementación del framework <i>Disciplined Agile Process Decision (DAPD)</i> , un marco para ayudar a las organizaciones a racionalizar sus procesos de tecnología de la información (TI) de una manera sensible al contexto en pos de lograr, entre otras cosas, Calidad de Software.
Resultados obtenidos	El artículo logra la obtención de una serie de principios efectivos para los equipos que implementen proyectos ágiles. Respecto a la calidad de software el artículo concluye que los resultados repetibles son mucho más importantes que los procesos repetibles. De acuerdo a lo establecido por el framework DAPD, los equipos efectivos se centran en producir resultados repetibles, como la entrega de software de alta calidad que satisfaga las necesidades de las partes interesadas de manera oportuna y rentable.

ID	A1
Título	Procedural Assessment Process of Software Quality Models Using Agility.
Año de Publicación	2015
Autores	Ahmad Al-Elaimat, Al-Ghuwairi Abdel-Raham.
Información relevante para el estudio	
Metodología	Elaboración de un multimodelo de calidad que contenga ISO / IEC 12207 para el proceso del ciclo de vida del desarrollo de software en conjunción con el marco CMMI para mejorar el proceso de desarrollo, en conjunción con ISO / IEC 25010.
Resultados obtenidos	Al Aplicar el multimodelo sugerido, las empresas de la industria del software mejoraran sin duda la productividad. Para demostrar que la productividad del uso de software de calidad multimodelo con enfoque ágil excede el uso de cualquier otro marco, realizaran un estudio empírico en un trabajo futuro.

ID	A2
Título	GuideGen: a tool for keeping requirements and acceptance tests aligned.
Año de Publicación	2018
Autores	Sofija Hotomski, Martin Glinz.
Información relevante para el estudio	
Metodología	Se presenta el desarrollo de GuideGen, una herramienta que ayuda a los ingenieros de requisitos, testers y otras partes involucradas a mantener alineados los requisitos y las pruebas de aceptación.
Resultados obtenidos	Se realizó un caso de estudio con datos de tres empresas. Para 262 cambios no triviales de los requisitos, las sugerencias generadas por GuideGen fueron correctas en más del 80 por ciento de los casos para los requisitos ágiles y alrededor del 67 por ciento en los tradicionales. Actualmente se sigue evaluando la utilidad y aplicabilidad de la herramienta, y aplicando mejoras a las funcionalidades.

ID	I1
Título	Formative evaluation of a tool for managing software quality.
Año de Publicación	2017
Autores	Liliana Guzman, Anna Maria Vollmer, Marcus Ciolkowski, Michael Gillman.
Información relevante para el estudio	
Metodología	Realizar entrevistas, talleres y un estudio de mapeo para operacionalizar la calidad del prototipo ProDebt, una herramienta que tiene por objetivo proporcionar a los profesionales asistencia para gestionar la calidad del software y la deuda técnica, y así lograr identificar instrumentos confiables para medirlo.
Resultados obtenidos	Los participantes de la evaluación expresaron que la información proporcionada por el prototipo ProDebt es comprensible y relevante. Consideraron las funcionalidades del prototipo ProDebt como fáciles de usar pero de uso limitado. Identificaron necesidades de mejora, por ejemplo, que los resultados del análisis deberían estar vinculados a otras fuentes de información.

ID	12
Título	A Reference Model for Agile Quality Assurance: Combining Agile Methodologies and Maturity Models
Año de Publicación	2014
Autores	Fernando Selleri Silva ; Felipe Santana Furtado Soares ; Angela Lima Peres ; Ivanildo Monteiro de Azevedo ; Pietro Pereira Pinto ; Silvio Romero de Lemos Meira.
Información relevante para el estudio	
Metodología	Diseño de un modelo de referencia ágil para el aseguramiento de la calidad, para ayudar a las organizaciones en la implementación de control de calidad. La propuesta del modelo considera los resultados de una revisión sistemática de la literatura sobre CMMI y ágil, y un estudio de caso sobre control de calidad en una organización con CMMI y ágil.
Resultados obtenidos	Las evaluaciones iniciales del modelo del modelo se realizaron a través de encuestas. Obteniendo resultados positivos, los cuales indican que el modelo puede contribuir a ayudar a las organizaciones a evaluar su situación actual sobre prácticas ágiles y apoyar la mejora de la calidad mediante la adopción de prácticas ágiles de control de calidad.

ID	13
Título	A Quantitative Framework for Task Allocation in Distributed Agile Software Development
Año de Publicación	2018
Autores	Waqar Aslam, Farah Ijaz.
Información relevante para el estudio	
Metodología	Se propone un marco de asignación de tareas, basado en un modelo, que consta de dos fases: una, la identificación de factores y dependencias que influyen fuertemente en la decisión de asignación de tareas y además un método cuantitativo que asigna tareas a los miembros del equipo que mejor cumplen los requisitos de la tarea.
Resultados obtenidos	El artículo expone resultados cuantitativos en la implementación de un modelo que permita mejorar la calidad en procesos ágiles de desarrollo de software con equipos distribuidos. Para ello establece una serie de estrategias de acuerdo a lo definido por el modelo y que guían las mejoras propuestas.

ANEXO B

Se presenta a continuación el compendio de 15 preguntas utilizadas para las encuestas a empresas respecto a situación actual en cuanto a Calidad y procesos de desarrollo de software. Las preguntas se estructuraron en función a 3 secciones: Datos demográficos, Proyectos de Software y Calidad del Proceso de desarrollo.

SECCIÓN A > DATOS DEMOGRÁFICOS	
1	<i>Localización de la Organización</i>
2	Sobre quien responde: <i>¿Cuál es su rol dentro de la Organización?</i> Por ejemplo: Project Manager, Lider de Equipo, Scrum Master, Analista de Negocio, etc.
3	<i>Ámbito de Trabajo</i> Ámbito Privado /Público (Área de Desarrollo / Sistemas / SI/TI) Empresa dedicada al Desarrollo de Software
4	<i>Tipos de Productos</i> Desktop - Web - Aplicaciones Móviles - HW Firmware
5	<i>Tamaño de la Organización</i> (Cantidad aproximada de integrantes/empleados)
SECCIÓN B > PROYECTOS DE SOFTWARE	
<i>B.1. Gestión de Proyectos</i>	
6	<i>Utilización de herramientas de gestión. Cuáles</i> (Herramientas ofimáticas, tableros digitales, etc)
7	<i>Actualizarían la herramienta de gestión de proyecto</i>
8	<i>Trabajan con equipos distribuidos</i>
<i>B.2. Enfoque Ágil</i>	
9	<i>Porcentaje de proyectos ágiles en la organización</i>
10	<i>Enfoques ágiles que conocen/utilizan</i>
11	<i>Cuáles son las prácticas ágiles que conoce</i>
12	<i>Cuáles son las prácticas ágiles que utiliza</i>
SECCIÓN C > CALIDAD DEL PROCESO	
13	<i>Conoce sobre Calidad de Software</i>
14	<i>Considera que la Calidad de Software mejoraría la productividad</i>
15	<i>Cuáles son los factores que considera impactan sobre la Calidad de Software</i>

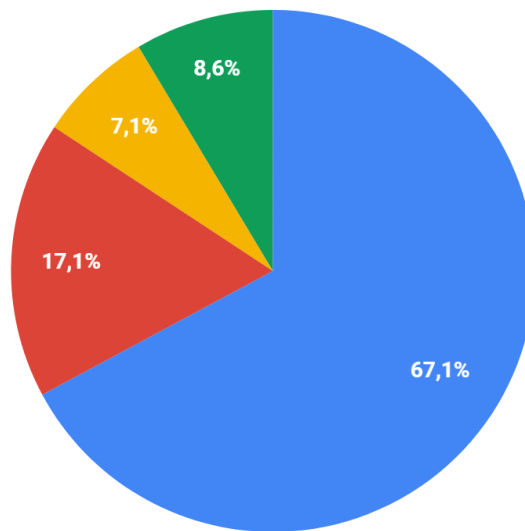
ANEXO C

Se exponen en este Anexo los resultados obtenidos del relevamiento de datos demográficos de los participantes en la encuesta de Situación Actual, cuyo análisis de presenta en el Capítulo 3 de esta tesis.

