

Modelo Conceptual basado en SQuaRE: un aporte para la especificación de requisitos de calidad del software en entornos de desarrollos ágiles

Conceptual Model based on SQuaRE: A Contribution for the specification of Software Quality Requirements in Agile Development Environments

Presentación: 26 y 27 de octubre de 2022

Juan P. Mansilla

Universidad Tecnológica Nacional – Facultad Regional San Francisco
juampimansilla17@gmail.com

Gastón Ferreyra

Universidad Tecnológica Nacional – Facultad Regional San Francisco
gastonferreyra49@gmail.com

Javier Saldarini

Universidad Tecnológica Nacional – Facultad Regional San Francisco
saldarinijavier@gmail.com

Resumen

Existen distintas metodologías utilizadas para guiar el desarrollo del software, de manera general se las podría clasificar en dos grandes grupos, las llamadas metodologías tradicionales (Cascada, En V, RUP, RAD, Espiral, etc.) y las denominadas metodologías ágiles (XP, Scrum, Crystal, etc.).

Sea cual fuere la metodología escogida, una de las primeras y principales etapas en el desarrollo del software es la identificación de los requisitos por parte de los clientes/usuarios, para esta actividad cada metodología propone diferentes técnicas y/o artefactos para lograr lo que se denomina una especificación de requisitos.

En la mayoría de los proyectos de desarrollo de software los mayores esfuerzos parecen estar centrados en el descubrimiento, especificación, codificación y seguimiento de los requerimientos funcionales, sin embargo, no parece darse el mismo tratamiento a los requerimientos no funcionales o también conocidos como requerimientos de calidad.

Visto lo planteado anteriormente, es que surge el interés de desarrollar una propuesta que permita identificar los aportes que se puedan realizar desde la perspectiva de calidad de producto software con el propósito de facilitar la identificación y especificación de requisitos relacionados a la calidad del software en el ámbito de las metodologías ágiles.

Por ello, y para alcanzar el propósito mencionado, es que este trabajo propone como primera instancia la construcción de un modelo conceptual que permita visualizar las relaciones y los aportes que desde la perspectiva

de calidad de productos software, específicamente tomando como marco de referencia lo establecido en SQuaRE, se puedan realizar en las etapas de especificación de requerimientos cuando es utilizado el marco de trabajo SCRUM.

Palabras clave: Metodologías Ágiles. Calidad de Software

Abstract

There are different methodologies used to guide software development, in general they could be classified into two large groups, the so-called traditional methodologies (Cascade, V-shaped, RUP, RAD, Spiral, etc.) and the so-called agile methodologies (XP, Scrum, Crystal, etc.).

Regardless of the chosen methodology, one of the first and main stages in software development is the identification of requirements by customers/users. For this activity, each methodology proposes different techniques and/or artifacts to achieve what is required. called a requirements specification.

In most software development projects, the greatest efforts seem to be focused on the discovery, specification, coding and monitoring of functional requirements, however, the same treatment does not seem to be given to non-functional requirements or also known as requirements. quality.

In view of the above, it is that the interest arises to develop a proposal that allows identifying the contributions that can be made from the perspective of software product quality with the purpose of facilitating the identification and specification of requirements related to software quality in the field of agile methodologies.

For this reason, and to achieve the aforementioned purpose, this work proposes as a first instance the construction of a conceptual model that allows visualizing the relationships and contributions that from the perspective of quality of software products, specifically taking as a reference framework what is established in SQuaRE, they can be done in the requirements specification stages when the SCRUM framework is used.

Keywords: Agile methodologies. Software Quality

Introducción

Las metodologías ágiles para el desarrollo de software surgieron en la década de los noventa, están basadas en el denominado Manifiesto Ágil (Manifiesto for Agile Software Development, s.f.), que establece los siguientes valores: Individuos e interacciones, software funcionando, colaboración con el cliente y respuesta al cambio.

Las principales características de las metodologías ágiles están centradas en el desarrollo iterativo e incremental, las entregas frecuentes, la priorización de los requisitos, la constante interacción con el cliente, la adaptación al cambio, el trabajo colaborativo en equipo, etc. Autores como Sommerville (Sommerville, 2011) y Pressman (Pressman, 2005) coinciden en que las más reconocidas en la industria del software son: Scrum (Scrum Alliance, s.f.), Programación Extrema (Extreme Programming, s.f.), Crystal (CRYSTAL, Agile project management, s.f.), entre otras.

Por otra parte, según un estudio realizado por la Scrum Alliance (Scrum Alliance, s.f.), Scrum es la metodología ágil más utilizada actualmente; alrededor del 95% de los encuestados, aseguran que utilizan prácticas de Scrum en la gestión de proyectos de software ágiles.

