

## METODOLOGÍAS Y HERRAMIENTAS EMERGENTES PARA CONTRIBUIR CON LA CALIDAD DEL SOFTWARE

Dapozo, Gladys N.; Greiner, Cristina; Irrazabal, Emanuel; Medina, Yanina; Ferraro, María; Mascheroni, Agustín, Company, Ana M., Badaracco, Numa H.  
Facultad de Ciencias Exactas y Naturales y Agrimensura. Universidad  
Nacional del Nordeste  
{gndapozo, cgreiner, eirrazabal, mascheroni.yanina}@exa.unne.edu.ar,  
mafferraro@gmail.com, anamacom@gmail.com, numahernan@gmail.com,

### RESUMEN

Este proyecto es una continuación de los proyectos F07-2009 y F10-2013, ambos enfocados en modelos, métodos y herramientas para la calidad del software. En esta propuesta se abordarán temas emergentes en el área de la calidad de software, en particular, aspectos vinculados con la gestión cuantitativa de proyectos y entrega continua en entornos ágiles buscando promover y/o generar métodos y herramientas que contribuyan a mejorar la calidad del proceso y del producto software.

Una vía para la obtención de un producto software de calidad tanto en el desarrollo como en el mantenimiento es una apropiada gestión de la asignación de los pedidos de cambio al RRHH más apropiado. Si esto se combina con una medición de los atributos que permiten tener una medida de la calidad, es posible brindar a los responsables de los proyectos de desarrollo de software una poderosa herramienta para la mejora de la calidad del mismo.

Por otro lado, se está trabajando en el desarrollo de procedimientos para la gestión de las actividades de prueba continua en la disciplina de entrega continua. Esto incluye el análisis comparado de las herramientas actuales de entrega continua y la evaluación de sus características. Se continúa con métodos cuantitativos para estimar esfuerzo de desarrollo y se atiende la problemática de la eficiencia de los sistemas en organismos públicos, mediante la trazabilidad y el enfoque de procesos de negocios.

**Palabras clave:** calidad de software, gestión cuantitativa, entrega continua.

### CONTEXTO

Las líneas de Investigación y Desarrollo presentada en este trabajo corresponden al proyecto PI-17F018 “Metodologías y herramientas emergentes para contribuir con la calidad del software”, acreditado por la Secretaría de Ciencia y Técnica de la Universidad Nacional del Nordeste (UNNE) para el periodo 2018-2021.

### 1. INTRODUCCIÓN

En un mundo globalizado y moderno, donde el software es cada vez más complejo y de gran tamaño [1], llevar adelante un proceso de desarrollo que finalice dentro de los parámetros acordados, a la vez que se entrega un producto de calidad no es una tarea trivial, así como tampoco lo es mantener esa calidad posteriormente.

La calidad del software es una compleja combinación de factores, que varían entre diferentes aplicaciones. Diversos autores como Pressman [2], McCall [3], y estándares tales como la ISO 25010 [4] han tratado de determinar y categorizar los factores que afectan a la calidad del software. Y en general se concluye que la calidad interna del producto software está directamente relacionada con diferentes atributos cuantificables a partir de su código fuente. Un ejemplo de ello son las métricas CK de diseño orientado a objeto [5], o las métricas de Halstead [6].

Asimismo el análisis del código fuente a partir de estas métricas puede ser tenido en

cuenta para describir la complejidad del producto software y la probabilidad de encontrar errores [7].

Una vez identificados los problemas en el código fuente, corregirlos no resulta una tarea trivial, sobre todo en proyectos grandes. Es esencial y deseable una gestión adecuada de los pedidos de cambio que atraviesa un proyecto, desde el momento que es reportado, asignado hasta que se considera finalizado.

El proceso de asignación de recursos humanos de la manera más eficiente posible puede tener un efecto directo en el cumplimiento de parámetros establecidos, evitando pérdidas de tiempo, esfuerzo y calidad del producto.

Encontrar el desarrollador apropiado para un cambio solicitado es una tarea fundamental para su resolución de la manera más eficiente [8]. Sin embargo, resulta una tarea difícil de automatizar y que consume tiempo. Requiere de diferentes habilidades cognitivas, comunicación con miembros del equipo y conocimiento del proyecto, tanto de los aspectos técnicos como organizacionales.

Siguiendo con este mismo enfoque y con el objetivo de lograr una mayor satisfacción del cliente, es crítico para las organizaciones entregar productos de calidad de manera aún más rápida. Esto dio lugar al surgimiento de un nuevo enfoque denominado “Entrega Continua de Software”, más conocido en inglés como Continuous Delivery (CD). En este enfoque los equipos mantienen la producción de software en ciclos cortos de tiempo, asegurando que el producto pueda ser lanzado de manera fiable en cualquier momento [9] [10].

