

Integration of the quality manager with the development model to guarantee the quality of products in an e-learning environment

Krissia Gómez-Román
Information Systems Master Program
Tecnológico de Costa Rica
Email: kgomez@tec.ac.cr

Iván Mata-Ortega
Information Systems Master Program
Tecnológico de Costa Rica
Email: imata@tec.ac.cr

Abstract—The quality assurance process and the involvement of stakeholders through an e-learning software development greatly affect the final user and its academic success associated with these tools. The following study is a work in progress that analyzes three indicators that references the quality of the applications at the TEC-Digital platform implemented through dotLRN a learning management system. The results from the analysis identified three main tools, these tools are associated with higher changes in the production environment, more time invested during the quality stage due to changes revisions and a higher level of incidents reported by users. A solution proposal is presented, based on the integration of the quality manager in a four-step process: integration with requirements workshop, developer testing, joint tests, and comprehensive tests. It is important to mention that once the proposal is implemented, its impact is measured according to the defined indicators.

Keywords—Quality test, quality control, e-learning, TEC Digital, quality manager.

1. Introducción

Una etapa de gran importancia en la producción de software es el aseguramiento de la calidad, sin embargo, en muchas organizaciones no siempre es contemplada y no existe conciencia de su importancia para obtener la satisfacción de los clientes con respecto a los productos brindados.

A nivel conceptual, se define el aseguramiento de la calidad como un proceso que orienta todos sus esfuerzos en el proceso y producto, contemplando desde su diseño hasta su entrega al cliente, la idea principal es obtener productos basados en una serie de especificaciones, definiendo actividades que logren obtener productos de alta calidad.

Con el fin de obtener estos resultados de calidad, es importante analizar diversas estrategias para lograrlo, de esta forma, autores como Serna [1] hacen mención de los métodos formales para verificar el desarrollo de los sistemas de software, involucrando diversas técnicas para verificar en profundidad los sistemas de información.

Estos sistemas formales persiguen que el desarrollo de

software, desde sus etapas iniciales procuren el éxito del proyecto, esto por cuanto, se constituyen en la base que sustenta las subsiguientes fases del ciclo de vida, y porque los errores surgidos en ella tienen gran impacto en los costos del proyecto (Perry, 2006, citado por Serna [1]).

Aunado a los sistemas formales, es importante la alineación de políticas según normas y estándares internacionales que definen pautas con respecto a las mejores prácticas, en este caso concreto para el tema de evaluación, control y aseguramiento de la calidad aplicado a los productos de software.

De esta forma, Angeleri y otros [2] hacen referencia a metodologías y frameworks basados en estándares tales como las normas ISO (International Standardization Organization). Como resultado del análisis de este trabajo se define una metodología para la evaluación del software llamado MyFEPS, así como un modelo de calidad llamado QSAT, el cual establece una estructura por niveles para las características, subcaracterísticas, atributos y métricas.

En concordancia con el trabajo anterior, se encuentra la investigación realizada por Losavio y otros [3], en la cual se presenta un modelo de negocio para la calidad de software en un ambiente de aprendizaje electrónico. El autor presenta características que son propias de este tipo de aprendizaje, tales como comunicación y dependencia de una plataforma de e-learning, clasificando estas últimas en tres etapas: los CMS (Content Management System), LMS (Learning Management System) y los LCMS (Learning Content Management System).

Bajo este modelo, se definen una serie de características para las plataformas de aprendizaje electrónico y basadas en el estándar ISO/IEC 25010, en la cual se contemplan características y subcaracterísticas enfocadas a la usabilidad, disponibilidad, eficiencia, entre otras, que deben ser priorizadas en relación con los sistemas de e-learning.

Finalmente, Echeverría y otros [4], hacen mención al testing como práctica para evaluar la eficiencia en aplicaciones web, haciendo uso y referencia a la norma internacional ISO/IEC 9126, en la cual se describen características para la calidad interna y externa, a su vez, divididas en subcaracterísticas, para lo cual menciona integrar aspectos de rendimiento, carga y estrés, describiendo actividades de identificación del entorno de pruebas, requisitos no funcionales, planificación

y diseño de pruebas, configuración del entorno de pruebas, entre otros.

2. Contexto

El análisis se ubica en el TEC Digital, la cual se constituye en la plataforma de e-learning del TEC (Tecnológico de Costa Rica) y que soporta las actividades de enseñanza-aprendizaje de esta Institución de Educación Superior, por lo cual, las herramientas puestas a disposición de los usuarios finales son de gran impacto para el éxito académico de los estudiantes que las emplean.