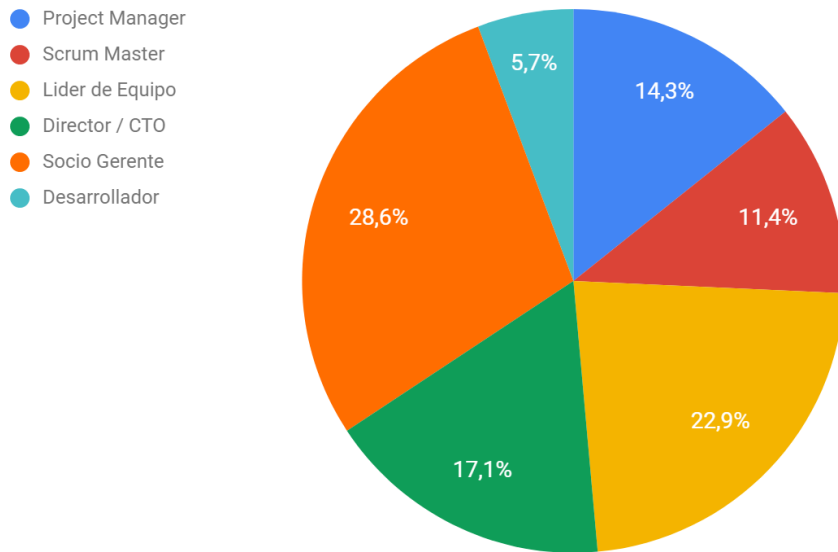
Instrumento: Entrevista
Población: Empresarios, Jefes de área de Sistemas y Responsables de proyectos de software que forman parte de empresas o áreas de software, localizados en Formosa, Misiones, Chaco y Corrientes
Muestra: 70 personas con roles en proyectos ágiles
Período: 2 meses
Mecanismo: Reuniones presenciales o a través de videoconferencias

Pregunta 1: Localización

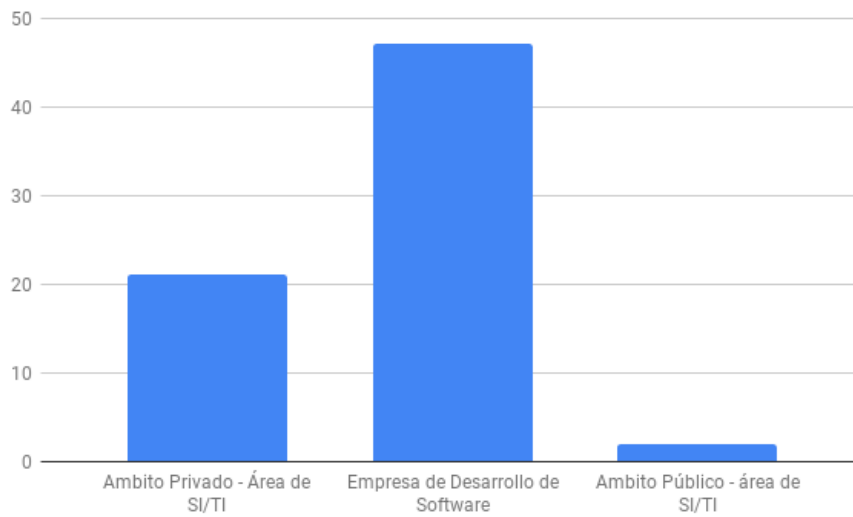
- Chaco
- Corrientes
- Formosa
- Misiones



Pregunta 2: ¿Qué rol ocupa en la organización?

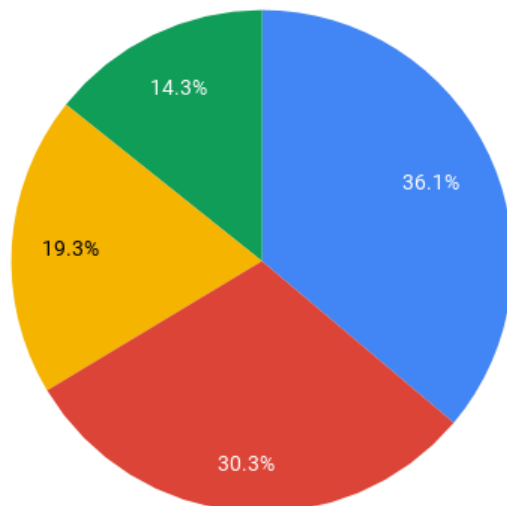


Pregunta 3: Tipifique su ámbito de trabajo



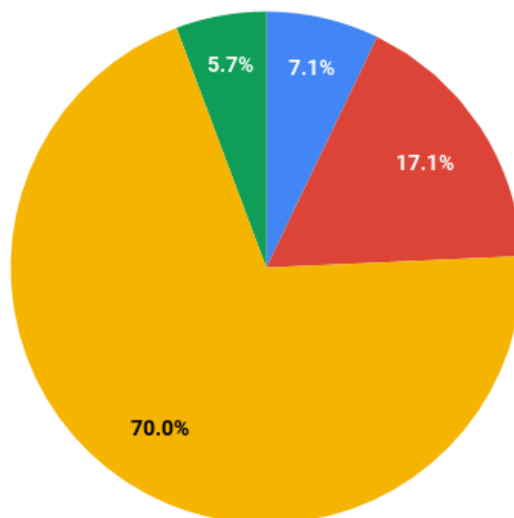
Pregunta 4: ¿Qué tipos de proyectos trabaja su organización?

- Desktop
- Web
- Aplicaciones Móviles
- HW Firmware



Pregunta 5: Indique el tamaño de su organización

- Menos de 5 personas
- Entre 5 y 10 personas
- Más de 10 personas
- Más de 50 personas



ANEXO D

Funcionalidades de QuAGI

a) Configuración inicial del Proyecto

QuAGI permite la carga de información básica del proyecto tal como se observa en la Figura F.1, además se puede editar y añadir más información a medida que se avanza en el proyecto.

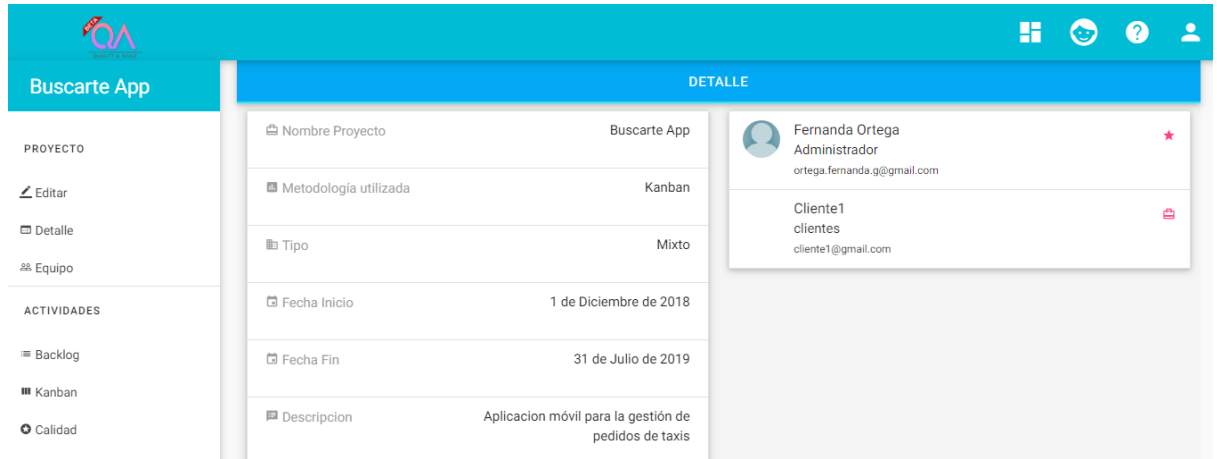


Figura D.1 Interfaz de QuAGI - Información básica del Proyecto
Fuente: de elaboración propia

QuAGI a su vez permite configurar el equipo de trabajo que formará parte del Proyecto a partir de la carga de datos en 3 perfiles: Administrador del Proyecto/Líder del Proyecto, Desarrolladores y Product Owner (o Cliente).

En la Figura F.2 se puede ver de qué forma se estructura la interfaz para mostrar la información correspondiente a los participantes del proyecto en todos los perfiles.