Según menciona Sommerville (Sommerville, 2011) el nuevo software se desarrolla rápidamente para aprovechar las actuales oportunidades, con la finalidad de responder ante la amenaza competitiva. En consecuencia, en la actualidad la entrega y el desarrollo rápidos son por lo general el requerimiento fundamental de los sistemas de software. De hecho, muchas empresas están dispuestas a negociar la calidad del software y el compromiso con los requerimientos, para lograr con mayor celeridad la implementación que necesitan del software.

Si hablamos de calidad de software, se puede observar que en la bibliografía existen diversas definiciones, por ejemplo, en (Pressman, 2005) se la define como “el cumplimiento de los requisitos de funcionalidad y desempeño

explícitamente establecidos, de los estándares de desarrollo explícitamente documentados, y de las características implícitas que se espera de todo software desarrollado profesionalmente”.

También se observa que, en ISO/IEC 25000 (ISO/IEC 25000:2014, 2014), se la define como el “grado en que el producto software satisface las necesidades expresadas o implícitas, cuando es usado bajo condiciones determinadas”.

La calidad del software puede ser abordada desde distintos enfoques, éstos pueden ser: calidad a nivel proceso, calidad a nivel de producto y calidad en uso. Para cada uno de estos enfoques, existen distintos tipos de modelos de calidad de software, lo cuales pueden aplicarse de acuerdo a lo descripto en (Villalta & Carvallo, 2015).

Visto el marco descripto anteriormente, es que, el presente trabajo propone como primera instancia la construcción de un modelo conceptual que permita visualizar los aportes que desde la perspectiva de calidad de productos software, específicamente tomando como marco de referencia lo establecido en SQuaRE, se puedan realizar en las etapas de especificación de requerimientos cuando es utilizado el marco de trabajo Scrum.

Familia de Normas ISO/IEC 25000

En esta sección se presenta un desglose general de las divisiones y normas que forman parte de SQuaRE (del inglés, Systems and software Quality Requirements and Evaluation) o también conocida como la familia de normas ISO/IEC 25000 (ISO/IEC 25000:2014, 2014), la misma es una serie organizada lógicamente, enriquecida y unificada, la cual permite cubrir dos procesos principales:

1-Especificación de requerimientos de la calidad de software.

2-Evaluación de la calidad del software, apoyados por un proceso de medición de la calidad de software.

El propósito de SQuaRE consiste en brindar ayuda al desarrollo y adquisición del producto de software, a través de la especificación y evaluación de requerimientos de calidad. Establece criterios para la especificación de requerimientos de la calidad del producto de software, su medición y evaluación. Además, incluye un modelo de la calidad que permite alinear las definiciones de la calidad del cliente, con atributos del proceso de desarrollo.

La serie proporciona también un conjunto de medidas recomendadas para los atributos de calidad del producto de software, las cuales pueden ser utilizadas por desarrolladores, adquirientes y evaluadores (ISO/IEC 25010:2011, 2011).

A continuación, se realiza una breve descripción acerca del conjunto de normas que integran la SQuaRE.

- **ISO/IEC 2501n:** División del Modelo de la Calidad: en esta división se presentan los modelos detallados de calidad para sistemas y productos de software, calidad en el uso y calidad de datos.

- **ISO/IEC 2502n:** División de la Medición de la Calidad: las Normas Internacionales que conforman esta división incluyen un modelo de referencia de medición de la calidad de sistemas y del producto de software, definiciones matemáticas de medidas de la calidad, y directrices prácticas para su aplicación.

- **ISO/IEC 2503n:** División de los Requerimientos de la Calidad: esta división ayuda en la especificación de los requerimientos de calidad.

- **ISO/IEC 2504n:** División de la Evaluación de la Calidad: en esta división se proporcionan requisitos, recomendaciones y directrices para la evaluación del producto, ya sea que la misma sea realizada por evaluadores independientes, adquirientes o desarrolladores.

Marco de trabajo Scrum

Según se detalla en (Manifiesto for Agile Software Development, s.f.) Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.

En pocas palabras, Scrum requiere un Scrum Master para fomentar un entorno donde:

1. Un Product Owner ordena el trabajo de un problema complejo en un Product Backlog.
2. El Scrum Team convierte una selección del trabajo en un Increment de valor durante un Sprint.
3. El Scrum Team y sus interesados inspeccionan los resultados y se adaptan para el próximo Sprint.
4. Repita

El marco de trabajo Scrum es incompleto de manera intencional, solo define las partes necesarias para implementar la teoría de Scrum. En este marco de trabajo pueden emplearse varios procesos, técnicas y métodos. Scrum hace visible la eficacia relativa de las técnicas actuales de gestión, entorno y trabajo, de modo que se puedan realizar mejoras.

En (Estayno & Meles, 2014) se expresa que, para introducir los requerimientos en el proceso de desarrollo, las *historias de usuario* son una práctica común en los métodos ágiles. A diferencia del enfoque de la ingeniería de requerimientos tradicional, las *historias de usuario* estimulan un diálogo fluido y enriquecedor entre clientes y el equipo de desarrollo, durante la construcción del software, buscando la mejor solución.

En el marco de la metodología ágil XP (Extreme Programming, s.f.) es que surgen las *historias de usuario*, las cuales hoy son muy utilizadas por Scrum (Scrum Alliance, s.f.), para trabajar la identificación de requerimientos y como unidad de estimación, planificación y gestión del trabajo del equipo en el contexto del proyecto.

Modelo Conceptual

Un modelo conceptual o de dominio basado en UML (Booch, Rumbaugh, & Jacobson, 2006) nos permitirá describir la problemática de un dominio determinado, aportando un marco de referencia claro y estructurado para una mejor comprensión del mismo.

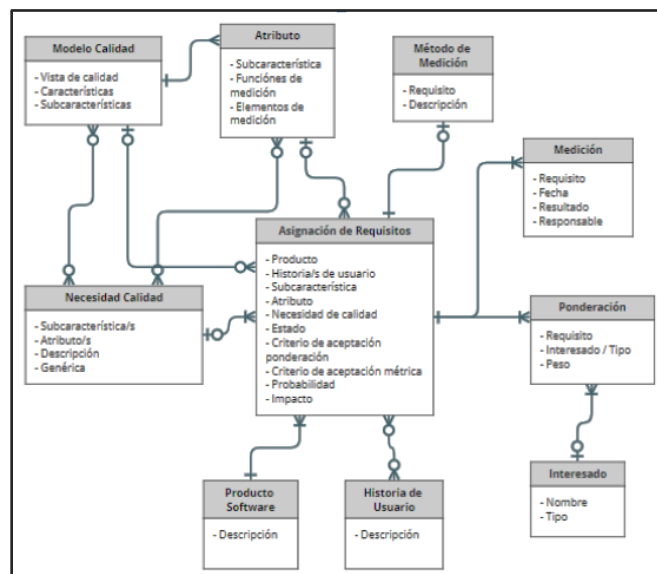


Figura 1. Modelo Conceptual Square-Scrum

El modelo conceptual logrado está enfocado en el marco de trabajo establecido por Scrum, específicamente sobre las *historias de usuario*, su construcción se realiza desde la perspectiva de calidad de producto de software utilizando como referencia lo propuesto por Square. En la Figura 1 se puede observar el modelo desarrollado.

Para la construcción del Modelo de este trabajo se siguieron los siguientes pasos:

1- Definición y delimitación del dominio.

El dominio para el cual se desarrolla el modelo está enfocado en el ámbito de la calidad del software y de manera específica se delimita a la especificación de requisitos de calidad de producto software, tomando como referencia el marco de trabajo Scrum y a Square para establecer la perspectiva de calidad de producto software.

2- **Identificación y definición de las principales clases conceptuales.**

- 1- **Producto Software:** Productos de software cuyos requisitos son relevados a través de un product backlog.
- 2- **Historia de Usuario:** descripción de una función de software escrita desde el punto de vista del cliente.
- 3- **Interesado:** Las partes interesadas de un sistema incluyen todas las personas (por ejemplo, los usuarios finales), las organizaciones (por ejemplo, las organizaciones de los usuarios finales o las organizaciones de desarrollo) y los organismos (por ejemplo, las autoridades legales y reglamentarias o el público en general) que tengan un interés legítimo en el sistema. Las partes interesadas tienen diferentes necesidades y expectativas en el sistema. [ISO/IEC 25030] (ISO/IEC 25030:2007, 2007)
- 4- **Modelo de Calidad:** La calidad de un sistema es el grado en el cual el sistema satisface las necesidades establecidas e implícitas de sus diversas partes interesadas, y por lo tanto, aporta valor. Estas necesidades establecidas e implícitas son representadas en la serie SQuaRE de Normas por modelos de calidad que categorizan la calidad del producto en características, que, en algunos casos, son luego subdivididas en subcaracterísticas. [ISO/IEC 25010] (ISO/IEC 25010:2011, 2011)
- 5- **Atributo:** propiedad o característica inherente de una entidad que puede distinguirse cuantitativa o cualitativamente mediante medios humanos o automatizados. [ISO/IEC 25020]. En este caso, tratándose de atributos de calidad de software, corresponde con aquellas propiedades de la calidad del software que pueden medirse utilizando métricas, compuestas por una función de medición y elementos de medición.
- 6- **Necesidad de Calidad:** son aquellas necesidades de calidad de los interesados en el software. Son el punto de partida para el proceso de definición y análisis de requisitos de calidad de software dados por ISO/IEC 25030 (ISO/IEC 25030:2007, 2007).
- 7- **Asignación de Requisitos:** aquellos requisitos de calidad (subcaracterísticas, atributos o necesidades de calidad) relevados para un determinado producto de software en su totalidad o solo para algunas de sus funcionalidades (representadas por historias de usuario).
- 8- **Ponderación:** para seleccionar un subconjunto de todos los posibles requisitos de calidad, estos deben ser ponderados según la importancia que le den cada interesado o tipo de interesado.
- 9- **Método de Medición:** secuencia lógica de operaciones, descritas genéricamente, utilizada en la cuantificación de un atributo con respecto a una escala especificada. [ISO/IEC 25020] (ISO/IEC 25023:2016, 2016 (E)). Descripción de los pasos a seguir, fuentes de datos, herramientas, configuraciones, etc. necesarios para medir un elemento de medición que conforma la métrica de un atributo de calidad.
- 10- **Medición:** de elementos de medición para obtener el valor de una métrica de calidad en un momento dado del ciclo de vida del software.

