

Estudio sobre realización y documentación de pruebas software

Diego Armando Villarreal Díaz¹, Sonia Cristina Gamboa Sarmiento², Luis Carlos Gómez Flórez³

¹ Estudiante de Maestría en Ingeniería de Sistemas e Informática, Universidad Industrial de Santander, Carrera 27 calle 9, Bucaramanga, Colombia, 68002.

² Profesora de la Escuela de Ingeniería de Sistemas e Informática, Universidad Industrial de Santander, Carrera 27 calle 9, Bucaramanga, Colombia, 68002.

³ Profesor de la Escuela de Ingeniería de Sistemas e Informática, Universidad Industrial de Santander, Carrera 27 calle 9, Bucaramanga, Colombia, 68002.

Autores para correspondencia: diegoarmandovillarreal@gmail.com, scgamboa@uis.edu.co, lcgomezf@uis.edu.co

Fecha de recepción: 28 de septiembre 2015 - Fecha de aceptación: 12 de octubre 2015

RESUMEN

El presente trabajo describe el proceso realizado y resultados obtenidos al estudiar la documentación técnica de los desarrollos de software (DS) del grupo de investigación en sistemas y tecnologías de la información (STI), grupo adscrito a la Universidad Industrial de Santander (UIS). El proceso de revisión seguido se basó en las recomendaciones propuesto por Kitchenham *et al.* (2007) para la revisión sistemática de literatura. Los DS estudiados fueron realizados como parte del proyecto de final de carrera, de estudiantes en Ingeniería de sistemas e informática de la UIS. Se tuvieron en cuenta los trabajos realizados entre 2001 y 2014, siendo revisados un total de 35 proyectos, de los cuales se analizaron los siguientes aspectos: metodológicos, documentación del diseño y pruebas, y coherencia entre la metodología declarada y el proceso seguido. Como resultado de esta revisión, se proponen una serie de recomendaciones que contribuyen a mejorar la calidad de los DS de los GI a través del uso de las pruebas de software.

Palabras clave: Pruebas de software, grupos de investigación, desarrollo software.

ABSTRACT

This paper shows the process and outcomes got by studying the software development (SD) technical documentation at the research group in Sistemas y Tecnologías de la Información (RGSTI), group attached to the Universidad industrial de Santander (UIS). The review process used was based on the recommendations made by Kitchenham *et al.* (2007) for systematic literature review. The DS studied were conducted as part of final degree projects of students in system engineering at UIS. It was considered projects carried out between 2001 and 2014. A total of 35 projects were reviewed and it was analyzed aspects related with the software methodology, design and software testing. As result of this review, it was proposed a set of recommendations to improve the SD quality in RG.

Keywords: Software testing, research group, software development.

1. INTRODUCCIÓN

Son varias las razones que motivan a un grupo de investigación (GI) a realizar algún tipo de desarrollo software (DS), algunas de ellas son: agilizar la realización de cálculos, análisis de datos, simulación de fenómenos, etc. (León & Pimentel, 2011: 141). Desarrollar software no es una tarea fácil. Las grandes compañías desarrolladoras de software invierten una gran cantidad de recursos en el aseguramiento de la calidad. El DS no es la actividad principal de los GI y los recursos con los que

cuentan son bastantes limitados. El grupo de investigación en tecnologías y sistemas de información (STI), grupo adscrito a la Universidad Industrial de Santander (UIS), ha encontrado varias dificultades al usar y realizar mantenimiento a los DS allí realizados, por esta razón, el grupo ha venido trabajando en el aseguramiento de la calidad del software creado en el GI (Martínez *et al.*, 2013; Martínez *et al.*, 2011; León & Pimentel, 2011). El grupo STI ha buscado acercar estándares internacionales tales como las normas ISO 14598, 25000 y 9126 a los DS en GI. Las pruebas software son una herramienta de medición valiosa en el aseguramiento de la calidad (ISO/IEC/IEEE 29119-1, 2013: 22), por esta razón en el año 2012 el grupo STI trabajó en la documentación de las pruebas de software (Martínez *et al.*, 2013). En este trabajo se realiza la revisión de los trabajos de fin de carrera de los estudiantes del STI que cuentan con algún DS. Esto con el propósito de identificar las posibles falencias de este tipo de DS.

