



EVALUACIÓN DE CALIDAD EN ENTORNOS ÁGILES: UN ESTUDIO COMPARATIVO DE LA LITERATURA

César Acuña, UTN FRRe, csr.acn@gmail.com

Noelia Pinto, GICS UTN FRRe, ns.pinto@gmail.com

Gabriela Tomaselli, GICS UTN FRRe, gabriela.tomaselli@gmail.com

Nicolás Tortosa, GICS UTN FRRe, nicotortosa@gmail.com

Resumen— Trabajar en el mejoramiento de los procesos software es un paso que las empresas en la Industria del Software deben dar 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 la adopción de un modelo de calidad adecuado a las características de la empresa, y de una metodología que guíe el ciclo de desarrollo de software.

El presente trabajo aborda de manera general el estado del arte respecto a Modelos de Calidad de Software y su aplicabilidad en procesos ágiles, y aporta elementos comparativos teniendo en cuenta diferentes dimensiones. Además, a partir de esta comparación, se presenta de forma preliminar el diseño de un nuevo modelo que contribuirá a evaluar la calidad en escenarios caracterizados por procesos ágiles.

El objetivo es determinar las características más relevantes y frecuentes entre los modelos seleccionados, estableciendo cuáles pueden considerarse, a priori, componentes de un nuevo modelo que surja como propuesta superadora e integradora y permita evaluar la calidad de procesos ágiles junto a los productos de software obtenidos.

Palabras clave— *Calidad del Producto y del Proceso de Software, Metodologías ágiles, Modelos de Calidad.*

1. Introducción

El rol de América Latina en la industria mundial de Software y Servicios Informáticos (SSI) todavía no es compatible con su importancia económica, pero gradualmente se observa un aumento de su participación, aprovechando un creciente mercado interno y las oportunidades en desarrollo [1].

Y si bien las micro, pequeñas y medianas empresas -PYMES- son una pieza muy importante en el engranaje de la economía mundial [2], en Latinoamérica este tipo de organizaciones insertas en la Industria del Software aún evidencian falta de efectividad en sus procesos generando riesgo respecto a su nivel competitivo en escenarios internacionales.

Particularmente en el caso de Argentina, y según el último informe anual [3] del Observatorio Permanente de la Industria del Software y Servicios Informáticos de la República (OPSSI), el sector se conforma con más del 85% de empresas PYMES con menos de 200 trabajadores, donde se ubica el mayor índice de empleabilidad de la industria. Las mismas constituyen, hoy, un eslabón fundamental en el sector productivo de Argentina, por lo que resulta imperioso implementar ciclos de mejora de procesos que contribuyan con el desarrollo de esta industria.

Es por ello que la calidad del producto final, los bajos costos y las entregas oportunas se transforman en elementos claves para el incremento de las ventas internas y la proyección a nivel internacional del sector. Así, una estrategia fundamental para la consecución de estos objetivos es la implementación de cuestiones asociadas a la gestión de calidad, no solo desde el punto de vista del proceso de desarrollo sino enfocado también al producto de software final que se obtenga.

Varios autores [4][5][6] coinciden en que resulta dificultoso 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.

La calidad del software se define como el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia [7]. Sin embargo no es correcto asociar calidad solo al producto final sino que también resulta necesario ajustar los parámetros asociados a los procesos que han facilitado la obtención del mismo.

En este sentido existen numerosas propuestas metodológicas para el desarrollo de software que inciden en distintas dimensiones del proceso. Las propuestas 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 [8].

Sin embargo, estos enfoques no resultan ser los más adecuados para muchos de los proyectos actuales, donde el entorno del sistema es muy cambiante y en donde se exige reducir drásticamente los tiempos de desarrollo, pero manteniendo una alta calidad. Surgen, entonces, metodologías ágiles las 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 [9].

Dado que la calidad del producto de software desarrollado está estrechamente relacionada con la calidad del proceso utilizado, las PyMEs necesitan implementar proyectos para la mejora de sus procesos que contribuya, de esta forma, incrementar la calidad de sus productos.

El objetivo de este trabajo es presentar un marco comparativo que permita analizar la aplicabilidad de evaluación de calidad sobre procesos ágiles en base a modelos de calidad ya existentes, no solo desde el punto de vista del proceso de software sino también del producto obtenido. A partir de este análisis se podrán seleccionar los componentes que darán forma a un nuevo modelo que se ajuste a la evaluación de calidad de procesos ágiles y los productos que se obtengan.