Conclusiones

Como una primer prueba de concepto se llevó a cabo la instanciación del modelo desarrollado tomando por ejemplo un determinado desarrollo de *producto de software*, el cual tiene asociado una o varias *historias de usuario*, este *producto de software* tiene asociado *interesados*, a través de los cuales se pueden detectar las *necesidades de calidad* de las partes *interesadas* con el uso de ISO/IEC 25030, estas necesidades luego pueden ser transformadas en una especificación de requisitos de calidad a través de las *características y subcaracterística* de calidad establecidas en el *modelo de calidad* propuesto por ISO/IEC 25010, también será posible asignar *atributos y medidas* a las *subcaracterística de calidad* utilizando, por ejemplo, la norma ISO/IEC 25023. De esta manera podemos ver que es factible arribar una

Asignación de requisitos teniendo en cuenta una perspectiva de calidad de producto de software en el marco de trabajo Scrum.

Del análisis llevado a cabo y el desarrollo del modelo conceptual se puede observar que, si bien, esta es una primera aproximación desde la perspectiva de calidad de producto software hacia el marco de trabajo propuesto por Scrum, se puede evidenciar que desde la familia de normas que componen SQuaRE se pueden llevar a cabo aportes concretos, de manera puntual en las *historias de usuarios*.

Como se mencionó anteriormente esta es una propuesta inicial que propone la introducción de la calidad en el marco de trabajo ágil Scrum desde la perspectiva de calidad de producto software. Con fin de mejorar y validar el modelo propuesto, es que se plantea llevar a delante una serie de pruebas de concepto del modelo desarrollado en proyectos de desarrollo de software reales.

Referencias

- Booch, G., Rumbaugh, J., & Jacobson, I. (2006). *El lenguaje Unificado de Modelado* (2° ed.). (A. Wesle, Ed.) Madrid: Pearson.
- CRYSTAL, *Agile project management*. (s.f.). Obtenido de <http://crystalmethodologies.org>
- Estayno, M., & Meles, J. (2014). El Rol del Product Owner en la definición y validación de las user stories. *Ciencia y Tecnología*(14), 145-162.
- Extreme Programming*. (s.f.). Obtenido de <https://www.agilealliance.org/glossary/xp>
- IEEE .Institute of Electrical and Electronics Engineers Computer dictionary. (1990). Standard 610. Compilation of IEEE Standard Computer Glossaries.
- ISO/IEC 25000:2014. (2014). Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE). ISO-Guide to SQuaRE.
- ISO/IEC 25010:2011. (2011). Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-System and software quality models. ISO.
- ISO/IEC 25023:2016. (2016 (E)). Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality. ISO.
- ISO/IEC 25030:2007. (2007). Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Quality requirements. ISO.
- Manifesto for Agile Software Development*. (s.f.). Obtenido de <http://agilemanifesto.org/>
- Pressman, R. (2005). *Ingeniería de Software*. (6 ed.). Mcgraw-Hill.
- Scrum Alliance*. (s.f.). Obtenido de <https://resources.scrumalliance.org/Article/quick-guide-things-scrum>
- Scrum Alliance*. (s.f.). Obtenido de <https://resources.scrumalliance.org/Article/quick-guide-things-scrum>
- Sommerville, I. (2011). *Ingeniería de Software* (9° ed.). México: PEARSON EDUCACIÓN.
- Villalta, A., & Carvallo, J. (2015). Modelos de calidad de software: Una revisión sistemática de la literatura. *CEDIA*. Maskana.