

## HERRAMIENTAS Y BUENAS PRÁCTICAS PARA EL ASEGURAMIENTO DE CALIDAD DE SOFTWARE CON METODOLOGÍAS ÁGILES

### TOOLS AND GOOD PRACTICES FOR SOFTWARE QUALITY INSURANCE UNDER SKILFUL METHODOLOGIES

Víctor Hugo Mercado-Ramos<sup>1</sup>

Julián Zapata<sup>2</sup>

Yony Fernando Ceballos<sup>3</sup>

Recibido: febrero 26 de 2015

Aceptado: mayo 19 de 2015

#### Resumen

La gestión de la calidad en aquellas empresas que se dedican al desarrollo de proyectos de software, ofrece una ventaja competitiva puesto que de esta forma aseguran que sus productos sean de calidad. Las metodologías ágiles surgen para aplicarse principalmente en proyectos cuya naturaleza sea cambiante o donde se hace necesario obtener pronto retorno de la inversión. Las organizaciones que optan por el cambio hacia las metodologías ágiles enfrentan retos como el aseguramiento de la calidad tanto del producto como del proyecto. Modelos como Capability Maturity Model Integration e ISO 25010, son conocidos y funcionan bien en entornos de desarrollo tradicionales. Para metodologías ágiles hay algunas propuestas menos populares, entre ellas se puede mencionar el Desarrollo Dirigido por Pruebas, la Integración Continua y la métrica Product Backlog Rating. Éstas, aunque en la literatura reportan buenos resultados, no están lo suficientemente documentadas para ser empleadas en cualquier proyecto ágil. Este trabajo hace un análisis sobre las propuestas que apuntan a mejorar la calidad en los productos realizados con metodologías ágiles, recopilando herramientas y buenas prácticas con los que se pretende diseñar una serie de recomendaciones en cuanto a gestión de calidad en los proyectos de software.

**Palabras clave:** calidad de software, administración de proyectos, desarrollo ágil de software, metodologías ágiles.

#### Abstract

The quality management in companies engaged in software development projects, provides a competitive advantage by guaranteeing that their products are high quality made. Skilful methodologies emerge for the application in projects whose nature is changing or where it is necessary to get a quick investment return. Organizations that choose to shift to skilful methodologies face challenges like ensuring the quality not only of the product but the product and the project. Models such as the Capability Maturity Model Integration and ISO 25010, are known and work well in traditional development environments. For skilful methodologies, there are some less popular proposals, including the Test Driven Development, Continuous Integration and Product Backlog Rating metrics. Nevertheless, although the literature reported good results, these are not sufficiently documented to be used in any skilful project. This paper analyzes the proposals aimed at improving the quality of products made with skilful methodologies, tools and best practices, and gives a series of recommendations for quality management in software projects.

**Keywords:** software quality, project management, agile software development, skilful methodologies.

<sup>1</sup> Ingeniero de Sistemas, Especialista en Ingeniería de Software, Profesor Instituto Tecnológico Metropolitano, Colombia.  
E-mail: victorh1mercado@gmail.com

<sup>2</sup> Ingeniero de Control, Especialista en Ingeniería de Software. Ingeniero de Desarrollo - Tech and Solve S.A.S., Colombia.  
E-mail: jjzapata2004@gmail.com

<sup>3</sup> Ingeniero de Sistemas e Informática, Doctor en Ingeniería, Profesor Universidad de Antioquia, Colombia.  
E-mail: yony.ceballos@udea.edu.co

## 1. Introducción

Las metodologías ágiles han tomado popularidad en los últimos años entre las organizaciones que están involucradas en desarrollo de software debido a características como (Fitzgerald, Stol, O'Sullivan, & O'Brien, 2013) *co-located teams developing non-critical systems. The first two constraining characteristics (small and co-located teams. La flexibilidad del desarrollo de proyectos y su gestión de forma ágil, las entregas tempranas de producto funcional que el cliente puede empezar a rentabilizar rápidamente, lo cual permite retorno temprano de la inversión (Schuh, 2007). Las propuestas de los principios fundamentales de las metodologías ágiles actuales (Buonomico, 2013) se enfocan en este tipo de ideas y tomaron fuerza en la década del 2000 con la publicación del manifiesto ágil (Beck et al., 2001).*

A pesar de las ventajas que ofrece el desarrollo de software bajo un enfoque ágil, existen aspectos relacionados con el aseguramiento de la calidad que aún deben ser tratados. Métodos como el Test Driven Development (TDD) y la Integración Continua, son empleados para tal fin; sin embargo, su uso no garantiza un desarrollo con calidad, ya que cada proyecto posee diferentes particularidades. Factores como la experiencia del equipo en desarrollo con TDD, la cercanía con el cliente y la disposición de éste para generar los casos de prueba, son, entre otros, elementos determinantes a la hora de entregar un producto con calidad.

Otro elemento que genera controversia, son los estándares que exigen una rigurosa documentación, los cuales permiten controlar los procesos que impactan la calidad del software. Es conocido que la documentación excesiva genera rechazo y dificultades en la eficacia de procesos en proyectos ágiles, no obstante, prescindir completamente de ella puede ser contraproducente, ya que facilita la historia y las métricas tomadas para hacer seguimiento tanto del proyecto como del producto (Ktata & Lévesque, 2010).