Existe un concepto similar al CD, que es el de Despliegue Continuo, en inglés Continuous Deployment (DC). El DC es una actividad que consiste en lanzar cambios continuamente al ambiente de producción. La principal diferencia se encuentra en la fiabilidad a la hora de lanzar una nueva versión del producto: el DC busca integrar código a producción una, dos, y muchas

más veces en el mismo día, en cambio, el CD se centra en hacerlo con la certeza de que el producto que se está lanzando a producción tiene un alto grado de calidad y se encuentre libre de defectos.

Uno de los principales problemas de estos enfoques estaría en la calidad del producto software. Ésta puede disminuir, dado que, al realizarse los despliegues del sistema con mayor frecuencia, aparecen más defectos en el producto [10]. Por tanto, es esencial desarrollar un enfoque de priorización de los diferentes aspectos en la calidad del producto software, teniendo en cuenta la forma de trabajo actual de las empresas de desarrollo software.

Para adoptar este enfoque, se utiliza el concepto de Tubería de Despliegue (DP - Deployment Pipeline), un estándar para automatizar el proceso de CD [11].

## 2. LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

A continuación, se describen las siguientes líneas de investigación y desarrollo:

### *Estimación*

Una estimación es “una predicción de cuánto tiempo durará o costará un proyecto”, y constituye la base para la planificación de los proyectos. Y en particular, la estimación del esfuerzo es una tarea principal en la gestión de un proyecto software. Se estudiarán los conceptos, métodos y herramientas orientados a la estimación, en particular para ambientes de desarrollo ágil, con el objetivo de desarrollar un modelo de implementación adecuado y una herramienta que lo soporte, teniendo en cuenta la gran cantidad de técnicas ágiles relacionadas con la estimación y los diferentes grados de adopción de agilísimo.

### *Entrega continua*

Se plantea analizar los tipos de pruebas que existen en la actualidad y los enfoques más actuales para agilizar el proceso de pruebas (como comparación de imágenes o pruebas con inteligencia artificial). Una vez hecho este análisis, se procederá con la selección

de las que más se adapten a las necesidades del Despliegue Continuo.

El paso siguiente es la creación de un primer modelo formado por distintas etapas de pruebas basándose en la Tubería de Despliegue. Para validar el modelo, se procederá con el desarrollo de un marco de trabajo que implemente estas pruebas. Se hará una serie de experimentos con este marco para comparar la velocidad de ejecución de las pruebas contra las convencionales. Se propone implementar el modelo en un ambiente real de desarrollo, para analizar los resultados obtenidos. Se someterá el modelo a mejoras, a medida que avance la etapa de implementación.

#### *Gestión de proyectos en el sector público*

Los proyectos de desarrollo software en la Administración Pública requieren cada vez mayor eficiencia, respecto a la optimización en el uso de los recursos involucrados; pero además deben lograr sumar apoyo en la estructura del estado y su cultura organizacional.

Una de las técnicas que pueden ser llevadas adelante para lograr mejorar la eficiencia de los sistemas en este tipo de organizaciones es la trazabilidad de los proyectos. Su importancia se resalta si consideramos que la trazabilidad en la Ingeniería de Software es una práctica de control que ayuda a obtener el producto en el dominio de la solución lo más exacto y fiable posible a las necesidades expresadas por el cliente en el dominio del problema.

Otro mecanismo que se prevé abordar en esta línea de investigación, es la identificación y modelado de procesos de negocios para guiar el diseño de aplicaciones y mejorar la usabilidad de las mismas.

### **3. RESULTADOS OBTENIDOS/ESPERADOS**

Los resultados provienen en parte del proyecto de investigación F010-2013: “Métodos y herramientas para la calidad del software”, finalizado en el año 2017.

A continuación, se indican los antecedentes:

- En [12] se realizó una revisión sistemática de la literatura. En ella, se buscaron propuestas, técnicas, enfoques, métodos, herramientas y otro tipo de soluciones para afrontar los problemas de pruebas en entornos de desarrollo continuo.
- En [13] se llevó a cabo una revisión sistemática con el objetivo de identificar, analizar y sintetizar las técnicas, herramientas y desafíos encontrados en la literatura sobre pruebas de compatibilidad web. Los resultados indicaron que la técnica de prueba más elegida es el análisis visual.
- En [14] se presentó una técnica para automatizar la detección de defectos de incompatibilidad, mediante el uso del procesamiento digital de imágenes.
- En [16] se propuso un patrón de diseño para realizar pruebas automatizadas sobre servicios web de tipo RESTful, basados en los principios de la arquitectura REST y las pruebas unitarias.
- En [16] se propuso una herramienta para la gestión de proyectos de software que integra la asignación y seguimiento de pedidos de cambios con el análisis de métricas aplicadas al código fuente. La información que la misma ofrece permitirá a los responsables y participantes del proyecto tener una visión amplia de los avances y retrasos del proyecto, y a la vez, un panorama de la calidad del mantenimiento correctivo y adaptativo en función de la medición del código fuente y los valores umbrales definidos.

En cuanto a Resultados Esperados, se trabaja actualmente en:

- Propuesta de estimación de software en contextos de desarrollo ágil.
- Elaboración de un marco de trabajo para guiar el diseño de aplicaciones, basado en el modelado de procesos de negocios para mejorar la usabilidad de las

aplicaciones en el contexto de un organismo de la administración pública provincial.

#### 4. FORMACIÓN DE RECURSOS HUMANOS

En el Grupo de Investigación sobre Calidad de Software (GICS) están involucrados 6 docentes investigadores, 1 tesista de doctorado, 1 becario de investigación de pregrado y 2 tesistas de maestría que finalizaron el cursado en el 2017. Un alumno de grado finalizó la carrera Licenciatura en Sistemas de Información con un trabajo vinculado al proyecto.

#### 5. REFERENCIAS

- [1] M. Lehman, "On Understanding Laws, Evolution and Conservation in the Large Program Life Cycle", *J. of Sys. and Software*, vol. 13, pp. 213-221, 1980.
- [2] R. Pressman, *Ingeniería de Software. Un Enfoque Práctico*, Sexta ed., McGraw Hill, 2005.
- [3] J. A. McCall, P. K. Richards y G. F. Walters, *Factors in Software Quality*, 1997.
- [4] International Organisation for Standardisation, *ISO/IEC 25010, Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuRE) -- System and software quality models*; Geneva, Switzerland, 2011.
- [5] S. Chidamber y C. Kemerer, «A metrics suite for object oriented design,» *Software Engineering, IEEE Transactions*, vol. 20, n° 6, pp. 476 - 493, 1994.
- [6] H. H. Halstead, *Elements of Software Science*, Nueva York: Elsevier Science Inc., 1977.
- [7] R. Malhotra y A. Jain, «Software Fault prediction for Object Oriented Systems: A Literature Review,» *ACM SIGSOFT Software Engineering Notes*, vol. 36, n° 5, pp. 1-6, 2011.
- [8] Y. C. Cavalcanti, P. A. da Mota Silveira Neto y J. do Carmo Machado, «Towards Understanding Software Change Request Assignment: a survey with practitioners,» de *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, Porto de Galinhas, Brazil, 2013
- [9] L. Chen, "Continuous Delivery: Huge Benefits, but Challenges Too" in *IEEE Software* 03/2015. V. 32(2).
- [10] J. Humble and D. Farley. "Continuous delivery: reliable software releases through build, test, and deployment automation", 1st ed. Boston, US: Pearson Education, 2010.
- [11] O. Prusak. "Continuous Testing: The Missing Link in the Continuous Delivery Process". *Blaze Meter*. 2015. <https://blazemeter.com/blog/continuous-testing-missing-link-continuous-delivery-process>
- [12] Mascheroni, M.A., Irrazábal, E. (in press). *Continuous Testing and solutions for testing problems in Continuous Delivery: A Systematic Literature Review*. *Computación y Sistemas*.
- [13] Sabaren, L; Mascheroni, M.A.; Greiner, C; Irrazábal, E. *Una Revisión Sistemática de la Literatura en Pruebas de Compatibilidad Web XXIII Congreso Argentino de Ciencias de la Computación CACIC 2017*. ISBN: 978-987-4251-43-5. La Plata (Argentina). Fecha: Octubre de 2017
- [14] Mascheroni, M.A., Irrazábal E, Cogolio M.K., *Automatic detection of Web Incompatibilities using Digital Image Processing*. *SADIO Electronic Journal of Informatic and Operation Research*. EJS 16 (1) 29-45 (2017)29
- [15] Mascheroni, M.A., Irrazábal, E. (in press). *A Design Pattern Approach for RESTful tests: A case study*. *IEEE 12th Colombian Computing Congress*.
- [16] Chiapello, Jorge. "Gestión de Pedidos de Cambio y Métricas Orientadas a Objetos". *Proyecto Final de Carrera*.

Licenciatura en Sistemas de  
Información. UNNE. Defendido el 27-  
12-2017.