

Calidad de Productos de Software: Un aporte para Criterios de Aceptación No Funcionales en Historias de Usuario

Software Product Quality: A Contribution to Non-Functional Acceptance Criteria in user stories.

Presentación: 13 y 14 de septiembre de 2023

Javier Saldarini

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

Claudio Carrizo

Universidad Tecnológica Nacional – Facultad Regional San Francisco
cjcarrizo77@gmail.com

Emiliana Funes

Universidad Tecnológica Nacional – Facultad Regional San Francisco
emifunes194@gmail.com

Carlos Salgado

Universidad Tecnológica Nacional – Facultad Regional San Francisco
csalgado@unsl.edu.ar

Alberto Sánchez

Universidad Tecnológica Nacional – Facultad Regional San Francisco
asanchez@unsl.edu.ar

Mario Peralta

Universidad Nacional de San Luis - Facultad de Ciencias Físico-Matemáticas y Naturales
mperalta@unsl.edu.ar

Resumen

En el contexto de las metodologías ágiles para el desarrollo de software, las Historias de Usuario son una práctica común para introducir requisitos del sistema. Con el propósito de determinar si el producto de software que se está desarrollando satisface los requisitos establecidos, se crean ciertos Criterios de Aceptación (funcionales y no funcionales), los cuales representan las condiciones de satisfacción que se deben cumplir para cada Historia de Usuario.

Como se mencionó anteriormente, los Criterios de Aceptación pueden ser de tipo funcional o no funcional, por lo general a estos últimos se los asocia con requisitos de calidad o a restricciones que tendrá el sistema.

En este trabajo se presenta una propuesta para facilitar la especificación de requisitos de calidad para Criterios de Aceptación no funcionales, tomando como perspectiva la calidad de productos de software, y lo establecido en la familia de normas ISO/IEC 25000.

Palabras clave: Calidad, Metodologías Ágiles, Requisitos No Funcionales, Software

Abstract

In the context of agile methodologies for software development, User Stories are a common practice to introduce requirements in the development process. To determine if the software product being developed satisfies the established requirements, certain Acceptance Criteria are created, which can be functional and non-functional, and are the satisfaction conditions that must be met for each User Story.

As mentioned, the Acceptance Criteria can be of a functional or non-functional type, generally the latter are associated with quality requirements or restrictions that the system will have.

This paper presents a proposal to facilitate the specification of quality requirements for non-functional Acceptance Criteria, taking as a perspective the quality of software products and what is established in the ISO/IEC 25000 family of standards.

Keywords: Quality, Agile Methodologies, Requirements non-functional, Software

Introducción

Los Sistemas de Información están presentes en todo tipo de organizaciones, y de hecho, son un componente vital para el éxito de los negocios (Soto, 2014). Según la Norma ISO/IEC 25030 (IRAM ISO/IEC 25030:2019, 2019), en un modelo de sistema, los sistemas informáticos son parte de un SI, y están compuestos por los siguientes 4 elementos: hardware, sistema operativo, software de aplicación y datos.

En (Pliattini, García, García, & Pino, 2018) se menciona la importancia de la calidad de los sistemas informáticos, para la supervivencia de las organizaciones, y para el cumplimiento de los objetivos estratégicos.

En lo que respecta a la calidad del software, existen varias definiciones en la literatura (IEEE .Institute of Electrical and Electronics Engineers Computer dictionary, 1990) (Pressman, 2005) (ISO/IEC 25010:2011, 2011), en las cuales se puede observar que todas ellas mencionan que la calidad del software debería ser el cumplimiento y/o grado de satisfacción respecto de determinadas necesidades y/o requisitos dados, los cuales pueden ser explícitos y/o implícitos, entre otros.

Para el desarrollo del software existen distintas metodologías, 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.) (Restrepo, y otros, 2019).