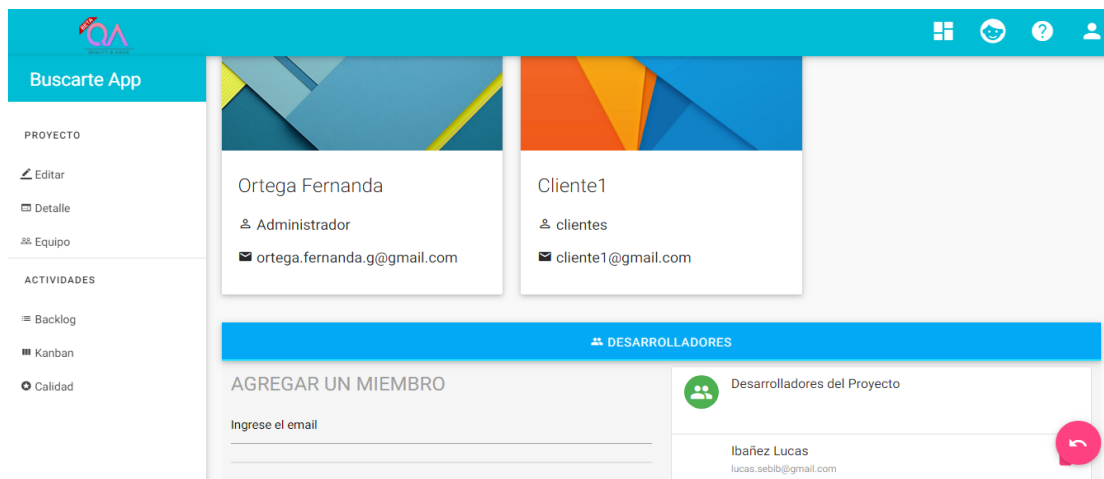


Figura D.2 Interfaz de QuAGI - Información del equipo
Fuente: de elaboración propia

b) *Gestión del Product Backlog*

Los requerimientos de cada proyecto se expresan en forma de Historias de Usuario, QuAGI permite la carga de cada una respetando el estilo que cada equipo prefiera¹ incluyendo la priorización de acuerdo al método MOSCOW [86] y la estimación teniendo en cuenta la técnica de Puntos de historia [87].

Además, como se observa en la Figura F.3, por cada historia de usuario se pueden añadir tareas de forma tal de facilitar la distribución de carga de trabajo en el equipo y el posterior seguimiento del avance del proyecto.



Figura D.3 Interfaz de QuAGI - Gestión de Historias de Usuario > Carga de Tareas

Fuente: de elaboración propia

Asimismo, QuAGI permite completar la información de las historias de usuario con los criterios de aceptación que defina el Product Owner, tal como se observa en la Figura F.4. De esta forma la gestión de requerimientos funcionales va acompañada también por la validación de condiciones, muchas veces identificados como requerimientos no funcionales, expresadas como criterios de aceptación.

¹ El estilo puede o no corresponderse con el template recomendado por la comunidad en la forma “Como [quien/rol] Quiero [que/acción] Para [razón/beneficio]”.

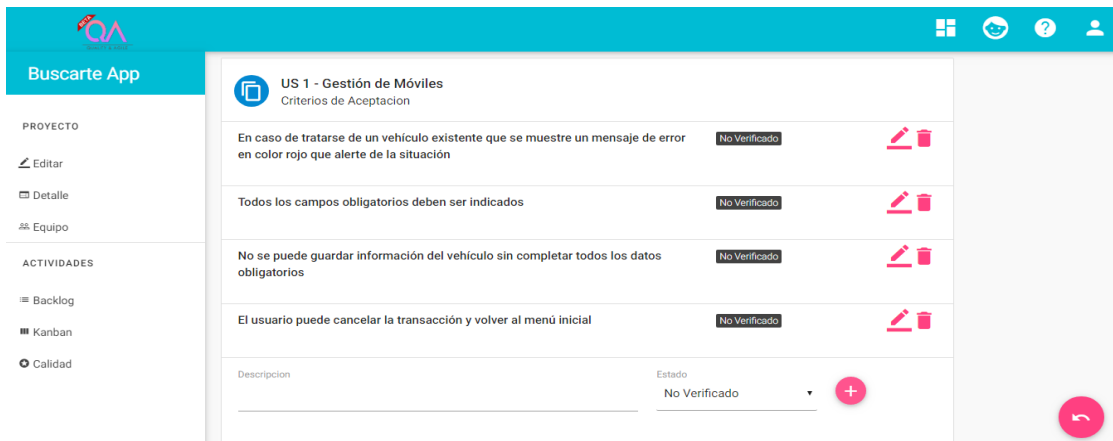


Figura D.4 Interfaz de QuAGI - Gestión de Historias de usuario > Criterios de Aceptación
Fuente: de elaboración propia

Cada historia de usuario admite también la carga de mockups que se hayan elaborado previamente y validado un prototipo inicial con el Product Owner. Tal como se muestra en la Figura F.5, las historias de usuario podrán ser asociadas a sus mockups generados previamente.

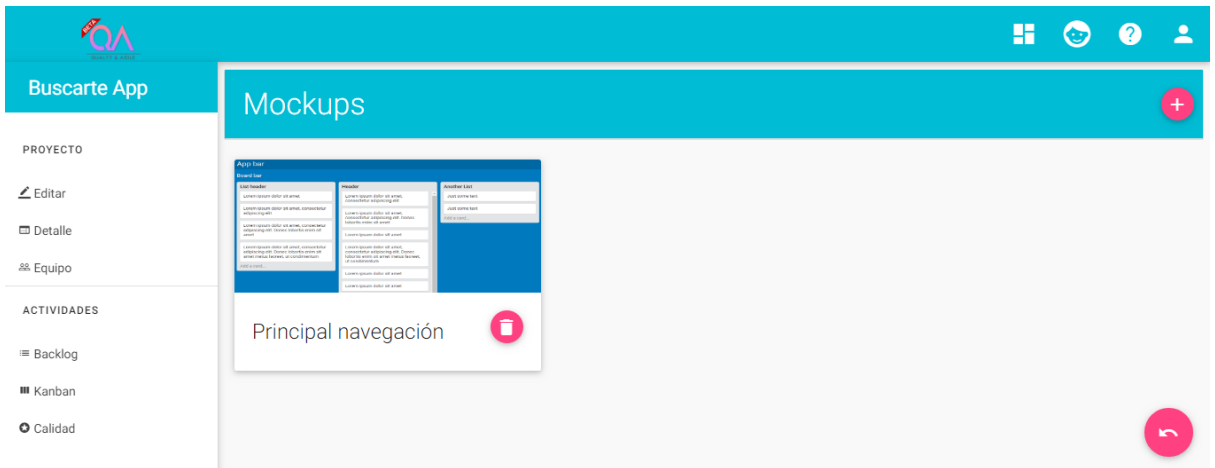


Figura D.5 Interfaz de QuAGI - Mockups de Historias de Usuario
Fuente: de elaboración propia

Una vez creado el Product Backlog, QuAGI permite, como se ve en la Figura F.6, la organización de las historias de usuario en Sprints Backlog facilitando la visualización del proyecto y la priorización.

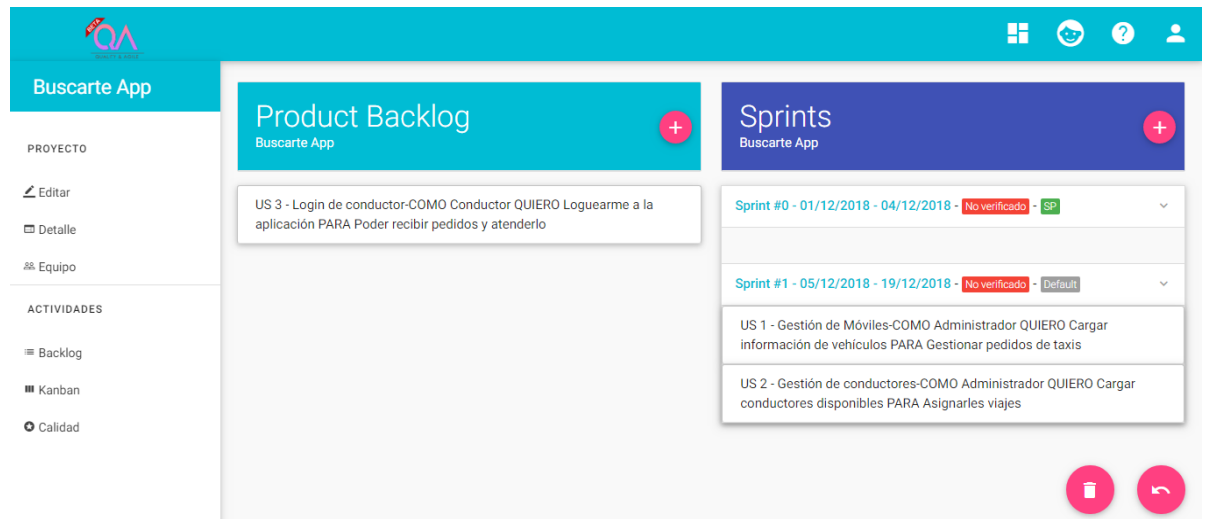


Figura D.6 Interfaz de QuAGI - Product Backlog y Sprint Backlog
Fuente: de elaboración propia

Asimismo, QuAGI permite que el equipo indique si existen dependencias entre historias de usuario dentro de un mismo sprint, a fin de poder gestionarlas más adelante. En la Figura F.7 se muestra un ejemplo en el que se observan dependencias entre historias de usuario del Sprint 1 del proyecto en cuestión.