El artículo se estructura como sigue: en la sección 2 se incluye el análisis de comparación entre diversos modelos de calidad que pueden aplicarse sobre metodologías ágiles. Luego, en la sección 3, se presentan las características que se consideran necesarias para diseñar la propuesta de modelo de calidad que se pretende obtener como resultado. Finalmente se exponen conclusiones y trabajos futuros que se pretenden abordar a partir de esta línea.

2. Materiales y Métodos

La necesidad de desarrollo rápido de aplicaciones de alta calidad ha llevado a darle gran importancia al concepto de calidad en todas las etapas. Un modelo proporciona un marco y un lenguaje para comunicarse, también proporciona un estándar y la experiencia necesaria en el tema a abordar. Siguiendo con este concepto, 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 [10]. Estos últimos pueden ser medidos directa o indirectamente a través de la definición de criterios de calidad.

En este sentido, por un lado los factores de calidad o atributos externos, representan la calidad desde el punto de vista del usuario. Y por otro 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. Las métricas se definen para cada criterio de calidad, representando medidas cuantitativas que indican el grado en el que está presente un atributo en el producto [11]. 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 sub-características (que en definitiva representa el modelo de calidad).

Entre los modelos de calidad que permiten la evaluación del producto de software, se destaca para este estudio en primer lugar el Modelo de Mc Call, creado por Jim Mc Call en 1977 [12]. El mismo establece 3 perspectivas para el análisis de la calidad de software, define 11 factores y 23 criterios relacionados a estos tal como se muestran en la Tabla 1. Estos factores se agrupan en 3 perspectivas (Operatividad del Producto, Revisión del Producto y Transición del Producto). Las métricas que propone son preguntas que aplican una ponderación numérica a un determinado atributo del producto de software. Después de obtener los valores para todas las métricas de un criterio específico, el promedio de todas ellas es el valor para ese criterio.

Tabla 1. Factores de Calidad y Criterios asociados en el modelo de Mc-Call

FACTORES	CRITERIO
Usabilidad	Operatividad
	Entrenamiento
	Comunicación
Integridad	Control de Acceso
	Auditoría de Acceso
Corrección	Rastreabilidad
	Compleitud
	Consistencia
Fiabilidad	Consistencia
	Exactitud
	Tolerancia a Fallos
Eficiencia	Eficiencia en Ejecución
	Eficiencia en Almacenamiento
Mantenibilidad	Simplicidad
	Concreción
Facilidad de Prueba	Simplicidad
	Instrumentación
	Modularidad
Flexibilidad	Capacidad de Expansión
	Generalidad
	Modularidad
Reusabilidad	Generalidad
	Modularidad
Interoperabilidad	Similitud de Comunicación
	Similitud de Datos
	Independencia del sistema
	Independencia del hardware
Portabilidad	Independencia del sistema
	Independencia del hardware

Fuente: De elaboración propia

Otro modelo seleccionado para su análisis fue FURPS [13], desarrollado por Hewlett-Packard en el año 1987, en el que se describen un conjunto de factores de calidad de software: funcionalidad (Functionality), usabilidad (Usability), confiabilidad (Reliability), desempeño (Performance) y capacidad de soporte (Supportability), cada uno de estos factores tienen atributos de calidad, como se incluye en la Tabla 2. Estos elementos pueden usarse para establecer métricas de calidad para todas las actividades del proceso de software.

Tabla 2. Factores de calidad y criterios asociados al modelo FURPS

FACTORES	CRITERIO
Funcionalidad	Características y capacidades del software
	Generalidad de las funcionalidades
	Seguridad del Sistema
Usabilidad	Factores Humanos
	Factores Estéticos
	Consistencia de la Interfaz
	Documentación
Confiabilidad	Frecuencia y severidad de fallas
	Exactitud de las salidas
	Tiempo medio de fallas
	Capacidad de recuperación de fallos
	Capacidad de predicción
Rendimiento	Velocidad de procesamiento
	Tiempo de Respuesta
	Consumo de recursos
	Eficacia
Capacidad de Soporte	Extensibilidad
	Adaptabilidad
	Capacidad de pruebas
	Compatibilidad

