

Modelo de Calidad de Productos Software: Un Plan de Pruebas

Carlos Salgado¹, Mario Peralta¹, Alberto Sanchez¹, Javier Saldarini², Claudio Carrizo², Julio Trasmontana², Silvana Armando²

¹Departamento de Informática Facultad de Ciencias Físico-Matemáticas y Naturales Universidad Nacional de San Luis, Ejército de los Andes 950. San Luis, C.P. 5700.
e-mail: {csalgado, mperalta, alfanego}@unsl.edu.ar

²Universidad Tecnológica Nacional, San Francisco, Córdoba, Argentina.
e-mail: {saldarinijavier, jcarrizo77, julio.trasmontana, silvana.armando}@gmail.com

INTRODUCCIÓN

Una metodología de desarrollo de software aporta un marco de trabajo utilizado para asistir, estructurar, planear y controlar el Proceso de Desarrollo de Software (PDS). La asistencia al PDS se materializa a través de un conjunto de métodos, técnicas y herramientas que, en conjunto, realizan una contribución significativa a la calidad, tanto del proceso como del producto de software resultante de la ejecución del mismo. La propuesta en este trabajo de investigación consiste en definir una estrategia que permita medir la calidad del Producto Software (PS) a través de la prueba de software. Para ello, se definió un modelo de calidad basado en el estándar ISO 25010 junto a un conjunto de métricas e indicadores, complementado con la utilización de testing dirigido por Casos de Uso (CU). Para poder realizar la prueba a través de CU, se propone utilizar CU bien definidos y completos, con un esquema predefinido para la estandarización de los mismos, en el que se detallan los distintos flujos de ejecución del PS.

Pressman y Lowe, en su libro *Web Engineering: A Practitioner's Approach* (Pressman & Lowe, 1991), enuncian y recomiendan un conjunto mínimo de prácticas a adoptar cuando el objetivo sea construir aplicaciones de calidad. Este conjunto de 'Buenas prácticas' recomendadas por Pressman, tienen la virtud de que, además de su efectividad evidente como resultado de su seguimiento, actúan como un elemento motivador para la adopción de una disciplina ingenieril ante la simpleza de su enunciado y la contundencia de los resultados posibles que se desprende de su comprensión.

Para poder evaluar o controlar el desarrollo de software, en cierta medida se necesita contar con un Modelo de Calidad (MC), el cual se debe utilizar como guía de las características esperables en el proceso de desarrollo de distintos tipos de aplicaciones, como ser web/móvil. Los MC son instrumentos específicamente diseñados y contruidos para soportar evaluación y selección de componentes de software. Permiten la definición estructurada de criterios de evaluación, la especificación de requerimientos, la descripción de componentes en relación a ellos y la identificación de desajustes de manera sistemática, facilitando el proceso de evaluación y selección del software (Bermeo Conto, Sánchez, Maldonado, & Carvallo, 2016).

Según Carvallo (Carvallo, Franch, & Quer, 2010), las propuestas existentes de MC se pueden clasificar acorde a si tienen un enfoque de MC fijos, se pueden observar los MC

de McCall (McCall, Richards, & Walters, 1977), (Boehm et al., 1978), entre otros; MC a medida existen diversas propuestas de métodos para crearlos, como la del estándar IEEE 1061 (IEEE, 1998); y MC mixtos, como ADEQUATE Horgan (Horgan, Khaddaj, & Forte, 1999), el MC de Gilb (Gilb, 1988) y el MC propuesto en el estándar ISO/IEC 9126-1 (ISO, 2001), este último actualizado y reemplazado por el estándar ISO/IEC 25010:2011 (ISO, 2011)).

En base a lo expresado, se propone un MC para la evaluación de la calidad de productos software en la Web.

MÉTODOS

La estrategia propuesta, consiste en seguir un MC que permita evaluar la calidad de un PS en la Web. Para ello, se apoya en una buena definición de los CU y la utilización de un proceso de desarrollo web adecuado a las características esperables de ese tipo particular de software. La estrategia consiste de tres fases que se detallan a continuación: (1) *Definir o adecuar el MC de Software*. (2) *Definir o adecuar Métricas e Indicadores*. (3) *Pruebas dirigidas por CU. Tener una sólida y completa definición de los CU*.

El MC que se utiliza, como parte de la estrategia propuesta, se muestra a continuación:

Adecuación Funcional

- Completitud funcional.
 - Flujos Básicos.
 - Flujos Alternativos.
- Corrección funcional.
 - Reglas de Negocio.
 - Reglas de estímulo-respuesta.
 - Reglas de restricción de los objetos (también aplicable a rangos de valores.
 - Reglas de restricción de la aplicación.
 - Reglas de inferencia.
- Pertinencia funcional.

Se utiliza la característica Adecuación Funcional de la norma ISO 25010 (ISO, 2011), y se adoptan las subcaracterísticas propuesta por el estándar y redefinen las propias del MC propuesto. En la fase dos, para cada característica del modelo se definió la correspondiente métrica e indicador.

RESULTADOS

Con el objetivo de validar la estrategia propuesta, y en un marco de colaboración con la industria local, se contactó

con la dirección de una empresa del medio que se encuentra en el proceso de migrar su negocio a la nube y adecuarlo a las nuevas tendencias y su adaptación a las nuevas TI. En este sentido, la dirección de la empresa decidió realizar un control sobre el software involucrado en la transición de la empresa en el nuevo panorama mundial en lo que respecta a los Procesos de Negocio (PN) en los nuevos paradigmas. Así, se tomó la decisión de reestructurar los PN, adaptándolos a las nuevas exigencias tecnológicas.