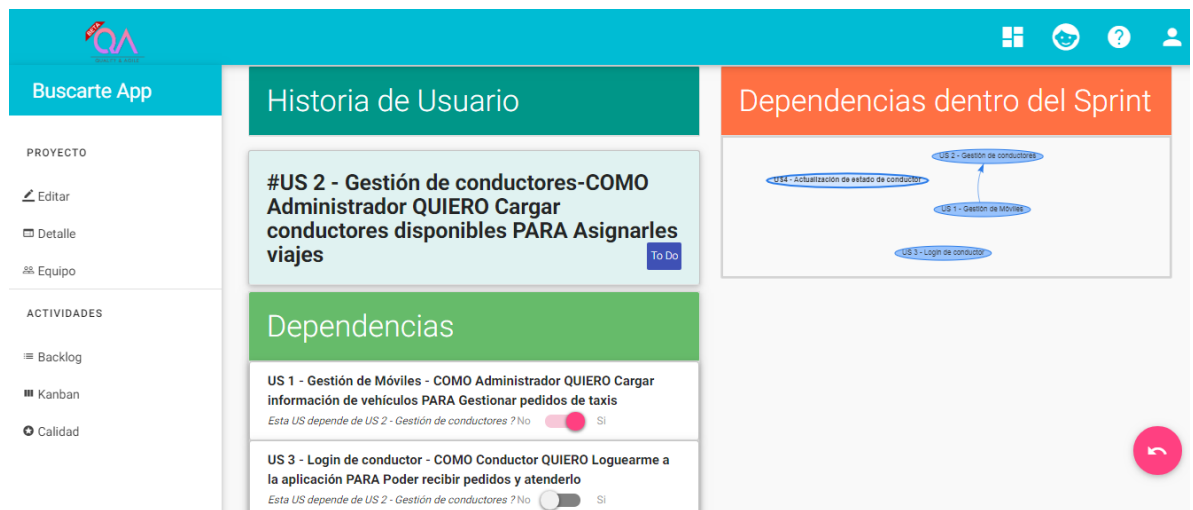


Figura D.7 Interfaz de QuAGI - Dependencias entre historias de usuario de un Sprint
Fuente: de elaboración propia

Otra herramienta que proporciona QuAGI al equipo ágil de desarrollo es la posibilidad de realizar el seguimiento del avance del proyecto a través del tablero Kanban [88]. Tal como se presenta en las Figura F.8, por un lado, el equipo tiene acceso al estado de las historias de usuario en el tablero Kanban.

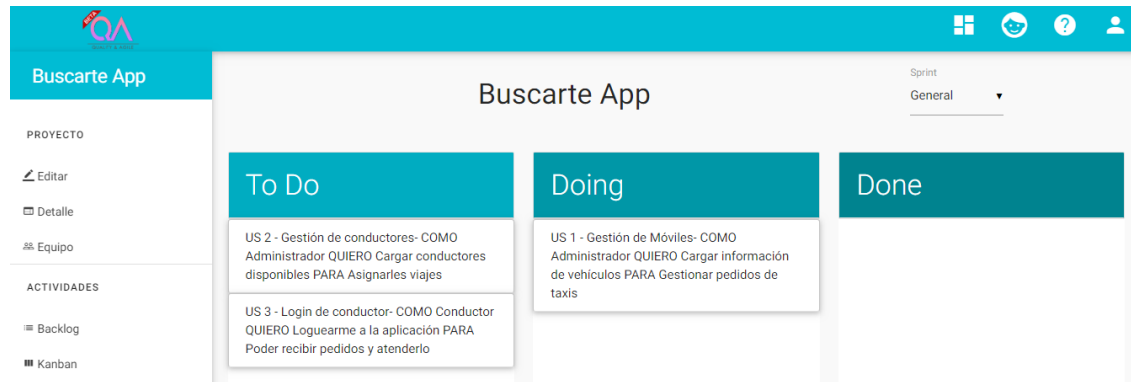


Figura D.8 Interfaz de QuAGI – Tablero Kanban para Historias de Usuario
Fuente: de elaboración propia

Y, por otro lado, como se ve en la Figura F.9, QuAGI permite seguir el avance de las tareas de cada historia de usuario por sprint también desde el tablero Kanban.

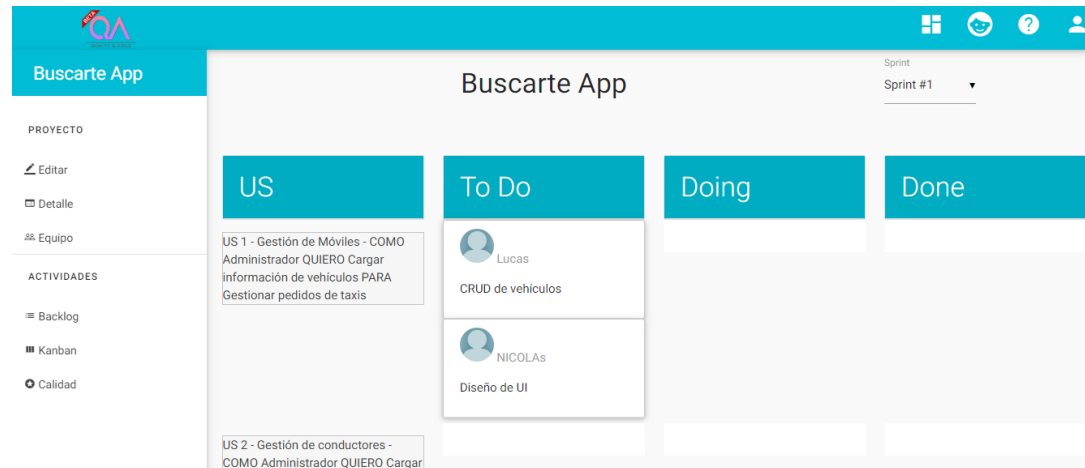


Figura D.9 Interfaz de QuAGI - Tablero Kanban de tareas por Historia de Usuario por Sprint
Fuente: de elaboración propia

c) Evaluación de Calidad

QuAGI aporta valor, al seguimiento automatizado de proyecto ágiles, a través de la evaluación de calidad simultánea que pueden realizar los Administradores.

Aquí se ponen de manifiesto los valores obtenidos por la medición de cada una de las métricas de los componentes de QuAM a partir de la actividad del equipo sobre el proyecto ágil involucrado.

En la Figura F.10 se presenta gráficamente la relación que define QuAGI para obtener los niveles de calidad por cada componente de QuAM y obtener entonces la

evaluación de calidad asociada al proyecto ágil. Por ejemplo, para evaluar el nivel de calidad del Componente 3 relacionado a la comunicación con el Cliente, QuAGI debe realizar el seguimiento de su participación en el proyecto a través del control de su actividad en la plataforma.



Figura D.10 Relación entre los componentes de QuAM y los elementos que observa QuAGI
Fuente: de elaboración propia

Al medir el flujo de trabajo y los elementos del proyecto ágil, QuAGI realiza los cálculos necesarios para asociar valores a las métricas definidas por cada componente y ya descritas en la sección 3.2 de esta tesis.

Para proporcionar una mejor comprensión de los niveles de calidad, se ha optado por presentar a los Administradores el informe de la evaluación de calidad en función a porcentajes de cumplimiento teniendo en cuenta el máximo valor posible en la escala establecida por QuAM. Se muestra en la Figura F.11 un caso del Informe de un determinado proyecto, en el que se utiliza diagrama de barras ubicando en el eje Y los componentes y en el eje X los valores segmentados desde el 0 al 100%. En la Figura F.10, por ejemplo, se han obtenido los siguientes niveles de calidad: para el Componente 1 se obtuvo el 50% de cumplimiento teniendo en cuenta el mejor valor posible, para el Componente 2 un 62.5% de cumplimiento, en el Componente 3 se tiene un 62.5% de cumplimiento y finalmente para el Componente 4 se obtuvo un 37.5% de cumplimiento sobre el mejor valor posible.