Fuente: De elaboración propia

Entre los estándares internacionales se destaca ISO/IEC 9126 [14], aplicable a todo tipo de software, está basado en un modelo jerárquico con tres niveles: Características, Subcaracterísticas y Métricas. El primer nivel tiene seis características principales: Funcionalidad, Fiabilidad, Eficiencia, Facilidad de Mantenimiento, Portabilidad y Facilidad de Uso. Estas características (factores) están compuestas a su vez por subcaracterísticas (subfactores) relacionadas con la calidad externa, y subcaracterísticas relacionadas con la calidad interna. La relación entre los factores y sus criterios se incluyen en la Tabla 3.

Tabla 3. Factores de calidad y criterios asociados al estándar ISO/IEC 9126

FACTORES	CRITERIO
Funcionalidad	Adaptabilidad
	Exactitud
	Interoperabilidad
	Seguridad
Usabilidad	Comprensibilidad
	Aprendizaje
	Operatividad
	Atractivo

Fuente: De elaboración propia

Tabla 4 (cont.). Factores de calidad y criterios asociados al estándar ISO/IEC 9126

FACTORES	CRITERIO
Mantenibilidad	Análisis
	Cambio
	Estabilidad
	Prueba
Fiabilidad	Madurez
	Tolerancia a fallos
	Recuperabilidad
Eficiencia	Comportamiento de tiempo
	Uso de los recursos
Portabilidad	Adaptabilidad
	Instalación
	Coexistencia
	Reemplazo

Fuente: De elaboración propia

Además existen modelos de calidad que han sido desarrollados teniendo en cuenta las metodologías ágiles y los procesos que se basan en ellas. Uno de ellos es el conocido como AGIS (combinación entre ÁGIL e ISO) [15], el cual establece un mecanismo para medir el grado de agilidad de procesos de desarrollo de software. Complementa el modelo ISO con 10 dimensiones; esta configuración se orienta a medir el grado de aplicación de los valores del manifiesto ágil [16] en las áreas de conocimiento de la ingeniería. AGIS tiene por objetivo satisfacer dos necesidades: por un lado se enfoca en las empresas, pues este modelo permitirá alcanzar una diferenciación de otras empresas que únicamente hayan certificado calidad a través de ISO 9001:2008. Además, AGIS brinda informe de sugerencias de mejora basadas en la valoración de las dimensiones que propone evaluar. Por otro lado, el modelo ofrecerá a los investigadores una definición objetiva del grado de agilidad de un proyecto, que podrá ser utilizado para comparar proyectos teniendo en cuenta sus resultados. Posee 10 dimensiones entre las cuales se pueden mencionar Definición de Ciclo de Vida (AGIS 1), Capacitación y competencias (AGIS 2), Producción de Software Ejecutable (AGIS 3), etc.

En este estudio se analizó también la aplicabilidad de la Norma ISO 9001:2008 [17] elaborada por la Organización Internacional para la Normalización (ISO), que permite determinar los requisitos para un Sistema de Gestión de la Calidad (SGC), utilizado para aplicación interna en las organizaciones, sin importar si el producto o servicio lo brinda una organización pública o empresa privada, cualquiera sea su tamaño, para su certificación o con fines contractuales.

Otro modelo que se ha estudiado es AGIT (AGile software developmenT) [18] el cual sugiere que la mejor performance es lograda cuando las metas de todos los stakeholders son satisfechas. Esto requiere una aproximación balanceada considerando los puntos de vistas de las diferentes stakeholders, para lo cual se definen indicadores adecuados a cada uno. AGIT considera cuatro diferentes puntos de vistas para stakeholders. En primer lugar se considera al stakeholder Administrador IT que es el actor principalmente preocupado con los aspectos tradicionales de la performance del desarrollo de SW considerando tiempo, costo y calidad. El segundo stakeholder se representa con los miembros del equipo cuya meta es la “satisfacción del trabajo”. El ScrumMaster es el tercer stakeholder cuya principal meta es la “resolución eficiente

de impedimentos”. Finalmente, el principal objetivo que buscan los clientes, el cuarto stakeholder, es su propia satisfacción. Este modelo sugiere evaluar la calidad de los procesos de desarrollo considerando los puntos de vistas de los diferentes stakeholders intervinientes, describiendo los indicadores que se adecúan a cada uno de estos perfiles.