Para el proceso de desarrollo, el TEC Digital trabaja bajo una metodología en cascada para las etapas previas al desarrollo de proyectos, con la incorporación de elementos de metodologías rápidas basadas en un marco de referencia SCRUM en la etapa propia de desarrollo. De esta forma, se cuenta con una etapa de análisis de requerimientos, en la cual se incorpora la participación del equipo de comunicación visual, encargado del diseño de interfaces, trabajando en conjunto con el equipo de desarrollo y los posibles clientes o usuarios finales, de donde surge la documentación de requerimientos y el prototipo de la interfaz de la herramienta respectiva.

Para la etapa de desarrollo, las entregas se estiman cada dos semanas, de ahí que se menciona una mezcla de la metodología de cascada y ágil. Concluido el proceso de desarrollo, se inicia con la etapa de control de calidad, en la cual el equipo de gestores se encarga de verificar aspectos de interfaz, integridad, seguridad, usabilidad, entre otros, para garantizar el correcto funcionamiento de la herramienta.

Este proceso de control de calidad se encuentra segmentado en una serie de pasos, descritos en la figura 1, y que comprenden revisión de la documentación, calidad de paquetes, funcionalidad, internacionalización y navegación, seguridad y rendimiento, integridad y compatibilidad y finalmente, la aplicación de un plan piloto. Es importante recalcar que estas actividades de calidad son efectuadas una vez terminado el desarrollo de la herramienta, sin embargo, los gestores de calidad no han estado involucrados en etapas anteriores, por lo cual el nivel de conocimiento de la herramienta y sus requerimientos no siempre es el óptimo, razón por la cual se da inicio al presente estudio.

2.1. Aplicación de métricas

Con el fin de determinar el impacto de las actividades de calidad, se procede a analizar varios indicadores relacionados con un seguimiento posterior a la liberación de las herramientas en ambiente de producción, después de haber sido sometidas a pruebas de calidad por parte de los gestores encargados del área. Para dicho fin, se estable una serie de indicadores:



Figura 1. Diagrama de modelo de Control de Calidad del Software, tomado de [5]

- 1) **Cantidad de incidentes en producción:** corresponde a la cantidad de incidentes reportados por los usuarios finales de la plataforma e-learning, a través del sistema de soporte o mesa de servicios Zendesk, para lo cual se analizan datos del 2017 y primer cuatrimestre de 2018. De este primer análisis, se obtiene un total de 98 incidentes relacionados a 12 aplicaciones específicas, de las cuales, las tres que registran mayor cantidad de incidentes son: evaluaciones (35,71 %), expediente estudiantil (22,45 %), documentos y GAAP (Gestor de Actividades de Aprendizaje) ambos con un 10,20 %. Con esta información, se puede determinar fácilmente las principales herramientas que han generado algún comportamiento erróneo o inesperado para el usuario final, por lo cual, representa un insumo importante tanto para los responsables de gestión como para los encargados del proceso de calidad.
- 2) **Cambios realizados por producto lanzados:** se contabiliza la cantidad de cambios realizados por herramienta en el ambiente de producción. Es importante señalar que estos cambios pueden responder a mejoras a la herramienta, corrección de errores, rectificación de dependencias o estructura de base de datos que no se haya instalado adecuadamente, entre otros. De esta variable, se puede identificar que las herramientas GAAP, evaluaciones y calendario, son las que representan un mayor número de cambios en el ambiente de producción.
- 3) **Tiempo en revisiones de software:** según datos estadísticos obtenidos de revisiones realizadas durante el año 2017 y primer cuatrimestre del 2018, se han realizado 78 pruebas de calidad de software

para aproximadamente un total de 23 proyectos de software para la unidad TEC Digital.

Cabe destacar que para el proceso de pruebas de calidad se han definido cinco categorías para tipificar las correcciones, siendo estas: revisión inicial, revisión de correcciones, revisión de cambio, nuevas funcionalidades y mixta. Del análisis de estos datos se obtiene que la mayor parte de revisiones ejecutadas son por concepto de correcciones, correspondiente a un 34,52 %, seguido por revisiones de nuevas funcionalidades, en un 29,49 %, tal como se observa en la figura 2.