En las metodologías ágiles no se encuentra documentado de manera explícita cuáles son las herramientas a usar para asegurar la calidad del producto. Sin embargo, se han aplicado de manera exitosa algunas prácticas, las cuáles serán el objeto de análisis del presente trabajo. A continuación se presenta la aplicación de un protocolo de revisión sistemática para ingeniería de software (Biolchini et al., 2005). Para ello se hará una revisión de información científica que trate sobre la gestión de la calidad para proyectos ágiles de desarrollo de software. La información recolectada a través del estudio y análisis realizados sobre los artículos seleccionados, se consolidará para determinar la manera de implementar elementos de calidad en entornos de trabajo ágil.

La primera parte de este documento presenta la estrategia empleada para la selección de los artículos, posteriormente se establece una discusión de los mismos, donde se hará una revisión estadística para identificar tendencias o patrones. Finalmente se presentará la propuesta de solución, basada en la información ya obtenida. Al final del documento se encuentran las conclusiones, en las cuales se destaca que no existe un estándar que se pueda utilizar para control en la calidad, pero si una serie de recomendaciones, que pueden ser adoptadas, según sea el caso particular o la necesidad del proyecto.

## 2. Metodología

### 2.1 Búsqueda de información

Para la selección de artículos se hizo una búsqueda en las bases de datos científicas ACM, ScienceDirect y EBSCOHost, así como en Google Scholar. Se definieron diferentes palabras claves, estableciendo las cadenas de búsqueda *"Quality on software projects with agile methodologies"* y *"Methodologies for software projects quality"*. Posteriormente fue necesario adicionar a la cadena de búsqueda filtros excluyentes así: el primer filtro limita el tipo de publicación a los últimos 10 años, el segundo a artículos de revista y el tercero a palabras claves

como “project management, SCRUM, Agile y Software”, definiendo específicamente los criterios de exclusión. La realización de este procedimiento

fue iterativa, recuperando la información que se observa en la tabla 1.

Fuente de Datos	Cantidad encontrada de artículos			
	Sin filtro	Primer Filtro	Segundo filtro	Tercer filtro
ACM	3.662	3.411	1.530	5
Google Scholar	34.700	16.500	16.400	56
EBSCHost	176	151	2	1
ScienceDirect	2.246	1.919	1.900	0
Total	40.784	21.981	19.832	62

Tabla 1. Estadísticas de primer resultado de búsqueda

Debido al número de resultados fue necesario adicionar a la cadena de búsqueda algunos filtros tales como limitar el tipo de publicación a artículos y fecha de publicación superior a los últimos 10 años, emplear algoritmos de búsqueda usando las herramientas avanzadas de las bases de datos

(el uso de los conectores lógicos “AND”, “OR”). Por último, se diseñó un tercer filtro en el cuál se limitó a buscar las palabras del algoritmo de búsqueda, únicamente en el título del artículo. La secuencia lógica para la realización de este proceso de revisión se ilustra en la figura 1.