2. DESARROLLO SOFTWARE EN GRUPOS DE INVESTIGACIÓN

Un grupo de investigación (GI) es una agrupación de personas que estudian una temática determinada con el propósito de generar resultados de conocimiento (Colciencias, n.d.). Un GI puede necesitar de algún tipo de desarrollo software (DS) como resultado directo o para el apoyo de sus actividades de investigación (León & Pimentel, 2011: 141). El DS no es la actividad central de un GI y por ende los recursos para tal fin son limitados. Es de notar que los DS de los GI son generalmente realizados por estudiantes de últimos niveles como parte de sus trabajos de fin de carrera.

El grupo de investigación en sistemas y tecnologías de la información (STI) es un grupo adscrito a la Universidad Industrial de Santander (UIS) y conoce de primera mano las implicaciones del DS. El grupo posee tres desarrollos bandera: QUIS (Martínez *et al.*, 2011), HSLAB y CYSACJ (Osorio Sanabria *et al.*, 2006). Estos DS surgieron como resultado de las investigaciones llevadas a cabo dentro del grupo. Sin embargo, conforme se ha ido avanzado en las líneas de investigación del grupo, ha sido necesario adicionar nuevas funcionalidades y realizar labores de mantenimiento a estos DS. Lo cual no ha sido una tarea fácil. Por esta razón el grupo STI ha trabajado en el mejoramiento de la calidad de los DS en GI. Para ello ha buscado acercar normas y estándares internacionales, tales como ISO 14598, 25000 y 9126, a los DS en GI (Martínez *et al.*, 2013; Martínez *et al.*, 2011; León & Pimentel, 2011).

3. PRUEBAS DE SOFTWARE

The International Software Testing Qualifications Board (ISTQB) define las pruebas de software como “Proceso que consiste en todas las actividades del ciclo de vida software, tanto estáticas como dinámicas, concernientes con la planificación, preparación y evaluación de productos software y los productos de trabajo relacionados para determinar que éstos satisfacen los requisitos especificados, para demostrar que se ajustan al propósito y para detectar defectos”. (ISTQB, 2011; Márquez Sosa, 2008). Las pruebas de software pueden ser vistas, como un medio para proveer información de las funcionalidades y atributos de calidad del software, así como identificar errores en esos casos donde la prevención de los mismos no fue efectiva (SWEBOK, 2004: 4-2). El objetivo de las pruebas software puede variar: un módulo, un grupo de módulos o un sistema completo (SWEBOK, 2004: 4-5). Se definen cuatro niveles o fases: pruebas de componentes, pruebas de integración, pruebas del sistema y pruebas de aceptación (Baresi & Pezzè, 2006: 105; Graham *et al.*, 2006: 44; ISO/IEC/IEEE 29119-1, 2013: 9). Las técnicas de pruebas de software se pueden clasificar en dos grandes grupos pruebas basadas en especificación también conocidas como pruebas de caja negra y pruebas basadas en la estructura también conocidas como pruebas de caja blanca. La diferencia entre las dos radica en que las pruebas de caja blanca se tienen en cuenta la estructura interna del elemento a revisar y en las pruebas de caja negra no (ISO/ IEC/ IEEE P29119-4/DIS, 2013: 4).

Entre los aportes que las pruebas software pueden brindar a los DS en GI se encuentran:

- 1- Enriquecer la documentación de los DS.
- 2- Brindar evidencia del uso de la aplicación.
- 3- Ver los casos de prueba definidos por el estudiante permitiendo al director del proyecto tener una mejor idea del grado de comprensión del estudiante frente al software a realizar.
- 4- Incrementar la confianza en la calidad del producto final software.
- 5- Le permite al estudiante contemplar escenarios que por su falta de experiencia podría haber pasado por alto.

4. PROCESO DE REVISIÓN

El proceso de revisión seguido partió de las recomendaciones propuestas en (Kitchenham *et al.*, 2007) para la revisión sistemática de artículos científicos. El proceso de revisión consto de seis etapas listadas a continuación:

1. Formulación de las preguntas de investigación
2. Selección de las fuentes de consulta
3. Selección de los estudios de interés
4. Extracción de la información
5. Resumen y resultados

4.1. *Formulación de la pregunta de investigación*

El propósito de esta trabajo es estudiar la documentación disponible de los desarrollo software que resultan de actividades de investigación con el propósito de identificar qué información es documentada con relación al DS y en especial de las pruebas de software. Teniendo esto en mente se formularon las siguientes preguntas de investigación:

- P1: ¿Qué se tiende a documenta de los desarrollos software en un grupo de investigación?
P2: ¿Qué evidencia se encuentra del uso de las pruebas de software en los desarrollos software de un grupo de investigación según la documentación disponible?

4.2. *Determinar la fuente de los documentos*

El software desarrollo por los grupos de investigación de instituciones universitarias suele ser realizado por estudiantes de final de carrera. La principal fuente de información de los trabajos de final de carrera son los libros que los estudiantes entregan. En estos libros se consigna el proceso y resultado obtenidos del desarrollo de sus proyectos. Este documento es un requisito para graduarse y puede ser consultado directamente en la biblioteca de la institución o a través del portal web dispuesto para tal fin (<http://tangara.uis.edu.co/>). El sitio web permite realizar búsquedas por autor, fecha de entrega, título, etc. Por esta razón, se decidió hacer uso de este portal web como la fuente de los documentos a revisar.

4.3. *Seleccionar documentos de interés*

Teniendo en cuenta las preguntas formuladas al inicio del proceso de revisión fueron seleccionadas los trabajos de final de carrera presentado por los estudiantes del grupo STI como fuente primaria de información. La elección del grupo STI surge como resultado de la experiencia con la que cuenta este grupo en el DS. Dado que la búsqueda se realizó a través del sitio web de la biblioteca de la UIS fue necesario definir los siguientes criterios de búsqueda:

- **Periodo de tiempo:** El grupo STI fue fundado en el 2001. Por esta razón, se definió una ventana de tiempo comprendida entre 2001 y 2014
- **Título:** Este campo no fue usado debido a que no siempre hay una relación clara entre el título y el hecho de haber realizado un DS

- **Autor:** El director del grupo STI fue su fundador y quien ha participado de forma continua en la mayoría de los proyectos realizados por el grupo. Por lo anterior, se buscaron los proyectos donde él haya participado como autor o coautor. Con lo que se facilita la detección de trabajos realizados por el grupo STI.
- **Tipo de documento:** El documento debe ser de tipo “Proyecto de grado”.
- **Formato:** El documento debe contar con una versión digital. Este criterio fue necesario para agilizar el proceso de revisión.

4.4. Extracción de la información

Antes de iniciar la revisión, se definieron una serie de categorías para agrupar la información encontrada en cada documento bajo revisión. Teniendo en cuenta que el proceso de revisión buscaba dar respuesta a dos preguntas, las categorías formuladas, también, fueron agrupadas de acuerdo a la pregunta que contribuyen (ver Tablas 1 y 2). Finalmente, La información encontrada fue consignada en hojas de calcular para agilizar su posterior análisis.

Tabla 1. Categorías que contribuyen a la pregunta P1.

| Categoría | Descripción | Criterios de Evaluación |
|---|---|--|
| Metodología de desarrollo software | Se busca conocer el proceso seguido para la construcción del software | - Existencia de una justificación en la elección de la metodología - Evidencia de seguimiento de la metodología |
| Información del diseño de la aplicación | Información que podría ser usada en posteriores DS | - Existencia de diagramas UML - Tipo de diagramas UML usados |
| Uso de la aplicación | Evidencia del uso de la aplicación | - Imágenes de la aplicación final - Evidencia del uso de la aplicación |

Tabla 2. Categorías que contribuyen a la pregunta P2.

| Categoría | Criterios de Evaluación |
|--|---|
| Nivel de las pruebas software | Existencia de pruebas de componentes, de integración, del sistema y aceptación |
| Reporte de errores | Existencia de errores encontrados tras la ejecución de la pruebas |
| Técnicas de pruebas software empleada Evidencia de casos de pruebas | Existencia de pruebas de caja negra y caja blanca Existencia de casos de pruebas |

4.5. Resumen y resultados

Usando los criterios de búsqueda definidos fueron encontrados 51 documentos, de los cuales 35 eran documentos de interés. La información encontrada con la revisión de estos documentos se resume en la Tabla 3.

Tabla 3. Hechos generales encontrados.

| Id | Hecho | # Proyectos | % |
|----|---|-------------|----|
| 1 | Proyectos donde se justifica la selección de la metodología de desarrollo software. | 21 | 66 |
| 2 | Proyectos con evidencia del seguimiento de la metodología de desarrollos software durante su ejecución. | 18 | 56 |
| 3 | Proyectos con imágenes de la aplicación en funcionamiento. | 30 | 94 |
| 4 | Proyectos con evidencia del uso de la aplicación. | 15 | 47 |
| 5 | Proyectos con desarrollos de arquitectura distribuidos. | 29 | 90 |

De los 35 proyectos revisados se encontró que 33 es decir un 94% de los proyectos estudiados, contaban con documentación UML (Tabla 4).

Tabla 4: Uso del UML.

| Hecho | # Proyectos | % con respecto a los 33 |
|--------------------------|-------------|-------------------------|
| Diagrama de casos de uso | 32 | 91% |
| Diagrama de clases | 13 | 39% |
| Diagrama de estados | 13 | 39% |
| Diagrama de secuencia | 12 | 36% |
| Diagrama de paquetes | 9 | 27% |
| Diagrama de actividades | 8 | 24% |
| Diagrama de despliegue | 5 | 15% |
| Diagrama de colaboración | 4 | 12% |

Con relación a las pruebas de software se recolectó información de los niveles de pruebas (ver Tabla 5), técnicas usadas (ver Tabla 6) y algunos hechos adicionales (ver Tbla 7).

Tabla 5. Uso de los niveles de pruebas software.

| Nivel de Pruebas de software | # Proyectos | % |
|------------------------------|-------------|-----|
| Pruebas de componentes | 3 | 9% |
| Pruebas de integración | 4 | 11% |
| Pruebas del sistema | 11 | 31% |
| Pruebas de aceptación | 2 | 6% |

Tabla 6. Uso de las técnicas de pruebas software.

| Técnicas de pruebas de software | # Proyectos | % |
|---------------------------------|-------------|-----|
| Pruebas de caja negra | 19 | 54% |
| Pruebas de caja blanca | 5 | 14% |

Tabla 7: Hechos adicionales

| Generalidades | # Proyectos | % |
|--|-------------|-----|
| Pruebas Unitarias | 0 | 0% |
| Reporte de errores | 11 | 31% |
| Trabajos con evidencia de Casos de pruebas | 13 | 37% |

5. ANALISIS DE RESULTADOS

P1: ¿Qué se tiende a documenta de los desarrollos software en un grupo de investigación?

La información recolectada evidencia que la forma como se documentan los desarrollos software del grupo de investigación bajo estudio no es suficiente ni adecuada, más del 50% de los trabajos no cuentan con evidencia de haber usado algún criterio al momento de seleccionar la metodología de software. En los trabajos donde se usó algún tipo de criterio para elegir la metodología de software un 85% contaban con evidencia de haber seguido dicha metodología. El UML es usado con frecuencia para diseñar y documentar los desarrollos software. Según lo revisado, solo el 15% de los DS mostraban evidencia de haber sido usados por el usuario final.

P2: ¿Qué evidencia se encuentra del uso de las pruebas de software en los desarrollos software de un grupo de investigación según la documentación disponible?

En cuanto al tema de las pruebas de software solo el 37% de los trabajos estudiados contaban con evidencia de haber realizado pruebas de software. En los trabajos donde se realizaron las pruebas de software están fueron realizadas en etapas avanzadas del desarrollo, siendo las pruebas del sistema las más frecuentes. Las pruebas de aceptación fueron usadas con una baja frecuencia, lo cual es coherente con la poca evidencia encontrada en el uso de la aplicación por parte del usuario final.

6. CONCLUSIONES

Un factor que afecta en gran medida los DS en GI es la inexperiencia de los estudiantes en el DS, lo que se evidencia en los bajos índices de seguimientos de metodologías de software, realización de pruebas de software y el uso parcial del UML. Es necesario poner más atención a la forma como se documentan los DS en GI, con forma las líneas de investigación del GI se van desarrollando estos DS tendrán que ser extendidos para lo cual una buena documentación es clave. Además, un buen proceso de documentación no solo contribuye al grupo sino también a la formación de los estudiantes de final de carrera los cuales van a ver mejorada sus competencias como profesionales.

Se debe fomentar el uso de las pruebas de software en los DS de los GI, las pruebas de software son una forma efectiva de detectar problemas y de esta forma mejorar la calidad de los DS realizados en GI. Además, el proceso de las pruebas de software no es una actividad única del software, la idea de verificar y validar con conceptos que están inmerso el desarrollo de actividades científicas. Son varios los beneficios que ofrecen las pruebas de software a los DS en GI, entre los que se encuentra: promover el uso del software desarrollado en grupos de investigación con la creación de casos de prueba que pueden ser ejecutados por miembros externos al proyecto; y contribuir a que los estudiantes comprendan y sigúan la metodología seleccionada. Esto último con la definición casos de pruebas de acuerdo a la etapa en la que se encuentre el DS.

Finalmente, si bien este trabajo fue realizado solo en un GI los problemas encontrados son ampliamente conocidos en la industria del software sin embargo un GI no es igual a una gran empresa desarrolladora de software lo cual abre la puerta a futuros trabajos de investigación orientados a resolver los problemas a los que los GI se enfrentan al realizar un DS.

AGRADECIMIENTOS

Este estudio fue patrocinado por Colciencias y su programa de jóvenes investigadores. Los autores agradecen de forma especial a la Universidad Industrial de Santander (UIS) y al grupo de investigación en tecnologías de la información (STI).

REFERENCES

- Baresi, L., M. Pezzè, 2006. An introduction to software testing, *Electronic Notes in Theoretical Computer Science*, 148(1), 89-111.
- Colciencias, (n.d.). *Colciencias*. Disponible en <http://www.colciencias.gov.co/faq/que-es-un-grupo-de-investigaci-n>.
- Graham, D., E. Van Veenendaal, I. Evans, R. Black, 2006. *Foundations of software testing*. Disponible en http://www.computing.dcu.ie/~ray/teaching/CA358/dorothy_graham.pdf, 207 pp.
- SWEBOK, 2004. *Chapter 4: Software testing*. Disponible en http://swbokwiki.org/Chapter_4:_Software_Testing, 19 pp.

- IEEE Computer Society, 2004. *Software Engineering 2004 Degree Programs in Software Engineering*. Disponible en <http://sites.computer.org/ccse/SE2004Volume.pdf>, 135 pp.
- ISO/IEC/IEEE P29119-4/DIS, 2013. *Draft IEEE Standard Systems and software engineering - Software testing - Part 4: Test techniques*. Disponible en <http://standards.ieee.org/findstds/standard/29119-4-2015.html>.
- ISO/IEC/IEEE 29119-1, 2013. *Software and systems engineering - Software testing - Part 1: Concepts and definitions*. Disponible en <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6588537>, 6 pp.
- ISTQB, 2011. *Foundation level syllabus*. Disponible en <http://www.istqb.org/downloads/finish/16/15.html>, 78 pp.
- Kitchenham, B., E. Mendes, G.H. Travassos, 2007. *A systematic review of cross- vs. within-company cost estimation studies*. IEEE Transactions on SE 33(5), 316-329.
- León, N.E., J.L. Pimentel, 2011. Herramienta computacional para la gestión y evaluación de proyectos software enmarcados en actividades de investigación. *Scientia et Technica*, XVI(47), 141-146.
- Martinez, N.E.L., J.E. Pimental Ravelo, L.C. Gómez Flórez, 2011. Herramienta computacional para la gestión y evaluación de procesos software enmarcados en actividades de investigación. *Scientia et Technica*, XVI, 3(49), 134-139.
- Martinez, N.E.L., J.L. Aguilar, E.F. Vega Morales, L.C. Gomez Florez, 2013. Herramienta computacional para la documentación de pruebas de software enmarcado en actividades de investigación Computer tool for software testing documentation under research activities. *Scientia et Technica*, 18(4), 682-689.
- Márquez Sosa, G., 2008. *Glosario estándar de términos utilizados en pruebas software*. Disponible en http://www.sstqb.es/ficheros/sstqb_file94-5205ea.pdf, 72.
- Osorio Sanabria, M.A., M.P. Gonzáles Zabala, L.C. Gómez Flórez, 2006. *Sistema de informacion para apoyar el control de las actuaciones de los estudiantes del consultorio juridico de la uis cysacj-uis*. Disponible en <http://docplayer.es/3607230-Sistema-de-informacion-para-apoyar-el-control-de-las-actuaciones-de-los-estudiantes-del-consultorio-juridico-de-la-uis-cysacj-uis.html>, 143 pp.
- UIS, 2013a. *Plan de Estudios del Programa Académico de Ingeniería de Sistemas*. Disponible en <http://www.uis.edu.co/webUIS/es/academia/facultades/fisicoMecanicas/escuelas/ingenieriaSistemas/programasAcademicos/ingenieriaSistemas/planEstudios.html>.
- UIS, 2013b. *Contenido Ingenieria del Software I*. Disponible en <http://www.uis.edu.co/webUIS/es/asignaturas/contenidoAsig.jsp?codigo=22969>).
- UIS, 2013c. *Contenido Ingenieria del Software II*. Disponible en <http://www.uis.edu.co/webUIS/es/asignaturas/contenidoAsig.jsp?codigo=22973>).