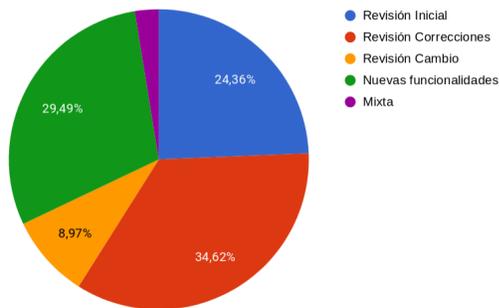


Figura 2. Porcentaje por tipo de revisión, años 2017 y 2018

3. Procedimiento propuesto

Una vez analizado el contexto del TEC Digital y la aplicación de las métricas definidas para analizar el impacto de la situación problemática, se procede a establecer una propuesta con el fin de mejorar el tema de control y aseguramiento de la calidad en los productos de desarrollo de software del TEC Digital.

Para dicho efecto, se recomienda la implementación de una metodología híbrida para el desarrollo de las actividades correspondientes a las pruebas para el software ejecutado en la unidad. Con esto, se busca mantener aspectos de la metodología de trabajo actual, pero con la incorporación de pruebas desarrolladas en ciclos; concretamente se propone una modificación para hacer más robusta la etapa de pruebas de tal forma que se le de al aseguramiento de la calidad la importancia necesaria para producir aplicaciones de calidad. Se detalla la modificación propuesta al modelo de desarrollo actual del TEC Digital, según figura 3.

3.1. Integración con taller de requerimientos

Tal como se expone en el trabajo desarrollado por Gómez y Navas [5], la implementación de las actividades de calidad deben originarse desde etapas tempranas del proyecto, con el fin de procurar la excelencia del producto de software en todas sus etapas, disminuyendo el retrabajo y costos asociados, así como garantizando la satisfacción

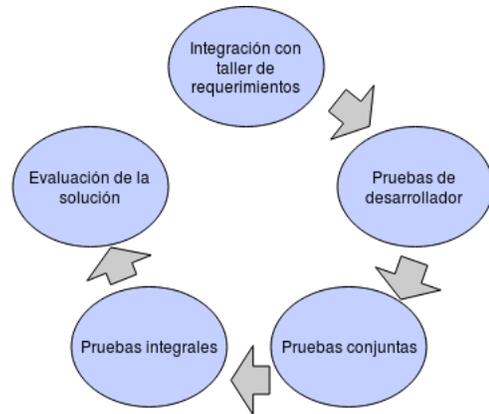


Figura 3. Actividades del procedimiento propuesto

del usuario final.

Para dicho efecto, se propone la inclusión del gestor de calidad desde la etapa de requerimientos, de manera que en conjunto con la definición de historias de usuario e interfaces, se vayan desarrollando posibles casos de prueba para corroborar el cumplimiento de los requerimientos definidos.

Es necesario que durante esta etapa se lleve a cabo una documentación detallada, que permita recabar la información que da inicio al proyecto y que estará asociada a la definición de los casos de prueba inicial que surjan durante la misma.

3.2. Pruebas del desarrollador

Esta etapa del ciclo comprende al equipo de desarrollo, quienes son los responsables de hacer sus propias pruebas antes de migrar al ambiente denominado *release*, el cual es utilizado por el equipo de gestores de la calidad para realizar sus pruebas. Con esto se “garantiza” que las líneas de código escritas por este equipo cumplen los estándares de desarrollo y se presenta como un primer filtro para la captura de errores, aquí el desarrollador genera el plan de pruebas que posteriormente es trasladada a los gestores de calidad.

Con esta etapa se busca identificar y capturar errores por parte del equipo de desarrollo, de esta forma se promueve una mayor calidad en el código desarrollado y en el producto que se va a entregar en el ambiente de pruebas. También se considera, que esta etapa conlleva a un mayor compromiso por parte del equipo técnico con respecto a las características de calidad que deben acompañar al producto final que será entregado a los usuarios.

3.3. Pruebas conjuntas

Esta etapa busca la ejecución de pruebas a través de un equipo líder de desarrollo, se recomienda la integración

de alrededor de tres miembros líderes con experiencia en desarrollo, de manera que sirvan como un punto intermedio antes de migrar la aplicación al ambiente de pruebas. Específicamente, este equipo líder es el encargado de revisar aspectos técnicos y propios del código de desarrollo y cumplimiento de estándares según se define a nivel interno de la unidad.

Una vez que este grupo aprueba el producto en términos de código, el mismo es instalado en el ambiente de pruebas para la realización de las mismas por parte de los gestores de calidad, nótese que de no ser aprobado el código por este equipo, el proyecto se devuelve al desarrollador para su respectiva corrección.