A pesar de la importancia de realizar esta transición, este proceso trajo aparejado un problema de adaptación para los equipos de trabajo. La migración representó un desafío, pues surgieron distintos requisitos que el software de la empresa debía satisfacer, y ajustar los procesos de validación y verificación para alcanzar el aseguramiento de la calidad. Fue necesario controlar, además, lo referente a aplicaciones, repositorios, etc., externos que surgieron en el nuevo paradigma de la nube. Para tal fin, se reorganizó el trabajo de los recursos humanos con la utilización de una estrategia.

Para este trabajo, se obtuvieron las funcionalidades del sistema a partir de los CU, mientras que los requisitos no funcionales se especificaron a partir de restricciones y reglas de negocio. Para ello, se propuso un MC de PS que tuviera como base la Adecuación Funcional. El MC está compuesto de un conjunto de métricas asociadas a atributos obtenidos a partir de CU bien formados. Estos atributos son: Función esperada, pre condiciones, Trigger y Postcondiciones como resultado esperado, Alternativas, Variaciones, Inclusiones, Extensiones y se depuran pre y post condiciones. A partir de ellos, se pueden anticipar: Funciones alternativas, nuevas negaciones de las pre condiciones. Se analizaron los CU de la empresa asociados a las distintas funcionalidades que proveían sus aplicaciones, confrontándolos con el MC propuesto. Los CU analizados resultaron pobres, con insuficiente información o incompletos. A partir de ellos solo fue posible determinar tópicos como Función esperada, Negación de pre-condiciones, validación de trigger, y pos-condiciones como resultado esperado.

Respecto de características de Completitud Funcional, utilizando las métricas propuestas, se pudo determinar que un 90% del total de CU presentaron flujo básico o de éxito, solo un 40% incluye descripciones de flujos alternativos o de excepción y, en algunos de los casos, las pre y pos-condiciones están muy vagamente definidas.

CONCLUSIONES

En el contexto de nuestra investigación, se definió una estrategia de pruebas de software basado en un MC de PS para medir y evaluar la calidad de dichos productos a través de la prueba de software. Para ello, se definió un MC basado en el estándar ISO 25010 junto a un conjunto de métricas e indicadores, complementado con la utilización de testing dirigido por CU. Para poder realizar este tipo de pruebas, fue necesario utilizar CU bien definidos y completos, con un esquema predefinido para la estandarización de los mismos. En dicho esquema, se detallan los distintos flujos de ejecución del PS. Esto fue un factor claro y decisivo para la puesta a punto del software que se desea migrar a la nube.

Con el fin de validar la estrategia propuesta, se la aplicó en una empresa del medio que pretendía migrar sus procesos a la nube. Se observó que la mayoría de los flujos básicos se encontraban definidos. En cuanto a los flujos alternativos, había cierta debilidad en sus definiciones, resintiendo la adecuación funcional. Respecto de las reglas de negocio, había un gran vacío. Esto, hacía que los recursos humanos de la empresa fueran sobre exigidos por falta de información o datos, y en otro momento no tenidos en cuenta.

Esta estrategia permitió organizar los grupos de trabajo y adecuar las distintas formas de trabajar. Poder describir cada uno de los flujos de evento, a partir de la estrategia propuesta, como si fuera una plantilla a completar para los analistas, les permite a los grupos de testing descubrir restricciones propias del negocio que deben ser probadas. Además, los que llevan a cabo las distintas pruebas, estructurales o funcionales, empezaron a tener los datos e informaciones necesarias y con determinada calidad para el armado de los casos de prueba. Conocer las reglas de negocio de la empresa y del mercado donde se encuentra inserta, posibilitó una mejor planificación de la prueba, que redundó en la mejor calidad del PS y calidad de la tarea del recurso humano.

En la continuidad del trabajo, se aplicará el MC, las métricas e indicadores definidos a nuevos casos de estudio, para lograr una mejor validación práctica de las mismas y de la estrategia. Además, se analizará la necesidad de automatizar mediante scripts o herramientas propios, nuevas métricas para evaluar otros aspectos del proceso, como la utilización y distribución de recursos dentro de la estrategia.

REFERENCIAS

- Bermeo Conto, J., Sánchez, M., Maldonado, J. J., & Carvallo, J. P. (2016). Modelos de Calidad de Software en la Práctica: Mejorando su Construcción con el Soporte de Modelos Conceptuales. *CEDIA*.
- Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., Macleod, G. J., & Merrit, M. J. (1978). Characteristics of Software Quality. *North Holland Publishing Company*.
- Carvallo, J. P., Franch, X., & Quer, C. (2010). Calidad de Componentes Software. In R.-M. Ed., *Calidad del Producto Y Proceso Software - Capítulo 10*: Calero, C. / Moraga, M^a A. / Piattini Velthuis, M. G.
- Gilb, T. (1988). Principles of Software Engineering Management. *Addison Wesley*.
- Horgan, G., Khaddaj, S., & Forte, P. (1999). Anessential Views Modelfor Software Quality Assurance. *Project Control for Software Quality, Shaker Publishing*.
- IEEE. (1998). IEEE Std 1061-1998.
- ISO. (2001). ISO/IEC 9126-1:2001. Software Engineering - Software Product Quality -Part 1: Quality Model, Int'l Org. For Standardization, Geneva. In.
- ISO. (2011). ISO/IEC 25010:2011. In *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuARE) -- System and software quality models*.
- McCall, J. A., Richards, P. K., & Walters, G. F. (1977). Factors in Software Quality. *RADC TR-77-369, Vols I, II, III, US Rome Air Development Center Reports NTIS AD/A-049*.
- Pressman, R., & Lowe, D. (1991). Isn't web engineering all about and technology. In *Web engineering: A practitioner's approach* (Sixth ed., pp. 19 -20): McGraw-Hill.