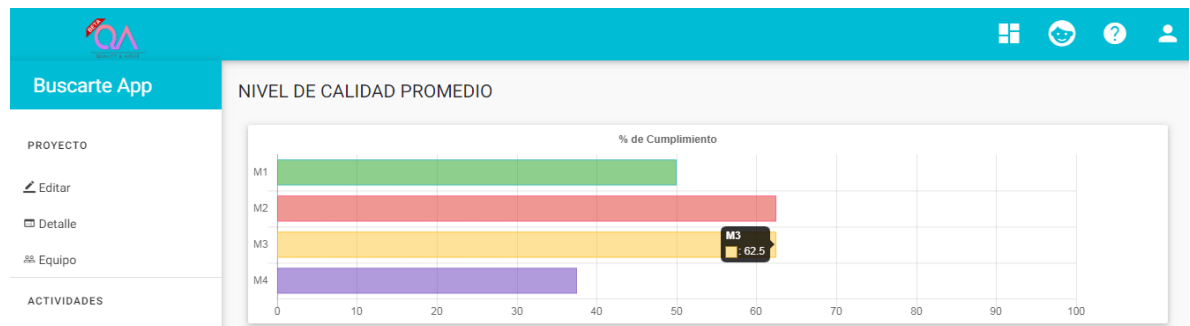


Figura D.11 Interfaz de QuAGI - Informe de Niveles de Calidad
Fuente: de elaboración propia

Resulta insuficiente para la toma de decisiones informar los niveles de calidad en función, solamente, a los porcentajes obtenidos en la evaluación de cada componente. Por ello, QuAGI extiende el gráfico estadístico incluyendo, además, por cada componente, el significado del nivel de calidad logrado de acuerdo con una escala que establece grados en función a los valores de calidad asociados a cada componente. En la Figura F.12 se representa la escala incluyendo los indicadores (Muy Malo, Mal, Bueno, Muy Bueno y Excelente) y la relación con el rango de valores posibles que puede tomar el Componente.

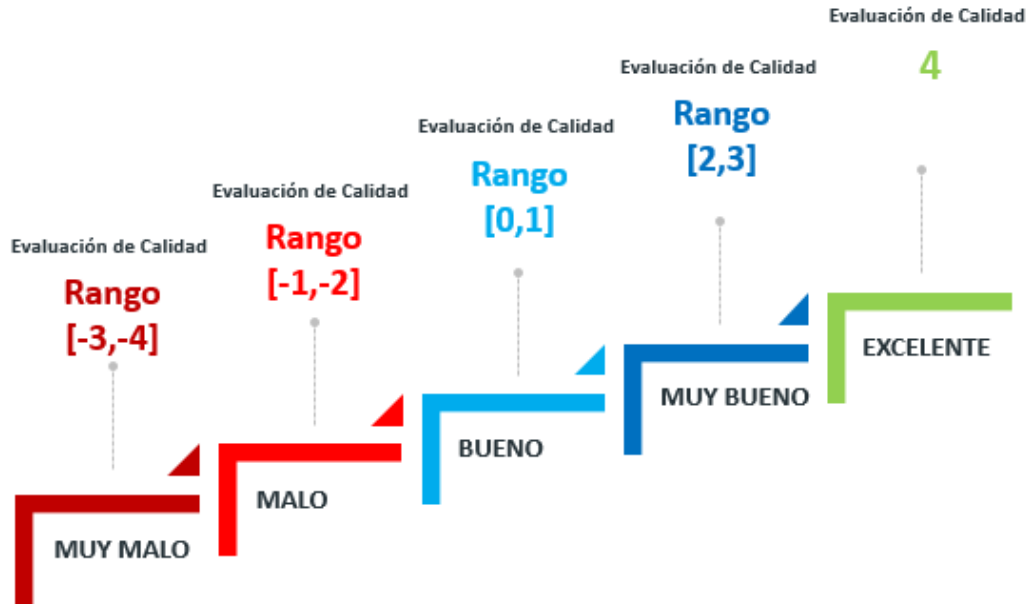


Figura D.12 Niveles de Calidad según el valor de evaluación obtenido
Fuente: de elaboración propia

Para más información, el resultado de la Evaluación de Calidad se muestra al Administrador, tal como se ve en la Figura F.13, indicando la relación entre atributos de cada componente y sus criterios correspondientes.



Figura D.13 Interfaz de QuAGI - Descomposición de la Evaluación de Calidad por componente
Fuente: de elaboración propia

En el ejemplo que se presenta en la Figura F.13, para el *Componente “Capacidad de Producción de Entregables”* se logró un nivel **BUENO** de calidad debido a que la medición del atributo positivo “*Valor a la estimación*” ha resultado en que todas las historias de usuario de los sprints tienen asignado puntos de historia (este criterio representa el mejor valor para el atributo). En el caso del atributo negativo “*Valor al lead time*”, se obtuvo el peor valor ya que de los sprints finalizados se encuentra validado menos del 10% de los criterios de aceptación. Combinando ambos valores se obtiene un nivel Bueno en el informe de evaluación de calidad del Componente 1.

d) Reportes varios

Por último, QuAGI posibilita la generación de reportes a fin de obtener información útil para la toma de decisiones con el fin de mejorar la calidad del proceso ágil correspondiente al proyecto. Así, y hasta el momento, se pueden obtener los siguientes reportes:

- *Nivel de cumplimiento de historias de usuario por sprint*
 Objetivo: Informar el estado del sprint en función al porcentaje de avance de las historias de usuario que lo componen.
- *Informe de avance de acuerdo con la priorización*

Objetivo: Identificar por sprint las historias de usuario de acuerdo con su priorización e informar el estado de ellas.

- *Velocidad del equipo*

Objetivo: Informar la velocidad del equipo en función a la cantidad de trabajo realizado por sprint.

- *Verificación de criterios de aceptación*

Objetivo: Informar acerca del nivel de cumplimiento de criterios de aceptación teniendo en cuenta la verificación por parte del Product Owner.

Se muestra en la Figura F.14 un ejemplo de reporte de Avances de Historias de Usuario. Allí se observa que, en este caso, se han finalizado todos los sprints del proyecto en cuestión. Sin embargo, el porcentaje de avance del sprint 1 es del 88%. Este informe permite al Administrador de Proyecto identificar, por ejemplo, la causa que impide lograr el 100% de cumplimiento del sprint, incluso habiendo finalizado. El informe muestra, como datos adicionales, el período asociado a cada sprint.

Además, se puede ver el estado de cada historia de usuario y su nivel de verificación de criterios de aceptación por cada una de ellas.

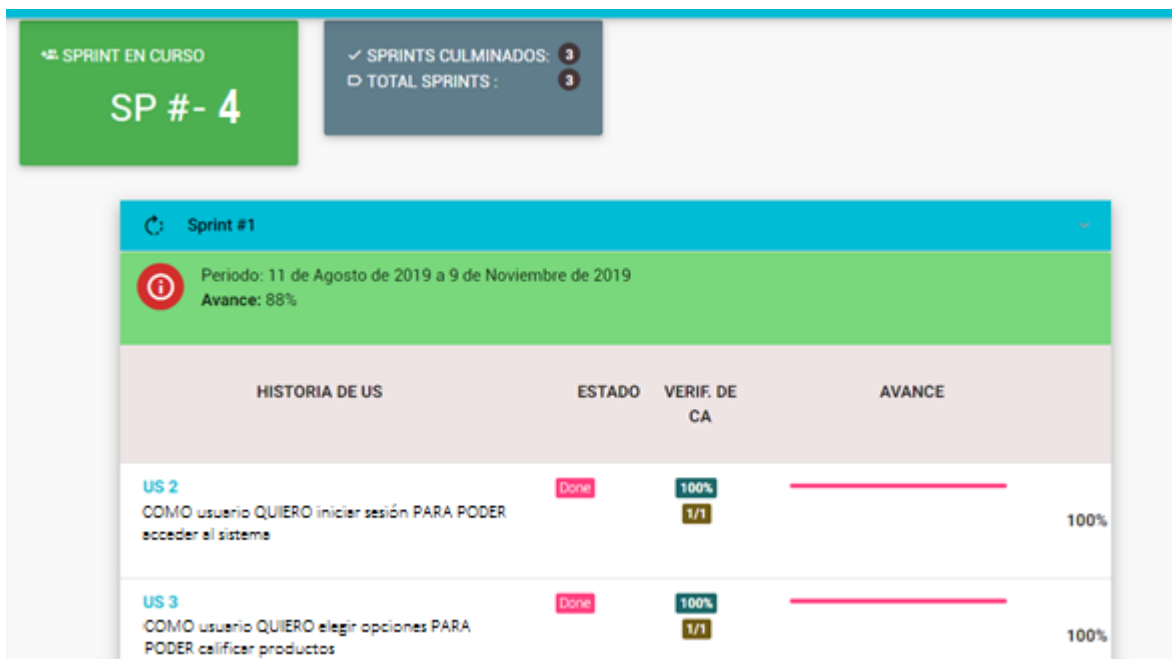


Figura D.14 Informe de avance de historias de usuario
Fuente: de elaboración propia

Aún resta mostrar cómo, teniendo en cuenta el diseño de los flujos de información y las interfaces generadas, se establecieron también los mecanismos que permitirían integrar la plataforma web a los requerimientos de medición definidos en QuAM. A continuación, se detalla el proceso de integración entre QuAM y QuAGI.