Este equipo no siempre va a ser el mismo, ya que es importante tener diferentes puntos de vista y explotar la experiencia de los diferentes desarrolladores, esto con el tiempo ayudará a optimizar y estandarizar la forma de trabajo del equipo de desarrollo, dándoles mayor confianza y experiencia con el objetivo de generar menos errores de código, disminuyendo el retrabajo y minimizando los tiempos de entrega, lo que implica menor costo para la unidad. Además, es una forma de mantener motivado al equipo de desarrollo ya que se les da un papel más protagónico del que desempeñan habitualmente.

Es indispensable que en este punto se lleve a cabo una documentación detallada que permita gestionar y compartir el conocimiento que el equipo líder va generando, con el fin de ponerlo a disposición del resto del equipo desarrollador y demás miembros que en algún momento pueden formar parte de este equipo de trabajo.

3.4. Pruebas Integrales

Finalmente, este tipo de pruebas involucran a los gestores de calidad, quiénes como lo define Pressman [6], vienen a asumir el rol de representantes del cliente en el interior de la unidad, es decir, su visión y percepción de la herramienta de software debe considerar al usuario final, involucrando análisis de múltiples aspectos, tales como:

- Revisión de la arquitectura del proyecto
- Revisión del prototipo
- Revisión de escenarios de uso
- Revisión del portal del proyecto
- Prueba de sistema y funcional
- Prueba de instalación y de operaciones

Adicionalmente, esta etapa como punto final antes de la liberación del producto de software, debe considerar el cumplimiento de aspectos de la interfaz según el diseño de la misma, integración con otras herramientas o servicios, facilidad de uso y ciertos aspectos de accesibilidad, dentro de otros detalles definidos por el grupo de diseño de interfaces de la unidad.

Es importante recalcar que esta etapa es la que integra los pasos del proceso de calidad descritos en el trabajo de Gómez y Navas [5], y representados en la figura 1.

3.5. Evaluación de la solución

Parte importante de esta propuesta de solución y del cambio a la metodología propuesta es medir su impacto para la organización, pero principalmente para el usuario final. Es importante volver a aplicar periódicamente, los indicadores definidos e identificar otros que ayuden a medir las necesidades del cliente y que conlleven a un sistema de mejora continua.

Es importante establecer métricas que permitan conocer en mayor medida el impacto que la calidad de estas herramientas tienen en el proceso de enseñanza aprendizaje, principalmente en un ambiente e-learning como lo es en el caso de estudio presentado.

4. Conclusiones

Con la presente investigación se logró realizar la definición y aplicación de indicadores que permiten medir la relación existente entre los productos de software liberados a producción y su relación directa con la etapa de control de calidad que se efectúa actualmente.

Se identificaron factores importantes como la cantidad de cambios realizados a herramientas en producción, tiempo invertido en revisiones, principalmente en aquellas de tipo cambio, así como la cantidad de reportes de incidentes de los usuarios finales, logrando establecer las principales herramientas con mayor incidencia en estos indicadores. Se espera que con la integración de la solución propuesta, se garantice el cumplimiento de aspectos no contemplados hasta el momento, como por ejemplo la integración rápida, confiable y eficaz con otras herramientas o servicios de la plataforma, la usabilidad y ciertos aspectos de accesibilidad, dentro de otros detalles que en la actualidad pueden ser pasados por alto durante el ciclo de vida del proyecto por los diferentes involucrados.

Referencias

- [1] E. S. Montoya, "Métodos formales e ingeniería de software," *Revista Virtual Universidad Católica del Norte*, no. 30, pp. 1–26, 2010.
- [2] P. Angeleri, R. Titiosky, A. Sorgen, J. Wuille Bille, and A. Oliveros, "Ajustes al framework de evaluación de productos de software myfeps," in *XVI Workshop de Investigadores en Ciencias de la Computación*, 2014.
- [3] F. Losavio and Y. Esteves, "Modelado del negocio como técnica centrada en la calidad del software para el análisis del dominio del aprendizaje electrónico," in *IV Simposio Científico y Tecnológico en Computación/SCTC 2016/ISBN: 978-980-12-8407-9. Universidad Central de Venezuela, Caracas, Venezuela*, 2016.
- [4] D. E. Perez and A. A. Paumier, "Testing como práctica para evaluar la eficiencia en aplicaciones web," *Revista Latinoamericana de Ingeniería de Software*, vol. 2, no. 5, pp. 307–309, 2014.
- [5] K. Gómez-Román and E. Navas, "Implementation of software quality control in e-learning development projects: Tec digital," in *Learning Technologies (LACLO), 2017 Twelfth Latin American Conference on*. IEEE, 2017, pp. 1–8.
- [6] R. Pressman, "Ingeniería del software uin enfoque práctico, septima edición ed," 2010.