De manera puntual, las metodologías ágiles para el desarrollo de software surgieron en la década de los noventa, están basadas en el Manifiesto Ágil (Manifiesto for Agile Software Development, s.f.), y sus principales características 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.), entre otras.

Tomando en cuenta la calidad del software en entornos de desarrollo ágiles, en (Estayno & Meles, 2014) se desarrolla una propuesta en la cual, entre otras, se plantea la necesidad de incluir requisitos no funcionales en las historias de usuarios a través de los criterios de aceptación; en este sentido, en (Tona, y otros, 2021) se expresa por ejemplo, que en Scrum, las partes interesadas y los propietarios del producto pueden cambiar los requisitos durante el proceso, y que la calidad del proceso y del producto final sigue siendo uno de los principales retos; también se expresa que la cuantificación de la calidad sigue siendo un desafío.

De manera puntual, vemos que en un caso sólo se limita a plantear la inclusión de requisitos no funcionales en artefactos puntuales del marco ágil, mientras que, en el otro, se plantea una modificación en el marco de trabajo de Scrum, que incluye la incorporación de un rol específico relacionado a la calidad, y se sugiere el uso de un modelo de calidad como el de ISO/IEC 25010 (ISO/IEC 25010:2011, 2011).

Desde la perspectiva de calidad de producto de software y, tomando como referencia lo dispuesto en la familia de Normas ISO/IEC 25000 (ISO/IEC 25000:2014, 2014), el presente trabajo pretende avanzar hacia resultados que estén enfocados en la especificación de requisitos de calidad para Criterios de Aceptación no funcionales.

De manera puntual el principal objetivo de este trabajo es poder desarrollar un modelo conceptual que permita evidenciar las relaciones entre los Criterios de Aceptación no funcionales y la familia de Normas ISO/IEC 25000 y en base a ello poder arribar a una especificación de requisitos de calidad.

Especificación de Requisitos en Métodos Ágiles para el Desarrollo del Software

Historias de usuarios

Según se expresa en (Estayno & Meles, 2014) , las historias de usuario se introducen por primera vez en el contexto de Extreme Programming, y son la base del desarrollo de software incremental; también se menciona

que las historias de usuario son una práctica común en los métodos ágiles, con el propósito de poder introducir los requerimientos en el proceso de desarrollo.

En el contexto del desarrollo de software guiado por el uso de metodologías ágiles, las historias de usuario se han convertido en la técnica más adoptada para el desarrollo y gestión de requisitos de usuario; se las define como una descripción corta de una funcionalidad valuada por un usuario o cliente de un sistema. Las historias de usuario están compuestas por tres aspectos, los cuales fueron definidos por (Jeffries, 2019): Card (Tarjeta), Conversación y Confirmación. Dado el enfoque de este trabajo, desarrollaremos sólo la primera de ellas (Card).

- **Card (Tarjeta):** La tarjeta donde se escribirá la historia de usuario deberá ser del tamaño estándar de un *post-it*, contendrá lo requerido por el usuario o cliente, no una especificación.

Si bien el estilo de redacción de una historia de usuario puede quedar a criterio de quien la redacte, hay tres preguntas que siempre se deberían poder responder: ¿quién se beneficia?, ¿qué se quiere? y ¿cuál es el beneficio?

En la Figura 1, se puede ver el patrón sugerido por (Cohn, 2004), el cual permite garantizar que la funcionalidad esté descrita a un alto nivel y de manera breve.

Como [rol del usuario], **quiero** [objetivo], **para poder** [beneficio].

Figura 1. Patrón de escritura para Historias de Usuario

A continuación, daremos un ejemplo de redacción de una Historia de Usuario.

Como usuario de la aplicación de agenda y reserva de laboratorios de computación, **quiero** reservar un laboratorio, **para poder** hacer uso del mismo en una fecha y horario determinado.

Criterios de Aceptación

En (Estayno & Meles, 2014) se expresa que los criterios de aceptación son condiciones de satisfacción específicos para cada Historia de Usuario, bajos las cuales el producto debe satisfacer los requisitos funcionales y no funcionales.