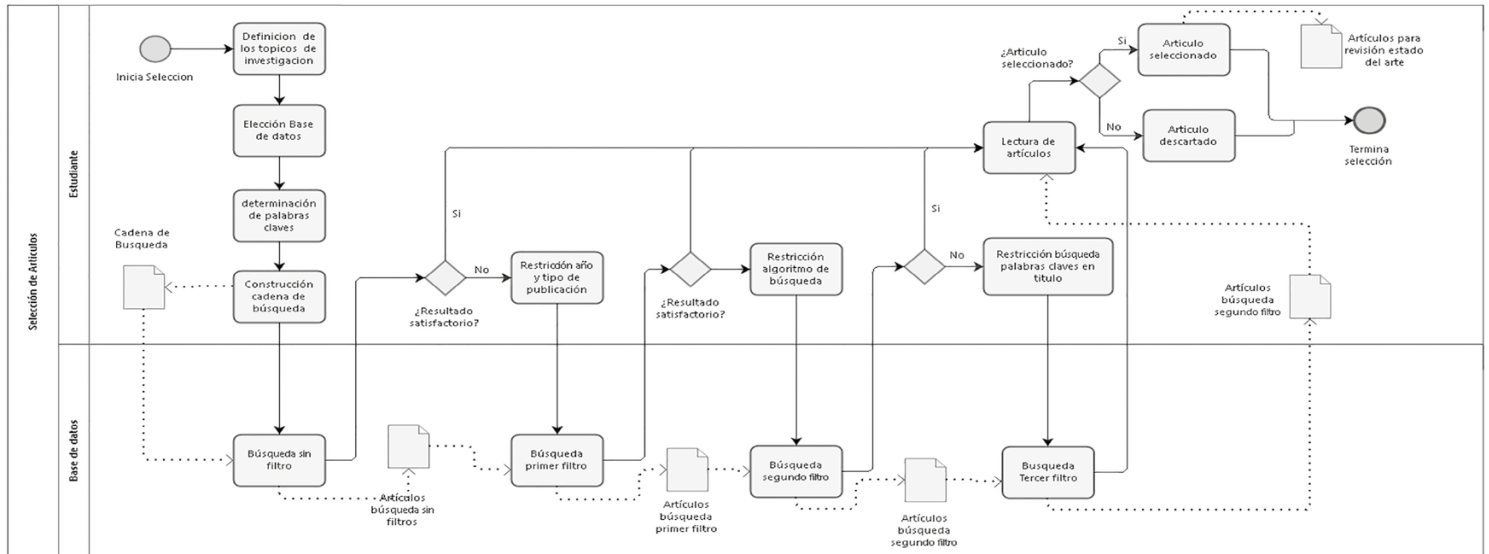


Figura 1. Diagrama del proceso de selección de artículos

## 2.2 Preguntas de investigación y criterios de selección

Las siguientes fueron las preguntas orientadoras del proceso de búsqueda de información: i) Cuáles son las herramientas y buenas prácticas empleadas para garantizar la calidad en proyectos de software gestionados mediante metodologías ágiles?, ii) Existen frameworks para la gestión de la calidad en proyectos de software que se puedan aplicar a metodologías ágiles?, iii) Las metodologías para el control de calidad utilizadas por los autores aplican para todo tipo de proyectos ágiles?.

El tema de la calidad en el desarrollo de proyectos de software es muy amplio. Para limitar el alcance de la revisión, se optó por buscar artículos en los cuáles se realizaran propuestas de modelos de calidad para proyectos ágiles, que bien podrían ser nuevas o, simplemente, adaptaciones de modelos actuales que se aplican sobre metodologías de desarrollo tradicional. Otro criterio importante de inclusión fueron los artículos que mencionaran el tema de estandarización de procesos en las metodologías ágiles, ya que esto puede estar muy ligado al tema principal de la revisión.

Las principales estrategias empleadas en la exclusión de documentos fueron: bases de datos: la consulta se realizó principalmente en las bases de datos ACM, Google Scholar, EBSCOHost y ScienceDirect, tipo de publicación: se limitó la búsqueda a únicamente artículos científicos formales, con una estructura de referencias significativa y que presenten casos de aplicación de las metodologías, tiempo: se hizo una delimitación de tiempo de los últimos 10 años, por lo cual se descartaron artículos cuya fecha de publicación fuese menor a 2004. Palabras clave: se realizó una cuidadosa selección de palabras clave, como resultado de esta selección se obtuvo "quality on software projects with agile methodologies". Descriptores de búsqueda: (agile AND development) OR software OR (software AND quality).

Los siguientes fueron los criterios empleados para la evaluación de los artículos seleccionados: **C1**. El artículo propone una nueva metodología para la gestión y control de calidad en metodologías ágiles; **C2**. Es claro el modelo propuesto y cómo se debe implementar para la gestión de la calidad en proyectos ágiles; **C3**. El modelo propuesto en el artículo es una adaptación de modelos implementados en otro tipo de metodologías para el desarrollo de productos de software; **C4**. En el artículo se puede evidenciar que el modelo propuesto ha sido implementado en un ambiente productivo; **C5**. Los resultados en el artículo demuestran que hace un aporte importante en el problema de investigación; **C6**. La propuesta del artículo aplica solo en proyectos ágiles; **C7**. El artículo propone solución basado en estadísticas generadas desde un estudio de caso real. Se espera que mediante este conjunto de criterios se logre la identificación de las necesidades en la implementación de metodologías ágiles.

La evaluación de los artículos seleccionados fue de tipo cuantitativo, basada en los criterios enunciados anteriormente. La calificación se establece con los siguientes valores y definiciones: 3: Cumple totalmente con el criterio; 2: Cumple parcialmente con el criterio; y 1: No cumple con el criterio. Basados en los criterios de inclusión, exclusión y evaluación mencionados previamente, se eligieron diez artículos, los cuales se describen en la sección 3.1.

## 3. Resultados y discusión

### 3.1 Resultado de la evaluación

Según los criterios ya mencionados, en la tabla 2 se presenta un resumen de los resultados de la evaluación, en la cual se asigna a cada artículo una calificación por cada criterio, en una escala que varía desde el 1 al 3, siendo el 1 la correspondencia más baja y el número 3 la más alta. Los mejores artículos, bajo esta estructura de evaluación son los que poseen la mayor suma de todos los criterios, como se muestra en la última columna de la tabla 2.

Trabajos	Criterios de evaluación	C1	C2	C3	C4	C5	C6	C7	T
	ID01: (Losada, Urretavizcaya, & Fernández-Castro, 2013)	2	2	3	2	3	2	2	16
	ID02: (Kayes, 2011)	3	3	2	3	3	3	3	20
	ID03: (Banerjee, Narasimhan, & Kanakalata, 2011)	2	3	2	3	3	3	3	19
	ID04: (Elssamadisy & Whitmore, 2006)	3	2	1	3	3	3	3	18
	ID05: (Minoli, Castro, & Garzás, 2010)	3	2	3	2	3	3	3	19
	ID06: (Mostefaoui & Simpson, 2008)	3	3	1	3	3	2	3	18
	ID07: (López-Nores et al., 2009)	2	3	2	3	3	3	3	19
	ID08: (Tarhan & Yilmaz, 2014)	3	3	1	3	3	3	2	18
	ID09: (Lee & Xia, 2010)	2	3	2	1	3	3	3	17
ID10: (Nazareno et al., 2013)	3	3	1	3	3	3	3	19	

**Tabla 2.** Evaluación de los artículos seleccionados

En **ID01** (Losada et al., 2013), se plantea la integración de tres metodologías: Métodos Ágiles (AM), Desarrollos Dirigido por Modelos (MDD) y Diseño Centrado en el Usuario (UCD); esto con el objetivo de proporcionar un nuevo marco de trabajo para las metodologías ágiles, en el cual se garantice la calidad de los desarrollos que se realicen bajo esta. Se muestra además como cada una de estas metodologías tienen prácticas que se pueden aprovechar ampliamente para garantizar la calidad en los proyectos de desarrollo de software, sin embargo, también poseen algunas características que los hacen difíciles de aplicar. En cuanto a las AM, se destaca su naturaleza evolutiva a partir de las necesidades del cliente y el contacto con el mismo para recibir una constante retroalimentación. En el MDD, se tiene un amplio conocimiento del proceso y éste se encuentra documentado; sin embargo, según los autores, se encuentra subvalorado ya que el diseño de los modelos por lo general solo sirve de documentación. Adicionalmente, al tratar de integrar el MDD con AM, los posibles cambios que surgen en cada iteración, hacen que los modelos pierdan validez, por tal motivo se deben generar de nuevo, lo cual puede resultar muy costoso para el proyecto. Finalmente, en UCD, como gran ventaja se tiene el contacto con el usuario final, lo cual significa un mejor entendimiento de las necesidades puntuales del negocio y la pronta retroalimentación que requieren las AM. Como aspecto

negativo en el UCD se destaca la necesidad de conocer y evaluar todos los requerimientos antes de poder ser implementados, de manera muy similar a como se hace en las metodologías tradicionales, lo cual nuevamente entra en conflicto con las AM en las que se trata de disminuir al máximo la documentación.

En **ID02** (Kayes, 2011), se propone una métrica llamada PRAT, empleada para medir el proceso de calidad en un proyecto que se desarrolla bajo la metodología Scrum. Este artículo enfatiza cómo el proceso de calidad que se siga en el proyecto, está altamente relacionado con la calidad en la entrega del producto como tal y en la complejidad de evaluar cada iteración que se realiza. El foco principal de la métrica PRAT, es el ProductBacklog, de allí se obtienen una serie de factores agrupados en dos categorías: ProductBacklog Complexity Level (PCL) y Test Assessment Rating (TAR). El primero indica qué tan complejo de evaluar puede resultar el ProductBacklog. Algunos de los factores que se cuantifican para determinar este valor son: complejidad de lógica de negocio, cantidad de datos de prueba, dependencias, tanto externas como internas, con otros módulos, entre otros. La segunda categoría es la calificación obtenida al probar el ProductBacklog de una determinada iteración. En ella se consideran aspectos como: cantidad de bugs reportados, relevancia de cada bug, impac-

to negativo de la solución de un bug sobre otros módulos del proyecto, etc. Una vez obtenidos todos estos factores, se procede a calcular el PRAT para cada liberación del producto. Esta métrica da cuenta de cómo se sigue el proceso de aseguramiento de calidad del proyecto, para determinar si el proceso mejora o se deteriora tras cada liberación, siempre bajo la premisa que un buen proceso de calidad da como resultado un producto de calidad.

En **ID03** (Banerjee, Narasimhan, & Kanakalata, 2011), se presenta el caso de desarrollo en la modalidad de Offshoring, ubicada en la India, certificada CMMI nivel 5 y que la mayoría de los proyectos en los cuales trabaja, se ejecutan bajo metodologías tradicionales (Cascada). Para el caso de estudio que presentan, su cliente es una empresa que está ubicada en Estados Unidos, con un proyecto a desarrollar en un tiempo máximo de 9 semanas y bajo metodología Scrum. Las características mencionadas dan pie a una serie de problemáticas que dificultan el desarrollo del proyecto, entre las cuales se encuentra el aseguramiento de la calidad. Para ello se opta por implementar el Desarrollo Dirigido por Pruebas (TDD), sin obtener los mejores resultados, debido a que los constantes cambios que surgen a lo largo de cada sprint, afectan las pruebas ya existentes obligando al equipo de desarrollo a modificar una y otra vez las pruebas. Al finalizar el artículo se muestra como el acoplamiento entre metodologías tradicionales y ágiles es posible, sin embargo, el aseguramiento de la calidad del proyecto no dio los mejores resultados ya que se convirtió en un sobrecosto para la empresa.

En **ID04** (Elssamadisy & Whitmore, 2006), la problemática planteada son las consecuencias de tener unas pruebas funcionales mal enfocadas. Para ello se propone un patrón de pruebas funcionales para el correcto diseño de la prueba de manera tal que realmente se pueda garantizar la calidad del proyecto. La propuesta se puede aplicar a cualquier proyecto que se desarrolle de manera iterativa y para ejemplificarla se hace referencia a un

caso práctico. Esta propuesta no solo es aplicable al actual problema de estudio sino que puede ser aplicable para otras metodologías ya que el centro del mismo son las pruebas de funcionalidad.

En el artículo **ID05** (Minoli et al., 2010), se plantea como estrategia de aseguramiento de calidad en las metodologías ágiles, la implementación de técnicas tales como Desarrollo Dirigido por Pruebas (TDD) y Desarrollo Dirigido por Pruebas de Aceptación (ATDD), de esta forma se tendrá como garante que todo lo desarrollado cumple con lo que el cliente solicita y por tanto es un producto de calidad ya que satisface las expectativas del usuario final.

El **ID06** (Mostefaoui & Simpson, 2008), presenta un conjunto de buenas prácticas y herramientas que pueden ser usadas en proyectos basados en web services que se desarrollan bajo metodologías ágiles. Para facilitar el proceso de calidad, se hace una clasificación en dos ejes, vertical y horizontal, de los aspectos más relevantes a tener en cuenta cuando se empiezan a diseñar los casos de prueba de determinado proyecto. De manera paralela, a medida que se listan las consideradas buenas prácticas, se mencionan las herramientas que pueden apoyar el proceso de calidad, herramientas tales como JUnit y JMeter son mencionadas. En este artículo no solo se habla de la calidad funcional del proyecto sino de la calidad no funcional.

En el **ID07** (López-Nores et al., 2009), se presenta una propuesta para asegurar la calidad del software basados en integración continua. Para ello se plantea un algoritmo que permite hacer el análisis sistemático del código, detectando falencias en el mismo y generando una propuesta de solución. Dicha propuesta puede ser aceptada o rechazada por los implicados en el proceso de desarrollo. La ejecución del algoritmo puede ser realizada en cualquiera de las fases del proyecto, teniendo en cuenta que entre más veces se implemente, mejor calidad tendrá el producto. El algoritmo presentado en este artículo fue desarrollado específicamente para ser aplicado en proyectos ágiles, sin

embargo, su implementación en ciertos escenarios puede ser compleja, por lo cual requiere cierto grado de experiencia y conocimiento de parte del equipo que lo implemente. El autor propone esta solución como una herramienta que permite detectar y resolver problemas tan pronto se presentan para garantizar la calidad en todo momento.

El **ID08** (Tarhan & Yilmaz, 2014), propone como solución al tema de la calidad un método empírico basado en la utilización de métricas cualitativas y cuantitativas llamado meta pregunta métrica, GQM por sus siglas en inglés. Esto permite realizar un análisis sistemático que ayude a detectar falencias u oportunidades de mejora en cada iteración del desarrollo. Esta práctica continua muestra un incremento en la calidad con cada iteración, debido a que la interacción con los usuarios finales hace que los cambios y mejoras al producto sean detectados en cortos períodos de tiempo. Una de las formas de obtener algunas métricas, fue gracias a la implementación de marcos de trabajo como CMMI. Según los resultados exitosos, se puede pensar en la viabilidad de acoplar marcos de trabajo ideados bajo esquemas de desarrollo tradicional a las metodologías ágiles, tal como se plantea en una de las preguntas de investigación del artículo.

El **ID09** (Lee & Xia, 2010), plantea una serie de hipótesis generadas según las características que tenga el equipo de trabajo en un proyecto ágil en cuanto a agilidad, autonomía y diversidad, las cuales afectan, de manera positiva o negativa, la calidad de los productos entregados; lo anterior debido a que un grupo ágil puede captar más fácilmente las necesidades de los clientes y si son autónomos toman decisiones sin tener que escalar a otros entes que pueden generar retrasos en las iteraciones; pero también es necesario que el grupo sea diverso en conocimiento ya que esto genera mayor potencial a la hora de enfrentar los retos que propone el desarrollo evolutivo, donde los roles pueden rotar, todos pueden ser tester y desarrolladores. El resultado de esta investigación es una serie de recomendaciones a tener en cuenta

en el momento de conformar equipos de desarrollo. A diferencia de los otros artículos, ID09 destaca el tema de la calidad del software como una consecuencia de un buen trabajo de equipo; sin embargo, este concepto solo aplica a la calidad del proyecto en la etapa de desarrollo y no se mencionan elementos que garanticen la calidad a futuro.

En **ID10** (Nazareno et al., 2013), se enfatiza en la dificultad que puede significar el proceso de calidad, en un proyecto desarrollado bajo metodologías ágiles. En este caso se usa como referente un proyecto realizado en Scrum, siendo una de las metodologías de mayor aceptación entre las comunidades ágiles. Según ID10, la complejidad del aseguramiento de la calidad, se debe principalmente a la poca documentación que se realiza en el proyecto y a la naturaleza cambiante del mismo. La propuesta concreta que se realiza, es la creación de un modelo conceptual, el cual responde una serie de preguntas sobre qué sería un producto de calidad esperado por el usuario final. Las preguntas y respuestas obtenidas, deben ser tenidas en cuenta en cada una de las etapas del desarrollo.

### 3.2 Análisis y discusión de resultados

De acuerdo con los documentos analizados, se identifica una tendencia de uno o dos artículos publicados por año desde 2006 hasta 2014, en donde 2006 tuvo un aporte muy bajo asociado a la novedad del tema de análisis. En 2007 no se presentan publicaciones con los criterios definidos; entre 2008 y 2011 se hicieron publicaciones que aportan la mayoría de la información analizada; los años 2009 y 2010 presentan un crecimiento en el interés sobre el tema. De acuerdo con la tendencia detectada, se espera que para el 2014 se presenten más publicaciones que busquen brindar una solución. No existe una clara estructura metodológica para la implementación de las metodologías mencionadas, lo cual imposibilita su replicabilidad en entornos de desarrollo. No obstante, la información relevante encontrada no puede asociarse directamente al conocimiento amplio de la meto-

dología para desarrollos ágiles, ya que la calificación dada por los autores denota dicha relación. En los artículos estudiados también se puede detectar que el país con más investigaciones en este tópico es España con un 30%, seguido de Estados Unidos con un 20%; teniendo en cuenta que el manifiesto ágil nació en Estados Unidos se infiere que las prácticas ágiles no son un tópico al que se dedique mucha investigación, caso contrario pasa con España, donde sí se percibe que la calidad del software adquiere mayor importancia.

Para las preguntas de investigación se realiza el análisis siguiente:

i. Cuáles son las herramientas y buenas prácticas empleadas para garantizar la calidad en proyectos de software gestionados mediante metodologías ágiles?: De acuerdo con la investigación realizada, se puede decir que existen diferentes herramientas que sirven para ayudar a garantizar la calidad en el software desarrollado con metodologías ágiles. Sin embargo, estas herramientas no nacieron orientadas para esta metodología, lo cual hace que no sean aplicables en la mayoría de los proyectos; son herramientas que se han adaptado de las metodologías tradicionales o creado para un proyecto específico. Respecto a las buenas prácticas, estas son muy variables; tampoco existe algo como ITIL, o un estándar creado que ayude a su generalización; en cada proyecto analizado se pueden identificar diferentes metodologías. Por ejemplo, respecto a las propuestas de solución al problema, de 10 artículos solo 2 apuntan al control de la calidad con integración continua; en ambos se presenta esta técnica como solución a la naturaleza iterativa de los procesos, ya que de esta forma se garantiza que no se generen pasos a producción con errores, aclarando que todo ello dependerá de una buena estructuración del proceso de creación del producto.

ii. Existen frameworks para la gestión de la calidad en proyectos de software que se puedan aplicar a metodologías ágiles?: Dos artículos hablan sobre CMMI, demostrando que este marco de trabajo

también puede ser usado en metodologías ágiles, aunque no se use en un 100% ya que solo se implementan ciertas prácticas para el control de procesos en el desarrollo del producto. Pese a que en 4 de los artículos se hace énfasis en las pruebas como garante de la calidad del proyecto, en uno de ellos la experiencia no fue positiva debido a la poca experiencia del equipo en el desarrollo dirigido por pruebas, TDD; en éste caso, la consecuencia fue un sobrecosto del proyecto. Esto demuestra que diseñar planes de prueba no es suficiente, sino que se requiere una preparación previa para el equipo y que la calidad se debe evaluar de manera constante, en las diferentes etapas del proyecto, desde el momento mismo de la creación de las historias de usuario, pasando por el diseño de arquitectura y llegando hasta la escritura del código como tal.

iii. Las metodologías para el control de calidad utilizadas por los autores aplican para todo tipo de proyectos ágiles?: La mayoría de los artículos seleccionados plantean una serie de elementos que tienen alto impacto en el mejoramiento de la calidad, entre ellos se puede mencionar: la adecuada creación de historias de usuario realizada de manera conjunta entre el cliente y el equipo de desarrollo, la experiencia del equipo, el control de la calidad en cada una de las fases del proyecto involucrando la mayor cantidad de roles posibles, el seguimiento de diferentes métricas de calidad en cada iteración estableciendo acciones para el mejoramiento continuo de dichas métricas, la combinación de técnicas que se complementen unas con otras, el grado de acoplamiento y sinergia de los integrantes del equipo al igual que propiciar un ambiente cómodo de trabajo, entre otras.

### 3.3 Recomendaciones

La propuesta que se plantea no es un herramienta como tal, es más un conjunto de recomendaciones a seguir durante el proceso de desarrollo del proyecto y que se aplicarán según las necesidades del mismo; sin embargo, algunas de estas recomendaciones sí pueden estar apoyadas en el uso



de ciertas herramientas. La elección de cuando usar cada recomendación se deja al libre criterio y experiencia del lector.

Desarrollo Dirigido por Pruebas, TDD. La implementación de TDD no solo indica que el componente que se desarrolla funcione en la manera prevista, también es un indicador del grado de acoplamiento del proyecto, ya que si las pruebas diseñadas fallan constantemente con la evolución del desarrollo, es posible que la solución como tal tenga falencias en su diseño. Se sugiere aplicar durante el diseño de la arquitectura y se debe buscar siempre que ésta promueva alta cohesión y bajo acoplamiento entre sus elementos, ya que estas son dos características necesarias en un buen diseño de software. Experiencia con TDD. Pueden emplearse herramientas como JUnit y NUnit.

Desarrollo Dirigido por Pruebas de Aceptación, ATDD. Se incluyen pruebas de aceptación orientadas a las funcionalidades que debe tener el proyecto, y son escritas inicialmente por el product owner o por un experto del lado del negocio e implementadas por el desarrollador. Se sugiere aplicar siempre, ya que este tipo de pruebas es fundamental porque es la mejor forma de entender con claridad la necesidad del cliente. Pueden emplearse herramientas como Fit/FitNess y Cucumber.

Integración Continua, IC. Esta es una práctica que para ser aplicada correctamente, debe cumplir con ciertos requisitos como son: mantener un único repositorio del proyecto, automatizar la compilación y despliegue para que se ejecuten cada vez que el equipo haga un *commit*, automatizar las pruebas para que se ejecuten de manera constante en períodos regulares de tiempo, si las pruebas fallan con algún *commit* realizado, éste deberá ser devuelto y corregido inmediatamente, realizar mínimo un *commit* al día, el ambiente de pruebas debe ser lo más cercano posible al ambiente de producción, mantener visible para el equipo los despliegues y las pruebas automáticas. Se sugiere aplicar en proyectos en los que cada iteración tiene salida obligada a producción, proyectos di-

vididos en módulos, donde cada módulo es desarrollado por diferentes equipos y proyectos en los cuáles los miembros del equipo se encuentran ubicados en diferentes zonas geográficas. Pueden emplearse herramientas como Git: Repositorio, Jenkins: Servidor de IC y Gradle: para automatizar los build de aplicaciones Java.

Mejoramiento Continuo. El uso de TDD, ATDD e IC garantizan la calidad funcional del desarrollo desde las etapas más tempranas del mismo, pero no se pueden dejar de lado los atributos no funcionales y mejoras que se puedan realizar sobre el código ya implementado. La realización de este tipo de pruebas no requiere que sea tan frecuente como las de TDD o ATDD, ya que se pueden realizar cada pocas iteraciones, de acuerdo a la cantidad o complejidad de los cambios que se realicen en las mismas, y sin dejar de lado diversas consideraciones asociadas a las características particulares de cada tipo de código. Puede aplicarse en proyectos que están definidos a grandes períodos de tiempo (superiores a 1 año), donde todas las funcionalidades van evolucionando con cada iteración y, con ella, la complejidad del proyecto aumenta. Las herramientas de control sugeridas en estos casos son el JMeter: Pruebas de rendimiento y SonarQube: Revisión y análisis estático de código fuente.

#### 4. Conclusiones

Pese a que existen frameworks o herramientas para asegurar la calidad del software en un proyecto ágil, aún existen desafíos por solucionar. Para ello se debe continuar con la investigación y la documentación de las experiencias propias. Además, un proyecto desarrollado con metodologías ágiles, por naturaleza, es flexible y se adapta al constante cambio de acuerdo a las necesidades del cliente. Esta característica hace complejo estandarizar una herramienta para asegurar la calidad del mismo. En vez de eso, se debe contar con un conjunto de buenas prácticas, recomendaciones y herramientas, que, según sea el caso, se apliquen en determinado momento del proyecto. Los frameworks que se usan en el aseguramiento de la

calidad en los proyectos de desarrollo tradicional, han sido probados y estudiados desde hace mucho tiempo. No se puede descartar el uso de estos, en cambio se debe buscar la manera de adaptarlos a los proyectos ágiles.

Las soluciones planteadas en los artículos estudiados, en general, están pensadas en proyectos que funcionan de manera iterativa y evolutiva, por lo cual se puede pensar en que éstas sean aplicadas a cualquier proyecto ágil; sin embargo, también se observó que los proyectos ágiles sufren cambios constantemente; por ello, la solución si bien se puede implementar, debe ser adaptada a cada caso de manera específica.

Así como las metodologías ágiles se basan en un concepto de desarrollo evolutivo, las diferentes pruebas que se realicen para garantizar la calidad del producto deben poseer esta misma característica, independientemente de si son pruebas funcionales, no funcionales o de aceptación.

En un entorno de desarrollo ágil, el aseguramiento de la calidad del proyecto no es simplemente una etapa durante el ciclo de vida del mismo y que pueda ser delegados a expertos en el tema que no hagan parte del equipo; por el contrario, el aseguramiento de la calidad es un tema que concierne a todos, desde el cliente que debe velar por tener bien definidas las historias de usuario y así tener escenarios de prueba efectivos, hasta el arquitecto que debe presentar una solución lo suficientemente desacoplada que permita la evolución en el tiempo del proyecto y pasando por los desarrolladores, quienes son los directamente encargados de crear las pruebas como tal.

Finalmente se hizo una propuesta de solución, no con un aplicativo sino con unas recomendaciones o pasos a seguir como mejores prácticas, con el fin de lograr un aseguramiento de la calidad, pero sin depender de marcos de trabajo ni herramientas. En el proceso de calidad lo realmente importante es que ésta se asegure en todas las fases del proyecto, por lo que es necesario utilizar diferentes

herramientas o metodologías que apoyen cada etapa del proceso de creación del producto, desde la toma de los requisitos y la creación de historias de usuario hasta las pruebas de aceptación. La solución aquí propuesta se puede considerar tan solo como una primera etapa. Es necesario seguir documentando las experiencias obtenidas en el aseguramiento de calidad para los proyectos ágiles y así tener más referentes para mejorar las recomendaciones planteadas en trabajos futuros.

### Referencias

Banerjee, U., Narasimhan, E., & Kanakalata, N. (2011). Experience of executing fixed price off-shored agile project. In Proceedings of the 4th India Software Engineering Conference on - ISEC '11 (pp. 69–75). New Delhi. <http://doi.org/10.1145/1953355.1953364>

Beck, Kent. Beedle, Mike. Bennekum, m Arie van. Cockburn, Alistair. Cunningham, Ward. Fowler, Martin. Grenning, James. Highsmith, Jim. Hunt, Andrew. Jeffries, Ron. Kern, Jon. Marick, Brian.

Martin, R., Mellor, S., Schwaber, K., & Jeff, T.D. (2001). Manifiesto por el Desarrollo Ágil de Software. Retrieved from <http://agilemanifesto.org/iso/es/>

Biolchini, J., Mian, P.G., Natali, A., & Travassos, G. (2005). *Systematic Review in Software Engineering*. Rio de Janeiro. Retrieved from <http://www.cin.ufpe.br/~in1037/leitura/systematicReviewSE-CO-PPE.pdf>

Buonamico, D. (2013). Historia de las Metodologías Ágiles en Contexto. Retrieved July 20, 2014, from <http://www.caminoagil.com/2013/03/historia-de-las-metodologias-agiles-en.html>

Elsamadisy, A., & Whitmore, J. (2006). Functional testing: a pattern to follow and the smells to avoid. In *PLoP® 2006 Conference Proceedings* (pp. 27:1–27:13). New York. <http://doi.org/10.1145/1415472.1415504>

- Fitzgerald, B., Stol, K. J., O'Sullivan, R., & O'Brien, D. (2013). Scaling agile methods to regulated environments: An industry case study. In *Proceedings - International Conference on Software Engineering* (pp. 863–872). IEEE. <http://doi.org/10.1109/ICSE.2013.6606635>
- Kayes, I. (2011, March). Agile Testing: Introducing PRAT as a Metric of Testing Quality in Scrum. *ACM SIGSOFT Software Engineering Notes*. <http://doi.org/10.1145/1943371.1943384>
- Ktata, O., & Lévesque, G. (2010). Designing and Implementing a Measurement Program for Scrum Teams: What do agile developers really need and want? In *C3S2E '10 Proceedings of the Third Conference on Computer Science and Software Engineering* (pp. 101–107). <http://doi.org/10.1145/1822327.1822341>
- Lee, G., & Xia, W. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility. *MIS Quarterly*, 34(1), 87–114. <http://doi.org/Article>
- López-Nores, M., Pazos-Arias, J.J., García-Duque, J., Blanco-Fernández, Y., Díaz-Redondo, R.P., Fernández-Vilas, A., ... Ramos-Cabrera, M. (2009). Procedures and algorithms for continuous integration in an agile specification environment. *International Journal of Software Engineering and Knowledge Engineering*, 19(01), 47–78. <http://doi.org/10.1142/S0218194009004106>
- Losada, B., Urretavizcaya, M., & Fernández-Castro, I. (2013). A guide to agile development of interactive software with a “user Objectives”-driven methodology. *Science of Computer Programming*, 78(11), 2268–2281. <http://doi.org/10.1016/j.sico.2012.07.022>
- Minoli, M., Castro, V., & Garzàs, J. (2010). Reduciendo distancia en proyectos de Desarrollo de Software Global Ágiles con técnicas de Ingeniería de Requisitos. *Revista Española de Innovación, Calidad E Ingeniería de Software*, 6(3), 66.75. Retrieved from <http://www.redalyc.org/src/inicio/ArtPdfRed.jsp?iCve=92218768006>
- Mostefaoui, G.K., & Simpson, A. (2008). On quality assurance of web services in agile projects: an experience report. In *Proceedings of the 3rd ACM workshop on Software engineering for pervasive services* (pp. 21–26). <http://doi.org/http://doi.acm.org/10.1145/1387229.1387235>
- Nazareno, R., Leone, H., Gonnet, S., Conicet, I., Avellaneda, U. T. N., & Fe, S. (2013). Trazabilidad de Procesos Ágiles: un Modelo para la Trazabilidad de Procesos Scrum. In *XVIII Congreso Argentino de Ciencias de la Computación* (pp. 920–929). Buenos Aires - Argentina.
- Schuh, P. (2007). Agile Configuration Management for Large Organizations. Retrieved November 14, 2014, from <http://www.ibm.com/developerworks/rational/library/mar07/schuh/>
- Tarhan, A., & Yilmaz, S.G. (2014). Systematic analyses and comparison of development performance and product quality of Incremental process and agile process. *Information and Software Technology*, 56(5), 477–494. <http://doi.org/10.1016/j.infsof.2013.12.002>