ANEXO E

Integración entre QuAM y QuAGI

Tanto el modelo conceptual QuAM como la herramienta web QuAGI trabajan sinérgicamente como parte del framework AQF, retroalimentándose en cada experiencia de uso de forma tal de lograr la mejor aproximación en la obtención de los niveles de calidad para los proyectos ágiles que se evalúen.

Un aspecto clave es la manera en que QuAGI implementa tecnológicamente todo lo dispuesto por QuAM. Para ello se ha dispuesto que:

- Cada uno de los componentes, descritos en la sección 3.2 de este capítulo, será implementado a través de un módulo lógico en la aplicación que encapsula todas las métricas y procesos de medición inherentes a él. En la Figura G.1 se presenta un ejemplo de la porción de código que se utiliza en QuAGI para la definición inicial de los componentes.

```
# 2do. Borramos todos los Atributos
atDelOk, atDelMal = del_obj_model(list_obj=Atributo.objects.all())

# 3ro. Borramos todas las Metricas
metDelOk, metDelMal = del_obj_model(list_obj=Metrica.objects.all())

# Creamos Componentes
c1 = Componente.objects.create( codigo = 1,
                               nombre = 'Capacidad de produccion de entregables',
                               descripcion = '')
c2 = Componente.objects.create( codigo = 2,
                               nombre = 'Evaluacion del Equipo de Trabajo',
                               descripcion = '')
c3 = Componente.objects.create( codigo = 3,
                               nombre = 'Comunicacion con el Cliente',
                               descripcion = '')
c4 = Componente.objects.create( codigo = 4,
                               nombre = 'Gestion de Requerimientos',
                               descripcion = '')

# Creamos Atributos
atr_M1P = Atributo.objects.create( codigo='A1.1', nombre = 'Valor a la estimación', descripcion='', met
atr_M1N = Atributo.objects.create( codigo='A1.2', nombre = 'Valorar el lead time por sobre la satisfacci
```

Figura E.1 Porción de código para la definición de componentes en QuAGI

Fuente: de elaboración propia

- Asimismo, en cada módulo se implementarán procesos que respondan a las métricas definidas en QuAM, de forma tal de organizar metodológicamente el código subyacente de QuAGI en base a ello. Esto puede observarse en el ejemplo que se muestra en la Figura G.2, en la cual se observa cómo el método *CalcularC1*, involucra la medición de los atributos correspondientes al componente C1 a través de las definiciones establecidas en las métricas asociadas en QuAM.


```

views.py
return render(request, 'informes/home.html', ctx)

def CalcularQi(proyecto, historias):
    atributo_positivo = Atributo.objects.get(codigo = 'A1.1')
    atributo_negativo = Atributo.objects.get(codigo = 'A1.2')

    #-----Atributo Positivo-----
    con_puntos = 0
    for us in historias:
        if us.puntos != 0:
            con_puntos+=1

    porcentaje_con_puntos = con_puntos/historias.count()

    criterio_cumplido_positivo = None
    if porcentaje_con_puntos == 1:
        criterio_cumplido_positivo = Criterio.objects.get(atributo = atributo_positivo, prioridad = 1)
    elif porcentaje_con_puntos >=0.8:
        criterio_cumplido_positivo = Criterio.objects.get(atributo = atributo_positivo, prioridad = 2)

```

Figura E.2 Porción de código en QuAGI para el cálculo de dos métricas
Fuente: de elaboración propia

De acuerdo con lo antes expuesto, para dar cumplimiento al diseño lógico de QuAGI y lograr su implementación se trabajó con una combinación de tecnologías que se describen a continuación.

Tecnologías utilizadas

A partir de la definición de la arquitectura conceptual de QuAGI se define el diseño técnico de la plataforma basado en el paradigma Modelo-Vista-Template (Model Template View o MTV) [89], cuya configuración permite que a medida que la complejidad va en aumento, se pueda estructurar el código para hacerlo más modular y facilitar así el desarrollo de nuevas herramientas individuales que se pudieran añadir al framework AQF en el futuro. La Figura G.3 resume gráficamente las tecnologías utilizadas para implementar las interfaces que permiten la interacción con usuarios, la infraestructura que da soporte a QuAGI y la base de datos que almacena la información que se genera a lo largo de cada proyecto.

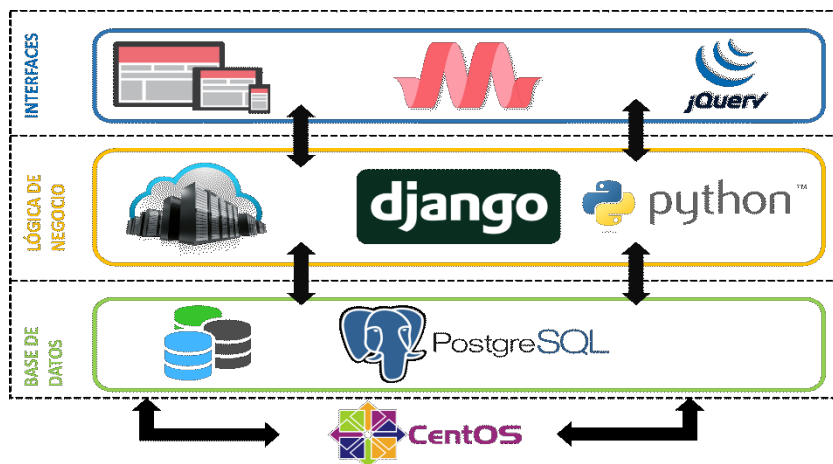


Figura E.3 Tecnologías utilizadas en cada capa de la Arquitectura conceptual
Fuente: de elaboración propia

QuAGI presenta un esquema cliente-servidor pesado¹ en el que la generación de la UI (User Interface o Interfaz de Usuario) y el control de las acciones del usuario recaen en el navegador del usuario, y la gestión de la información se realiza en el servidor.

La UI se implementa utilizando HTML, CSS y Javascript. El estilo de las interfaces de QuAGI se define en un archivo CSS, y debido a que se pretende generar una buena experiencia de usuario al interactuar con las funciones de la herramienta, se optó por implementar el front-end usando el framework CSS Materialize, el cual adopta las tendencias de diseño web planteadas por Google con un diseño sencillo y plano [90].

Toda la lógica de negocio y el modelo se ha implementado utilizando el framework DJANGO [91], esta elección se debe fundamentalmente a:

- La filosofía de re-utilización, conectividad y extensibilidad de componentes sumado al desarrollo rápido de aplicaciones web.
- La provisión de una API de base de datos robusta en sintonía con su filosofía que facilita la integración con otras bases de datos.
- La posibilidad de utilizar un sistema de templates como herramienta de apoyo para reducir el código de la capa de presentación y homogeneizarla, así como simplificar su funcionamiento.
- La provisión de mecanismos de seguridad y autenticación integrados.

Cabe destacar que debido a que el desarrollo de QuAGI se trató de un ciclo iterativo e incremental, primero se diseñó e implementó una plataforma inicial constituida por los componentes básicos y sobre estos se ha ido construyendo el resto de la arquitectura de control, de forma tal de utilizar, desde las primeras experiencias de validación, el seguimiento automatizado de los proyectos.

¹ Un cliente web pesado es aquel en que la carga de generación de la interfaz de usuario se realiza en el navegador del usuario

ANEXO F

Formulario de seguimiento

Para lograr resultados de comparación en la calidad de los procesos ágiles a partir del uso de herramientas de gestión de proyectos se utilizó una encuesta que fue respondida por los Administradores de Proyectos en 2 ocasiones (al finalizar el primer sprint, y al finalizar el proyecto).

El esquema de preguntas que comprendió la encuesta se presenta a continuación:

a) Primer Sección

- i) Equipo al que pertenece
- ii) Herramienta: QuAGI o Taiga

b) Segunda Sección

- i) Si usó QuAGI
 - 1) Qué niveles de calidad se obtuvo para cada componente
 - 2) En qué medida está de acuerdo con lo obtenido en la evaluación de calidad por cada componente
 - 3) Para finalizar se deja al encuestado que complete un campo con observaciones generales
- ii) Si uso Taiga
 - 1) En qué medida comprende lo definido en cada historia de usuario
 - 2) Cantidad de historias de usuario completamente generadas al momento (es decir que incluya tareas, responsables y criterios de aceptación)
 - 3) Cantidad de historias de usuario generadas de forma incompleta al momento (es decir que no incluya tareas, responsables y/o criterios de aceptación)
 - 4) Cantidad de historias de usuario estimadas
 - 5) Cantidad de historias de usuario priorizadas
 - 6) Existe dependencia de historias de usuario dentro de cada sprint
 - 7) Cantidad de historias de usuario que finalizaron y han sido validadas por el Product Owner
 - 8) Porcentaje de tareas que no tienen responsable asignado en el equipo
 - 9) Ha participado el Product Owner de los Sprints Planning al momento
 - 10) Ha participado el Product Owner de la definición de historias de usuario

c) Tercer Sección

Cierre con comentarios finales

ANEXO G

Encuesta de Usabilidad

La encuesta fue conformada por las 8 de las 10 preguntas recomendadas por Brooke en [85] junto a 2 más incorporadas a los fines de esta tesis. A continuación, se presenta entonces la encuesta utilizada:

1. Creo que me gustaría utilizar este sistema frecuentemente.
2. El sistema me resultó innecesariamente complejo.
3. Creo que el sistema es bastante fácil de utilizar.
4. Creo que necesitaría el soporte de un técnico para poder utilizar este sistema.
5. Creo que las diferentes funciones del sistema se encuentran muy bien integradas.
6. Opino que hubo demasiada inconsistencia en el sistema.
7. Me sentí muy seguro al utilizar este sistema.
8. Necesito aprender muchas otras cosas antes de poder utilizar correctamente el sistema.
9. El sistema ha contribuido en mejorar la calidad del proceso ágil.
10. El sistema sirve como instrumento de soporte para la evaluación de calidad del proceso ágil.

El resultado de la encuesta que utiliza la escala de usabilidad del sistema se llama puntaje SUS. Este puntaje está numerado del 0 al 100, siendo el 0 el puntaje mínimo y el 100 el puntaje máximo. Solo se toma en cuenta una pregunta a la vez al momento de la evaluación. Hay 5 respuestas disponibles cada vez que un encuestado responde una pregunta individual.

El algoritmo para la obtención de la medición final consta de los siguientes pasos:

Paso 1: Convertir la escala a números

- Totalmente en desacuerdo: 1 punto
- En desacuerdo: 2 puntos
- Neutral: 3 puntos
- De acuerdo: 4 puntos
- Totalmente de acuerdo: 5 puntos

Paso 2: Realizar los cálculos correspondientes

1. Considerar el valor de X a partir de restar 5 a la suma de todos los puntos obtenidos en preguntas impares.
2. Luego calcular Y a partir de restar 25 menos la suma de puntos de todas las preguntas pares.
3. Obtener el valor final >> Puntuación SUS = $(X + Y) * 2.5$

Por ejemplo, supongamos que la suma de los puntos de preguntas impares es 20.
Entonces, $X = 20 - 5 = 15$

Ahora, supongamos que la suma de los puntos de todas las preguntas con número par es 25.
Entonces, $Y = 25 - 25 = 0$

$$\text{Puntuación SUS} = (15 + 0) * 2.5 = 37.5$$

Esta puntuación se expresa en porcentaje. Cabe mencionar que, de acuerdo a lo establecido por Brooke, si el resultado final de SUS tiene un valor de 100%, la usabilidad del sistema es *perfecto*. Los valores por debajo del 100% y por encima del 50% son considerados *buenos* y se cree que cualquier valor inferior al 50% es *muy ineficiente* en términos de usabilidad.

Referencias

1. Cámara de Software y Servicios Informáticos - CESSI. Anuario de la Industria Argentina de TI 2007/2008, http://www.cessi.org.ar/argentina/anuario_2007-2008.php
2. Ministerio de Producción de la Nación, Subsecretaría de Servicios Tecnológicos y Productivos, Observatorio de la Economía del Conocimiento, Informe de Servicios Basados en el Conocimiento, Octubre de 2017, <https://www.produccion.gob.ar/wp-content/uploads/2017/10/IOEC4-informe.pdf>
3. Reporte anual 2018 sobre el Sector de Software y Servicios Informáticos de la República Argentina. OPSSI, 2018. Disponible en <https://www.cessi.org.ar/opssi>
4. Navarro, J. M., & Garzías, 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
5. 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
6. 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.
7. Pflegger, S. (2002) "Ingeniería de Software. Teoría y Práctica." Pearson Education.
8. Humphrey, W.S (2005), "Acquiring Quality Software", CrossTalk, The Journal of Defense Software Engineering. Vol. 18, N°.12, December 2005.
9. Letelier, P., Penadés, P. (2006) "Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)" Técnica Administrativa, Buenos Aires. ISSN 1666-1680
10. Alliance, A. (2001). "Agile manifiesto". Disponible en <http://www.agilemanifesto.org>
11. ISO (2001). Sistemas de gestión de la calidad-Conceptos y vocabulario. Norma Internacional ISO 9000
12. Humphrey W. S. (1989). Managing the Software Process. New York: Addison-Wesley.
13. ISO/IEC 25010. (2011). Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models.
14. Papa, M. F. Aseguramiento de la calidad de un recurso organizacional: evaluando y mejorando una estrategia integrada de medición y evaluación. Tesis Doctoral, UNLP Facultad de Informática, 2015
15. IEEE Standard 610. Institute of Electrical and Electronics Engineers Computer dictionary. Compilation of IEEE Standard Computer Glossaries. 1990
16. Pressman, R. S. "Ingeniería de Software. Un enfoque práctico". Editorial MCGRAW-HILL. 2005
17. Ian Sommerville. "Ingeniería del Software". 7º Ed. Cap. 27. Pearson Educación S.A., Madrid 2005.
18. Constanzo, M. A., Casas, S. I., & Marcos, C. A. (2014). Comparación de modelos de calidad, factores y métricas. Informes Científicos-Técnicos UNPA,6(1), 1-36.
19. Olsina, L., Bertoa, M. F., Lafuente, G., Martín, M. A., Matrib, M., & Vallecillo, A. (2002, November). Un Marco Conceptual para la Definición y Explotación de Métricas de Calidad. In JISBD (pp. 189-200)
20. Mc Call, J.A., Richards, P.K., Walters, G.F.: Factors in Software Quality (1977). Vol. I, II, III: Final Technical Report, RADC-TR-77369. Rome Air Development Center, Air Force System Command, Griffith Air Force Base, Nueva York.

21. Swebook, [Online] Disponible en: <http://www.computer.org/portal/web/swebook>
22. Version One 12th Annual State of Agile™ Report (2018) Available at <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>
23. 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.
24. Marcos, E., & Marcos, A. (1999). An Aristotelian Approach to the Methodological Research: a Method for Data Model Construction. Information Systems. The Next Generation. McGraw Hill, 532-543.
25. Bunge, M. (1989). La investigación científica (No. 330.018 BUN).
26. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. Experimentation in software engineering: an introduction. Kluwer Academic Publishers, Norwell, MA, USA, 2000.
27. Fink, A. (2003). Reliable and valid survey instruments. The Survey Handbook. Thousand Oaks: SAGE Publications, Inc.
28. Kitchenham, B., Pickard, L., & Pfleeger, S. L. (1995). Case studies for method and tool evaluation. IEEE software, 12(4), 52-62.
29. Klein HK, Myers MD (1999) A set of principles for conducting and evaluating interpretative field studies in information systems. MIS Q 23(1):67–88 doi:10.2307/249410
30. Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. Empirical software engineering, 14(2), 131.
31. Runeson, P., & Höst, M. (2008). Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, 14(2), 131–164. doi:10.1007/s10664-008-9102-8
32. Kitchenham, B. (2004). Procedures for performing systematic reviews. Keele, UK, Keele University, 33(2004), 1-26.
33. Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., & Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. IEEE Transactions on software engineering, 28(8), 721-734.
34. Kitchenham B.A., Budgen D. and P. Brereton The value of mapping studies - A participant - observer case study [Conference] // Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering (EASE'10). - UK : [s.n.], 2010.
35. Kitchenham Barbara A., Budgen David and Brereton Pearl Using Mapping studies as the basis for further research - A participant - case study [Journal] // Information and Software Technology. - 2011. - 53. - pp. 638-651.
36. Kitchenham B. A. and Charters S. Guidelines for performing systematic literature reviews in software engineering [Report] / School of Computer Science and Mathematics; Keely University. - UK : Technical Report EBSE-2007-001, 2007.
37. Noruzi, A. (2005) Google Scholar: The New Generation of Citation Indexes. Libri, 55, 170-180. <http://dx.doi.org/10.1515/LIBR.2005.170>
38. Noruzi, A. (2005) Google Scholar: The New Generation of Citation Indexes. Libri, 55, 170-180. <http://dx.doi.org/10.1515/LIBR.2005.170>
39. Petticrew M, Helen R. Systematic reviews in the social sciences: A practical guide. Blackwell Pub; 2006.
40. García, M. D. A., & Navarro, V. M. (2011). La literatura gris en entornos digitales: estrategias de calidad y evaluación. Revista Interamericana de Bibliotecología, 27(2)
41. Cascaes da Silva, F., Valdivia Arancibia, B. A., da Rosa Iop, R., Gutierrez Filho, P. J. B., & da Silva, R. (2013). Escalas y listas de evaluación de la calidad de estudios científicos. Revista Cubana de Información en Ciencias de la Salud (ACIMED), 24(3), 295-312.