Por otro lado se estudió también al Modelo COBIT (Control Objectives for Information and Related Technology) [19], una herramienta que representa determinada colección de documentos que pueden ser clasificados como buenas prácticas generalmente aceptadas para la administración, control y garantía IT. En COBIT, estos dominios se denominan: Plan y Organización (PO): Provee direccionamiento para solucionar la entrega (AI) y la entrega del servicio (DS); Adquisición e implementación (AI); Provee las soluciones y las convierten en servicios; Entrega y Soporte (DS): Recibe las soluciones y las hace utilizables para usuarios finales; Monitoreo y Evaluación (ME): Monitorea todos los procesos para asegurar que la dirección provista es seguida. A través de estos cuatro dominios, COBIT ha identificado 34 procesos IT. Para cada uno de estos 34 procesos COBIT define metas y métricas para definir y medir sus resultados y performances, basado en los principios de la tarjeta de puntuación de negocios balanceada (BSC).

Finalmente entre los modelos que se aplican a procesos de desarrollo de software se analizó a CMMI (Capability Maturity Model Integration) [20] o Integración de modelos de madurez de capacidades, un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. CMMI tiene cuatro disciplinas para elegir: Sistemas de Ingeniería (SE), Ingeniería de Software (SW), Procesos de Desarrollo y Productos (IPPD) y Distribución (SS). El modelo en sí tiene dos representaciones. La representación por etapas se centra en un conjunto de áreas de proceso, que se organizan por niveles de madurez (1-5), mientras que en la representación continua cada área de proceso se clasifican en términos de nivel de capacidad (0-5).

CMMI y métodos ágiles también han sido comparados en varios estudios [21][22][23], por ejemplo, Paulk [24] sugiere que el uso de historias XP, en las instalaciones del cliente y la integración continua permite cumplir los objetivos de gestión de requisitos CMMI-SW. Por otra parte, Turner y Jain [25] determinaron en su estudio que varios de los componentes de CMMI y métodos ágiles estaban en conflicto, la mayoría de ellos relativos a los procesos de organización.

Con todo este análisis se procedió a realizar la comparación de acuerdo a los componentes de cada modelo, poniendo énfasis en la aplicabilidad de cada enfoque sobre procesos ágiles.

3. Resultados y Discusión

En este sentido, primeramente se obtuvo el cuadro comparativo representado en la Tabla 4, donde se indican los factores de calidad y atributos internos que se priorizan y se utilizan en los diferente enfoques analizado desde el punto de vista de la calidad del Producto.

Tabla 4. Factores de Calidad por Enfoque

FACTORES	ENFOQUES		
	MC-CALL	FURPS	ISO/IEC 9126
Usabilidad	X	X	X
Integridad	X		
Corrección	X		
Fiabilidad	X		X
Eficiencia	X		X
Mantenibilidad	X		X
Facilidad De Prueba	X		
Flexibilidad	X		
Reusabilidad	X		
Interoperabilidad	X		
Portabilidad	X		X
Funcionalidad		X	X
Confiabilidad		X	
Rendimiento		X	
Capacidad De Soporte		X	

Fuente: De elaboración propia

En relación al análisis comparativo resultante, se observa que en cada enfoque se cubren diferentes factores de calidad pero los tres analizados consideran a la Usabilidad como factor de calidad relevante para evaluar productos de software.

Además cada enfoque define diferentes métricas aplicadas para medir los factores de calidad que lo caracterizan. Estas métricas utilizadas, en general fueron creadas para medir los atributos en consideración con las variables internas de cada enfoque, pero algunas de ellas son compartidas o adaptadas de acuerdo al modelo o estándar. Todos los factores y criterios estudiados podrían tenerse en cuenta para evaluar la calidad de productos de software obtenidos a través de procesos ágiles, adecuando su medición de acuerdo a los principios ágiles.

Respecto a los enfoques orientados a procesos en el caso de AGIS, la herramienta ofrece una medición objetiva de la agilidad con la que un proyecto implementa sus procesos. Esta medida, al estar basada en los valores y prácticas del manifiesto, permite diferenciar o segmentar los proyectos de desarrollo de software por su grado de agilidad. Además, al estar construido en

base a la norma ISO 9001:2008, asegura que el equipo ha establecido las bases para una mejora de procesos. Sin embargo, presenta una fuerte desventaja al momento de realizar la evaluación de calidad pues se necesita conocer de Normas ISO 9001, ya que su implementación y metodología tiene sus bases asentadas allí.

En el caso de las normas ISO 9001, el principal punto de controversia para su aplicabilidad en proyectos guiados por metodologías ágiles radica en que los procesos de mejoras tienen planes de largo plazo, fechas que cumplir, se enfocan fuertemente en la definición de procesos y documentación que se genera, diferenciándose así de la filosofía ágil que enfoca sus esfuerzos en obtener un producto terminado y no en la redacción de documentos. Sumado a esto, y teniendo en cuenta el caso de las PyMEs, la posibilidad de utilizar este tipo de herramientas es mucho más dificultosa [18], por el tamaño reducido de los grupos de trabajo, por los problemas de definición de roles dentro de estos grupos, por las responsabilidades poco precisas y por los recursos restringidos con que cuentan este tipo de organización.

Por otro lado para AGIT, la principal desventaja en su aplicabilidad sobre proyectos ágiles se relaciona a la falta de indicadores en áreas claves del proceso, como la gestión de calidad mediante testing o la gestión de cambios.

Analizando las características de COBIT, se observa que sus métricas han sido desarrolladas con las siguientes características en mente: “un alto índice de visión a esfuerzo”; “comparable interna y externamente”; “es mejor tener unas pocas buenas métricas” y “fácil de medir”. Estas características son principalmente compatible con los principios ágiles. Sin embargo, la principal desventaja del modelo es que la totalidad de los indicadores que utiliza son valores numéricos por lo que es necesario almacenar información cuantitativa de todos los procesos, proyectos y actividades que se lleven a cabo para luego poder calcular los índices del modelo. En proyectos guiados por metodologías ágiles, difícilmente se almacene toda la información requerida por el modelo para calcular dichos indicadores, y más aún en las PYMES, donde es probable que ni siquiera se lleven a cabo los procesos que se requieren medir, lo que atenta sobre la completitud del modelo.

Respecto a CMMI, pueden surgir problemas si se desea implementar el modelo en entornos ágiles y resulten necesarias evidencias escritas de las prácticas usadas en el proyecto. Esto no se alinea con la filosofía ágil, principalmente con uno de sus valores que sostiene mantener los niveles de documentación lo más bajo posible durante el proceso.

Además, como los procesos ágiles enfatizan en las prácticas de trabajo adaptables, el desarrollo iterativo y documentación tardía, una instantánea del proyecto puede no corresponderse con las prácticas generales de trabajo de la organización requeridas por CMMI.

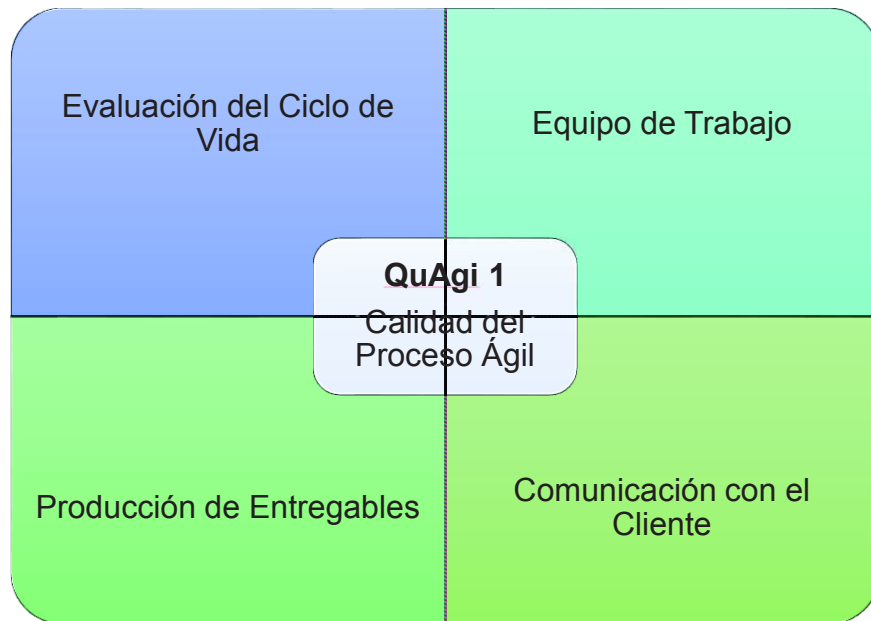
Este análisis comparativo permitió además iniciar el proceso de definición de un framework que permita la evaluación de calidad de proyectos basados en procesos ágiles.

Así, a continuación se presenta una primera propuesta del esquema de componentes que permitirán configurar un nuevo modelo de evaluación de calidad que permitirá ofrecer una medición objetiva de la agilidad con la que un proyecto implementa sus procesos y obtiene sus productos de software, determinando el perfil de agilidad de cada proyecto.

El modelo en el que se ha comenzado a trabajar se denomina QuAgi, y se enfoca en dos perspectivas posibles de evaluar: el proceso ágil y el producto obtenido del mismo. Hasta el

momento se han definido los componentes correspondientes a la primera dimensión y que se muestran en la Figura 1.

Figura 1. Componentes de la dimensión QuAgi 1



Fuente: De elaboración propia

Por tanto, QuAgi 1 será la dimensión que permita evaluar la Calidad a nivel de Proceso en entornos ágiles incluyendo el estudio de los siguientes componentes:

- Evaluación del Ciclo de vida: El Ciclo de Vida de un proyecto de software define el orden de las actividades del proceso productivo. QuAgi ponderará mejor a los ciclos de vida evolutivos y a los incrementales por sobre los demás. Se hará foco en la ejecución del mismo, y no en la documentación que se genere.
- Equipo de Trabajo: El componente humano del proyecto de software a evaluar deberá contar con skills adecuados a la filosofía ágil, y la empresa deberá disponer los medios para lograrlo. Para QuAgi será importante evaluar el flujo de comunicación entre los miembros del equipo y la capacidad del mismo para afrontar prácticas ágiles. Por tanto, en este componente se deberá medir capacidad técnica de los recursos humanos, eficiencia de las reuniones de equipo y tamaño del equipo.
- Producción de Entregables: QuAgi evaluará la periodicidad con la que el proyecto produzca versiones entregables del producto al cliente. En este componente se tendrá en cuenta el cumplimiento del lead time y la validez de cada entregable, favoreciendo a aquellos proyectos cuya validación haya sido automatizada. A este nivel se incluirá también la medición del seguimiento del proceso de gestión de cambios sobre el producto y la implementación de procesos de verificación y validación de los mismos.
- Comunicación con el cliente: El modelo de calidad propuesto propiciará la incorporación del cliente transversalmente a todas las etapas del proyecto, considerándolo como actor principal del mismo. Así, este componente evaluará la ejecución regular de mecanismos de comunicación entre el cliente y el equipo.

4. Conclusiones y recomendaciones

En este trabajo se ha presentado un análisis comparativo entre modelos y normas de calidad existentes que pueden aplicarse a procesos y productos de software. El estudio se realizó teniendo en cuenta aspectos cualitativos, y se enfocó en la aplicabilidad sobre entornos ágiles.

El análisis puso de manifiesto la ausencia de enfoques que permitan abordar la calidad de forma integral teniendo en cuenta ambientes ágiles. Además en la investigación se determinó que existen contextos de desarrollo en el que muchos proyectos se denominan ágiles pero no trabajan respetando los valores del manifiesto ágil. Sumado a esto, al no haber una definición consensuada de agilidad, no es posible comparar resultados de proyectos.

Teniendo en cuenta las cuestiones similares entre los modelos estudiados y tomando como punto de partida el Manifiesto Ágil, se inició el proceso de definición de componentes de una propuesta de modelo que permita evaluar la calidad en proyectos ágiles. Se presentó, como primera aproximación, las dos dimensiones que incluirá el modelo QuAgi, describiéndose particularmente la asociada a Calidad a Nivel del Proceso Ágil.

Como trabajos futuros se pretende continuar en el desarrollo completo de QuAgi, estableciendo la configuración que permita gestionar todos los elementos del mismo y definiendo las estrategias de validación para el mismo. Este modelo constituirá uno de los ejes de un framework para la evaluación de calidad en entornos ágiles, no solo a nivel de proceso sino también de los productos de software que se obtengan.