Por su parte, en (Menzinsky, López, Palacio, & Sobrino, 2022) se menciona que los criterios de aceptación, que a veces se traducen en pruebas, son un excelente lenguaje para detallar requerimientos funcionales, y es por ello que toman una gran importancia en las historias de usuario debido a que facilitan el entendimiento respecto de cómo debe funcionar el producto; y por otra parte, el Dueño del Producto usará los Criterios de Aceptación para tomar decisiones sobre que Historias de Usuario pueden ser aceptadas y darlas por finalizadas.

También, se pone de manifiesto que se suelen reflejar Requisitos No Funcionales en los Criterios de Aceptación de las historias de usuario, y que, si las mismas representan restricciones persistentes, se añaden como elementos nuevos en la definición de hecho (DoD).

Otro aspecto importante de los criterios de aceptación está relacionado con su calidad, y la misma puede ser medida a través del método SMART, definido por (Wake, 2003).

- **Specific**, específicos.
- **Measurable**, medibles.
- **Achievable**, alcanzables.
- **Relevant**, relevantes.
- **Time-boxed**, limitados en el tiempo.

La redacción de los criterios de aceptación puede tomar diferentes formas (Menzinsky, López, Palacio, & Sobrino, 2022) y pueden escribirse en lenguaje natural o utilizar, por ejemplo, el lenguaje denominado Gherkin, un lenguaje creado para las descripciones de comportamiento de software.

Tomando lo descrito anteriormente, y a modo de ejemplo, se redactan a continuación algunos Criterios de Aceptación en lenguaje natural para la Historia de Usuario tomada como ejemplo en el apartado anterior.

Historia de Usuario

Como usuario de la aplicación de reserva de laboratorios de computación, **quiero** reservar un laboratorio, **para poder** hacer uso del mismo en una fecha y horario determinado.

Criterios de Aceptación

- CA1: El usuario deberá estar registrado en la aplicación para poder reservar un laboratorio
- CA2: El usuario no podrá reservar ningún laboratorio que ya esté reservado
- CA3: El usuario solo podrá elegir de una lista lo/s laboratorios que desea reservar
- CA4: El tiempo de respuesta para la búsqueda de laboratorios disponibles no deberá ser mayor a tres segundo
- CA5: El usuario deberá poder acceder a la aplicación desde cualquier navegador (IE, Crome, etc.)
- CA6: La aplicación deberá ser de fácil de usar y de entender por los usuarios
- CA7: La aplicación deberá estar disponible las 24hs todos los días del año
- CA8: La aplicación deberá ser segura respecto de los métodos de autenticación de los usuarios

Requisitos de Calidad - ISO/IEC 25000

La Familia de normas que integran el estándar ISO/IEC 25000 (ISO/IEC 25000:2014, 2014), conocido también como SQuaRE (del inglés, Systems and software Quality Requirements and Evaluation), es una serie organizada lógicamente, enriquecida y unificada, la cual contiene un conjunto de normas, que permiten 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 ayudar 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 calidad que permite alinear las definiciones de la calidad del cliente, con atributos del proceso de desarrollo. 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.

Criterios de Aceptación e ISO/IEC 25000

Dado el marco de este trabajo, según lo descrito en las secciones anteriores, es que se desarrolla un modelo conceptual (Figura 3) con el fin de tener una primera aproximación acerca de las relaciones y aportes que se pueden realizar desde la familia de normas IOS/IEC 25000, hacia los Criterios de Aceptación No Funcionales; de manera puntual, se determina cómo especificar Requisitos de Calidad en los Criterios de Aceptación.