42. Kitchenham B. A. and Charters S. Guidelines for performing systematic literature reviews in software engineering [Report] / School of Computer Science and Mathematics ; Keely University. - UK : Technical Report EBSE-2007-001, 2007.
43. Medeiros, J., Vasconcelos, A., Silva, C., & Goulão, M. (2018). Quality of software requirements specification in agile projects: A cross-case analysis of six companies. *Journal of Systems and Software*, 142, 171-194
44. Guzmán, L., Vollmer, A. M., Ciolkowski, M., & Gillmann, M. (2017, November). Formative evaluation of a tool for managing software quality. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (pp. 297-306). IEEE Press
45. de Freitas, R. C., Rodrigues, L. A., & da Cunha, A. M. (2016, July). AGILUS: A Method for Integrating Usability Evaluations on Agile Software Development. In *International Conference on Human-Computer Interaction* (pp. 545-552). Springer, Cham
46. Aslam, W., & Ijaz, F. (2018). A Quantitative Framework for Task Allocation in Distributed Agile Software Development. *IEEE Access*, 6, 15380-15390
47. Hotomski, S., & Glinz, M. (2018, May). GuideGen: a tool for keeping requirements and acceptance tests aligned. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings* (pp. 49-52). ACM.
48. Jeffries, R., et al.: User interface evaluation in the real world: a comparison of four techniques. In: *SIGCHI Conference on Human Factors in Computing Systems, Proceedings*, pp. 119–124. ACM, New York (1991)
49. Perkusich, M., Gorgônio, K. C., Almeida, H., & Perkusich, A. (2017). Assisting the continuous improvement of scrum projects using metrics and bayesian networks. *Journal of Software: Evolution and Process*, 29(6), e1835
50. Benbasat Izak, Goldstein David K. and Mead Melissa The Case Research Strategy in Studies of Information [Journal] // *MIS Quarterly*. - 1987. - 3 : Vol. 11. - pp. 369-386.
51. Noelia Pinto; Gabriela Tomaselli; César J. Acuña; Liliana Cuenca Pletsch; Nicolás Tortosa. Hacia un modelo de evaluación de calidad de procesos ágiles. *CONAIIISI 2016*. ISSN 2347-0372
52. McNamara, C. (1999). General guidelines for conducting interviews.
53. Bernal Pablo, P. (2018). *La Investigación en Ciencias Sociales: Técnicas de recolección de la información*. Universidad Piloto de Colombia.
54. Pinto, N., Acuña, C., & Cuenca Pletsch, L. R. (2016). Quality Evaluation in Agile Process: A First Approach. In *XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016)*...
55. Cuenca Pletsch, L., Acuña, C., Tortosa, N., Tomaselli, G., & Pinto, N. (2015). Vinculación Universidad-Industria-Estado: Una experiencia en la Industria del Software del NEA.
56. Merchán Rodríguez, V. R. (2017). *Evaluación de la calidad de gobierno de tecnologías y sistemas de información basada en valor* (Doctoral dissertation, Facultad de Informática)
57. Otzen, T., & Manterola, C. (2017). Técnicas de Muestreo sobre una Población a Estudio. *International Journal of Morphology*, 35(1), 227-232.
58. Olsina, L., & Rossi, G. (2002). Measuring Web application quality with WebQEM. *Ieee Multimedia*, 9(4), 20-29.
59. Olsina, L., Papa, F., & Molina, H. (2008). How to measure and evaluate web applications in a consistent way. In *Web Engineering: Modelling and Implementing Web Applications* (pp. 385-420). Springer, London.
60. 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

61. Olsina, L., & Martin, M. D. L. A. (2004). Ontology for software metrics and indicators. *J. Web Eng.*, 2(4), 262-281.
62. Becker, P., Papa, M. F., & Olsina, L. (2014). Process Conceptual Base for Enriching a Measurement and Evaluation Ontology. *CIBSE*, 14, 53-66.
63. Olsina, L., Garrido, A., Rossi, G., Distante, D., y Canfora, G. (2008a). Web Application Evaluation and Refactoring: A Quality-Oriented Improvement Approach. *Journal of Web Engineering*, 7(4):258–280, ISSN: 1540-9589.
64. Beck, K., & Boehm, B. (2003). Agility through discipline: A debate. *Computer*, 36(6), 44-46.
65. 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.
66. Mason-Jones, R., & Towill, D. R. (1999). Total cycle time compression and the agile supply chain. *International journal of production economics*, 62(1-2), 61-73.
67. Vadillo, M. T. P. (2013). Liderazgo y motivación de equipos de trabajo. ESIC Editorial.
68. Manyoma Velásquez, P. C., & Alarcón Henao, A. M. (2013). Manufactura ágil: aclaraciones y confusiones.
69. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). *Manifiesto for agile software development*.
70. Beck, K., & Gamma, E. (2000). *Extreme programming explained: embrace change*. Addison-Wesley Professional
71. Anderson, D. J. (2003). *Agile management for software engineering: Applying the theory of constraints for business results*. Prentice Hall Professional.
72. Cao L, Ramesh B (2008) Agile requirements engineering practices: an empirical study. *Software* 25(1):60–67
73. Wake, B.: INVEST in Good Stories, and SMART Tasks (2003). <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
74. Hendrickson, E. (2008). Driving development with tests: ATDD and TDD. STARWest 2008.
75. Waters, K. (2009). Prioritization using moscow. *Agile Planning*, 12, 31.
76. Pinto, N., Tortosa, N., Cabas Geat, B., Ibáñez, L., & Acuña, C. (2018, November). Validación de la reingeniería aplicada sobre la primera versión de Agile Quality Framework. In XIX Simposio Argentino de Ingeniería de Software (ASSE)-JAIIO 47 (CABA, 2018).
77. C. Abras, D. Maloney-Krichmar, J. Preece, "User-centered design," Bainbridge, W. *Encyclopedia of Human-Computer Interaction*. Thousand Oaks: Sage Publications, vol. 37, pp. 445-456, 2004
78. Parnas, D. L. (1972). On the Criteria To Be Used in Decomposing Systems into Modules. *Communications of the ACM*, 15(12), 1053-1058.
79. Kitchenham, B. A., Dyba, T., & Jorgensen, M. (2004, May). Evidence-based software engineering. In *Proceedings. 26th International Conference on Software Engineering* (pp. 273-281). IEEE.
80. Cuenca Pletsch, L., Acuña, C. J., Tomaselli, G., & Pinto, N. (2017). QUAGI: Una propuesta para el seguimiento y evaluación de proyectos de Software Ágiles. V SABTIC, VIII STIN e XVIII Fórum.
81. Pinto, N., Tortosa, N., Cabas Geat, B., Ibáñez, L., & Acuña, C. (2018, November). Validación de la reingeniería aplicada sobre la primera versión de Agile Quality Framework. In XIX Simposio Argentino de Ingeniería de Software (ASSE)-JAIIO 47 (CABA, 2018).
82. Nemoto, T., & Beglar, D. (2014). Likert-scale questionnaires. In *JALT 2013 Conference Proceedings* (pp. 1-8).

83. Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2), 131.
84. Romero, G. R., Pinto, N., & Burgos Boero, M. (2015). Informatario Chaco: una experiencia de educación no formal para la Knowmad Society de TICs. In X Congreso sobre Tecnología en Educación & Educación en Tecnología (TE & ET)(Corrientes, 2015).
85. Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.
86. Waters, K. (2009). Prioritization using moscow. *Agile Planning*, 12, 31.
87. Peter Hill, 2010 “Practical Software Project Estimation- A Toolkit for Estimating Software Development Effort and Duration” Mc Graw Hill Education
88. Hammarberg, M., & Sunden, J. (2014). *Kanban in action*. Manning Publications Co..
89. Información disponible en <https://docs.djangoproject.com/en/1.10/>
90. Prabhu, A., & Shenoy, A. (2016). *Introducing Materialize*. In *Introducing Materialize* (pp. 1-9). Apress, Berkeley, CA.
91. Rubio, D. (2017). *Django Model Forms and Class Views*. In *Beginning Django* (pp. 403-439). Apress, Berkeley, CA