5. Agradecimientos

Este trabajo se ha realizado en el marco del proyecto de investigación acreditado “Framework para la evaluación de Calidad del Software”, EIUTIRE0002205TC de la Universidad Tecnológica Nacional.

Cabe destacar también que el artículo se enmarca en las actividades planificadas en el Proyecto de Desarrollo Tecnológico y Social (PDTS) presentado, “Aporte a la competitividad de las empresas de desarrollo de Software del NEA”, IP253, evaluado y aprobado por el Consejo Nacional de Investigaciones Científicas y Técnicas de Argentina (CONICET).

6. Referencias

- [1] Bastos Tigre, P., Silveira Marques, F. (2009). “Desafíos y oportunidades de la industria del software en América Latina” Cepal.
- [2] Pino, F., F. Garcia, M. Piattini (2006). “Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas.” Revista Española de Innovación, Calidad e Ingeniería del Software. (REICIS). Vol. 2(1) Abril pp. 6-23.
- [3] “Reporte anual sobre el Sector de Software y Servicios Informáticos de la República Argentina” (2015). Disponible en <http://www.cessi.org.ar/opssi>.
- [4] 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

- [5] 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.
- [6] Pflieger, S. (2002) “Ingeniería de Software. Teoría y Práctica.” Pearson Education.
- [7] Letelier, P., Penadés, P. (2006) “Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)” Técnica Administrativa, Buenos Aires. ISSN 1666-1680
- [8] Alliance, A. (2001). “Agile manifiesto”. Información disponible en <http://www.agilemanifiesto.org>
- [9] Rujana, M., Romero Franco, N., Tortosa, N., Tomaselli, G., & Pinto, N. (2016, May). Análisis sobre adopción de metodologías ágiles en los equipos de desarrollo en pymes del NEA. In *XVIII Workshop de Investigadores en Ciencias de la Computación (WICC 2016, Entre Ríos, Argentina)*.
- [10] 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.
- [11] 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).
- [12] Córdoba, J., Cachero, C., Calero, C., Genero, M., & Marhuenda, Y. (2007, October). Modelo de Calidad para Portales Bancarios. In *XXXIII Conferencia Latinoamericana de Informática (CLEI'07)*.
- [13] Behkamal, B., Kahani, M., & Akbari, M. K. (2009). Customizing ISO 9126 quality model for evaluation of B2B applications. *Information and software technology*, 51(3), 599-609.
- [14] ISO/IEC 9126: “Software Engineering - Product quality”, International Organization for Standardization, 2000.
- [15] 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.
- [16] Información disponible en <http://www.agilemanifiesto.org/iso/es/> Último acceso 06/2016
- [17] International Organization for Standardization. (2000). ISO 9001: 2008: Quality Management Systems-Requirements. International Organization for Standardization.
- [18] Cohen, D., Lindvall, M., & Costa, P. (2003). Agile software development. DACS SOAR Report, 11.
- [19] Mahnic, V., & Zabkar, N. (2008). Using COBIT indicators for measuring scrum-based software development. *WSEAS Transactions on Computers*, 10(7), 1605-1617.
- [20] Paulk, M. C. (2001). Extreme programming from a CMM perspective. *Software, IEEE*, 18(6), 19-26.
- [21] Piattini, Oktaba, Orozco, “COMPETISOFT. Mejora de procesos software para pequeñas y medianas empresas”, Editorial Ra-Ma, Año 2008.

- [22] D. Kane and S. Ornburn, "Agile Development: Weed or Wildflower?" CrossTalk, The Journal of Defense Software Engineering, <http://www.stsc.hill.af.mil/crosstalk/2002/10/kane.html>, 2002. (1.3.2006)
- [23] J. Nawrocki, W. Bartosz, and A. Wojciechowski, "Toward Maturity Model for eXtreme Programming," In proceedings of the 27th Euromicro Conference, pp. 233-239, 2001.
- [24] M. C. Paulk, "Extreme Programming from a CMM Perspective," Software, vol. 18, issue 6, pp. 19-26, 2001
- [25] R. Turner and A. Jain, "Agile Meets CMMI: Culture Clash or Common Cause," In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe, pp. 153-165, 2002.