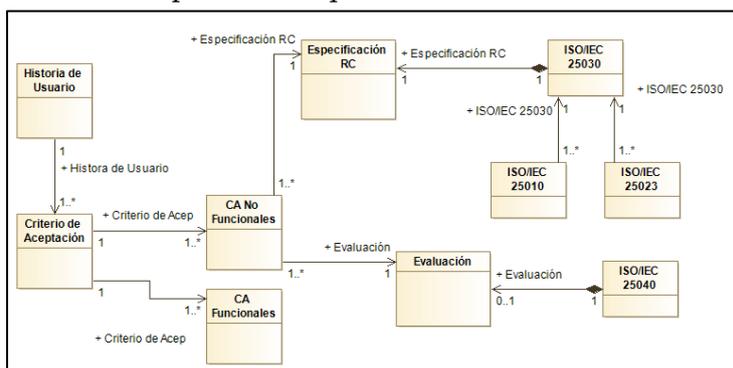


Figura 3. Modelo Conceptual Criterios de Aceptación – ISO/IEC 25000

A través del modelo conceptual desarrollado podemos observar que una **Historia de Usuario** puede tener uno o varios **Criterios de Aceptación**, éstos a su vez están compuestos de **Criterios de Aceptación Funcionales y No Funcionales**; de manera puntual, los **Criterios de Aceptación No Funcionales** tendrán asociada una **Especificación de Calidad**, la cual será una generalización de una especificación guiada por la norma **ISO/IEC 25030**; esta contendrá una o varias características y subcaracterísticas del modelo de calidad de producto **SO/IEC 25010**, y una o muchas métricas correspondientes a la Norma **ISO/IEC 25023**.

A su vez los **Criterios de Aceptación No Funcionales** estarán asociados a una **Evaluación**, la cual es una generalización de lo establecido en la norma **ISO/IEC 25040**.

Como instanciación del modelo, tomaremos el ejemplo que venimos desarrollando, y le incorporaremos los Criterios de Aceptación de Calidad.

Historia de Usuario

Como usuario de la aplicación de reserva de laboratorios de computación, **quiero** reservar un laboratorio, **para poder** hacer uso del mismo en una fecha y horario determinado.

Criterios de Aceptación

- CA1: El usuario deberá estar registrado en la aplicación para poder reservar un laboratorio
- CA2: El usuario no podrá reservar ningún laboratorio que ya esté reservado
- CA3: El usuario solo podrá elegir de una lista lo/s laboratorios que desea reservar
- CA4: El tiempo medio de respuesta para la búsqueda de laboratorios disponibles no deberá ser mayor a tres segundo
- CA5: El usuario deberá poder acceder a la aplicación desde cualquier navegador (IE, Chrome, etc.)
- CA6: La aplicación deberá ser de fácil de usar y de entender por los usuarios
- CA7: La aplicación deberá estar disponible las 24hs todos los días del año
- CA8: La aplicación deberá ser segura respecto de los métodos de autenticación de los usuarios

Como podemos observar en el ejemplo, la Historia de Usuario tiene asociado 8 Criterios de Aceptación, de los cuáles, 5 de ellos son criterios que se podrían clasificar como No Funcionales (CA4, CA5, CA6, CA7 y CA8); para esos Criterios y, utilizando el modelo planteado en la Figura 3, podremos llevar a cabo la Especificación de Requisitos de Calidad para así obtener las características, subcaracterísticas y métricas para cada Criterio de Aceptación.

En la Tabla 1 se pueden observar los Criterios de Aceptación No Funcionales relacionados a sus correspondientes Requisitos de Calidad; por una cuestión de espacio, sólo se muestran 2 Criterios de Aceptación (CA4 y CA5).

Tabla 1. Criterios de Aceptación relacionados Requisitos de Calidad

Criterio de Aceptación	Característica de Calidad	Subcaracterística de Calidad	Métrica	Función de Medición
CA4	Eficiencia del Desempeño	Comportamiento relativo al tiempo	Tiempo medio de respuesta	$X = \sum (A_i) / n$ con $i=1$ a n . Donde A_i = Tiempo que tarda el sistema en responder a una tarea de usuario específica o a una tarea de sistema en la medición i -ésima y n = Número de respuestas medidas.
CA5	Portabilidad	Adaptabilidad	Adaptabilidad del software del sistema	$X = 1 - A/B$ donde A = Número de funciones no completadas o resultados insuficientes para cumplir los requisitos durante el ensayo y B = Número de funciones probadas en diferentes entornos de software del sistema.

Conclusiones

A través del modelo conceptual desarrollado en la Figura 3, y a partir de los resultados parciales obtenidos en la Tabla 1, se puede observar que desde la perspectiva de la calidad de productos de software es factible establecer

relaciones y aportes que nos permitan especificar requisitos de calidad en Criterios de Aceptación No Funcionales.

Si bien lo desarrollado configura un ensayo inicial, se observa que la especificación de requisitos de calidad no solo permite explicitar mucho mejor a aquellos Criterios de Aceptación No Funcionales, sino que también, permite incluir métricas y su función de medición, lo cual colabora para cumplir con el requisito de medición que deben cumplir los Criterios de Aceptación (según se establece en el método SMART).

Dado que los Criterios de Aceptación son uno de los parámetros para tomar conocimiento si el software cumple con requisitos planteados por los usuarios y/o clientes, el poder contar con modelos que permitan especificar requisitos de calidad en los Criterios de Aceptación, permitirá al dueño del producto poder tomar mejores decisiones sobre cuando liberar Historias de Usuarios en la Definición de Hecho.

Referencias

- Cohn, M. (2004). *User stories Applied*.
- 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*. (n.d.). Retrieved from <https://www.agilealliance.org/glossary/xp>
- IEEE .Institute of Electrical and Electronics Engineers Computer dictionary. (1990). Standard 610. Compilation of IEEE Standard Computer Glossaries.
- IRAM ISO/IEC 25030:2019. (2019). Ingeniería de Software – Requisitos de la calidad del producto de software (SQuaRE) – Requisitos de Calidad. Buenos Aires: IRAM.
- 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 25040:2011. (2011). Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Evaluation process,. ISO.
- Jeffries, R. (2019). *Three-C's Revisited*. Retrieved from <https://ronjeffries.com/articles/019-01ff/3cs-revisited/>
- Manifesto for Agile Software Development*. (n.d.). Retrieved from <http://agilemanifesto.org/>
- Menzinsky, A., López, G., Palacio, J., & Sobrino, M. Á. (2022). *Historias de Usuario - Ingeniería de Requisitos Ágil*. Retrieved from https://www.scrummanager.com/files/scrum_manager_historias_usuario.pdf
- Pliattini, M., García, F., García, I., & Pino, F. (2018). *Calidad de Sistemas de Información-4° Edición Ampliada y Actuaizada*. Madrid: RA-MA.
- Pressman, R. (2005). *Ingeniería de Software*. (6 ed.). Mcgraw-Hill.
- Restrepo, S., Vahos-Montoya, J., Gómez-Adasme, M., Pino-Martínez, A., Restrepo-Zapata, E., & Londoño-Marín, S. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Cintex*, 24(2), 13-23.
- Scrum Alliance*. (n.d.). Retrieved from <https://resources.scrumalliance.org/Article/quick-guide-things-scrum>
- Sommerville, I. (2011). *Ingeniería de Software* (9° ed.). México: PEARSON EDUCACIÓN.
- Soto, F. (2014). F. Soto, “Análisis de la problemática asociada con la baja calidad de datos en los sistemas de información,” 2014.
- Tona, C., Juárez, R., Jiménez, S., Quezada, A., Guerra, C., & González, R. (2021). Scrumlity: An Agile Framework Based on Quality Assurance. *9th International Conference in Software Engineering Research and Innovation (CONISOFT)*.
- Wake, B. (2003). *INVEST in good stories, and smart tasks*. Retrieved from <http://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>: