

Metodología para la evaluación de calidad de los productos software de la Universidad Cundinamarca

Autor¹, Autor², Autor³

1 Estudiante de pregrado en Ingeniería de sistemas. Universidad de Cundinamarca Facatativá. **2** Estudiante pregrado en ingeniería de sistemas, Universidad de Cundinamarca Facatativá **3** Ingeniera de Sistemas y especialista en gestión de proyectos. Docente TCO universidad de Cundinamarca
correo1@gmail.com, correo2@gmail.com, correo3@gmail.com.

Resumen— La metodología que se va a asociar pretende evaluar la calidad de los desarrollos web de la Universidad Cundinamarca, con el fin de que estén sincronizados con los estándares internacionales como la ISO/IEC/IEEE 29119 y se documenten bajo el lenguaje grafico de UML 2.0.

Palabras clave— Calidad, Diagramas, Estándares, Gestión de pruebas, Herramientas libres, Investigación, Metodología, Métricas, Modelado, Testeo.

Abstract— the methodology to be associated aims to evaluate the quality of web developments of Cundinamarca University, with the purpose to synchronize them with International standards such as ISO/IEC/IEEE 29119 and documented it under the graphical language UML 2.0.

Keywords— Free tools, Metrics, Diagrams, Methodology, Modeling, Quality, Research, Standards, Test Management, Testing.

I. INTRODUCCIÓN

Según el informe dado por Software Engineering Institute (SEI) Colombia se encuentra en el primer lugar a nivel Latinoamérica en producción de software de nivel cinco destacándose de esta manera en el desarrollo de software de calidad. (Mintic, 2015). [1]

Por lo tanto la Universidad de Cundinamarca se centra en que sus desarrolladores se enfoquen en proyectos informáticos orientados a

tecnología web los cuales deben ser diseñados y construidos con estándares internacionales de calidad, de este modo se propone esta metodología que abarca dicha evaluación partiendo desde el modelado, hasta las pruebas funcionales. En el planteamiento de esta se tuvo en cuenta la más reciente actualización de evaluación de calidad de software la ISO/IEC/IEEE 29119 la cual hace una recopilación de los criterios de calidad presentes en la ISO 9126 que está orientada a la funcionalidad y la satisfacción del cliente; en la ISO 25000 se realiza un compendio de la 9126 y 14598 dirigida directamente al evaluador. Basado en estas características dicho estándar hace una valoración de la calidad encaminada al testeo.

Como seguimiento a esta actividad se estudian distintas pruebas que se recomiendan realizar al software. También se tiene en cuenta el modelado uml2.0, con el fin de construir una metodología íntegra.

Se busca que los evaluadores cuenten con una guía y un grupo de herramientas que faciliten la valoración de los distintos aplicativos.

De esta manera los estudiantes, desarrolladores, arquitectos de software, directores de proyectos se enfoquen en generar criterios enfocados a la calidad de software durante todo el ciclo de vida del producto tecnológico.

II. ¿QUÉ ES UNA METODOLOGÍA DE EVALUACIÓN?

Antes de estudiar el testeo o evaluación de software es necesario definir un plan investigativo que permita cumplir con los

objetivos, ayudando a definir y establecer los criterios que se encargarán de facilitar el proceso tanto de investigación como de ejecución. [2] Atendiendo a estas consideraciones la metodología para evaluar la calidad de software en la Universidad de Cundinamarca es la encargada de apoyar el proceso de creación de un software competitivo a partir de estándares internacionales para desarrollo de aplicaciones web elaboradas en la Universidad de Cundinamarca y así mejorar notablemente el proceso de diseño, desarrollo e implementación de software.

La metodología para la calidad de software está compuesta de dos partes; la primera enfocada a la documentación y modelación de los proyectos y la segunda dirigida a las pruebas de testeo entre estas las funcionales, de carga y estrés las cuales están dadas por herramientas libres. Como se había mencionado anteriormente está basada en la norma ISO/IEC/IEEE 29119 en esta se define vocabulario, procesos, documentación, técnicas y un modelo para realizar las pruebas de software que se puede utilizar dentro de cualquier ciclo de vida de desarrollo. [3]

De esta manera la presente metodología permite posicionar competitivamente a la Universidad de Cundinamarca no sólo a nivel latinoamericano sino a nivel mundial, para cumplirlo fué necesario basarse en el framework UML 2.0 [5] y los criterios a tener en cuenta para el éxito en el modelado del proyecto, así como los diferentes componentes de cada uno de los diagramas como se muestra a continuación:

Clasificación para modelado de aplicaciones	
Diagrama	Componentes
Actividades	Actividad
	Acción
	Restricciones de acción
	Flujo de control

	Nodo inicial
	Nodo final de actividad
	Nodo final de flujo
	Flujo de objetos
	Nodos de decisión y combinación
	Nodos de bifurcación y unión
	Región de expansión
	Gestores de excepción
	Región de actividad interrumpible
	Partición
Casos de uso	Actores
	Casos de uso
	Inclusión de casos de uso
	Casos de uso extendidos
	Puntos de extensión
	Límite del sistema
Secuencia	Línea de vida
	Mensajes
	Ocurrencia de ejecución
	Mensajes self
	Mensajes perdidos y encontrados
	Inicio y final de línea de vida
	Restricciones de tiempo y duración
	Fragmentos combinados
	Puerto
	Descomposición en parte
	Continuaciones
	Invariantes de Estado
	Objeto o rol
Colaboración	Enlace
	Mensaje
	Anidamiento
	Iteración
	Bifurcación
Clases	Clases
	Notación de clase
	Interfaces
	Tablas
	Asociaciones
	Generalizaciones
	Agregaciones
	Clase asociación
	Dependencias

científicos

Despliegue	Trazado
	Relaciones
	Anidamientos
	Nodo
	Instancia de nodo
	Estereotipo de nodo
	Artefacto
	Asociación
	Nodo como contenedor

Tabla1. Clasificación modelado UML. Autores: Erika Alejandra Nieto Guevara / Ingrid Ximena Acosta Mesa.

Estos diagramas a su vez se dividen según la relación que existe entre ellos así: [6]

- Comportamiento: Exhibe las reacciones de un sistema. Los diagramas de comportamiento son: Diagrama de actividades y diagrama de casos de uso.
- Interacción: Son un conjunto de diagramas de comportamiento que permiten enfatizar las interacciones entre objetos. Los diagramas de interacción son: diagrama de secuencia y diagrama de colaboración.
- Estructura: Muestran elementos que actúan de forma independiente al tiempo. Los diagramas de estructura son: diagrama de clases y diagrama de despliegue.

Así mismo para garantizar el proceso de evaluación se cuenta con 25 pruebas diferentes las cuales se dividen a su vez en 3 subcapas las cuales son: Gestión de pruebas, pruebas funcionales y pruebas de carga y rendimiento. Para la definición de estas pruebas se tuvo en cuenta la norma ISO/IEC/IEEE 29119 parte 1 en la que se definen 41 pruebas de software y al analizar la evaluación que se lleva a cabo en la Universidad de Cundinamarca se definieron los criterios necesarios para establecer las pruebas, documentación de pruebas y técnicas de pruebas, las cuales garantizarán la calidad del producto software después de verificar su funcionalidad, lo cual facilitará el análisis y consecución de los objetivos previamente establecidos. Para realizar la clasificación de las pruebas se tuvo en cuenta la prioridad de para realizarlas: [7]

Clasificación de pruebas de evaluación de calidad		
Prueba	prioridad	Tipo de prueba
Prueba de informe de preparación de datos	Alta	Gestión de pruebas
Prueba de proceso de diseño e implementación	Alta	Gestión de pruebas
Prueba de entorno	Alta	Gestión de pruebas
Prueba de gestión	Alta	Gestión de pruebas
Prueba exploratoria	Alta	Gestión de pruebas
Prueba de compatibilidad	Media	Gestión de pruebas
Prueba de escenario	Media	Gestión de pruebas
Prueba de condición	Media	Gestión de pruebas
Prueba de proceso de planificación	Media	Gestión de pruebas
Prueba de requisito	Media	Gestión de pruebas
Prueba de especificaciones	Media	Gestión de pruebas
Prueba organizacional	Baja	Gestión de pruebas
Prueba de instalabilidad	Baja	Gestión de pruebas
Prueba de artículos	Baja	Gestión de pruebas
Prueba basada en riesgo	Alta	Funcional
Prueba puesta en marcha de procesos	Alta	Funcional
Prueba de ejecución	Alta	Funcional
Prueba de verificación y validación	Alta	Funcional
Prueba de regresión	Media	Funcional
Prueba estructural	Media	Funcional
Prueba de resultado	Media	Funcional
Prueba de procedimiento	Baja	Funcional
Prueba de resistencia	Alta	carga y rendimiento
Prueba de carga	Alta	carga y rendimiento
Prueba de estrés	Alta	carga y rendimiento
Prueba de rendimiento	Media	carga y rendimiento

Tabla 2. Clasificación de pruebas para evaluar calidad. Autores: Erika Alejandra Nieto Guevara / Ingrid Ximena Acosta Mesa.

A continuación una breve descripción de cada una de las capas de pruebas:

- Gestión de pruebas: Tiene en cuenta los requerimientos, casos de prueba, la ejecución de las pruebas y los ciclos de estas así como los resultados de las pruebas.
- Pruebas funcionales: Se basan en la ejecución e incluyen la implementación de la prueba y la notificación de incidentes.[4]
- Pruebas de carga y rendimiento: Permiten comprobar el comportamiento del software bajo condiciones variables de carga y de esta manera verificar el desempeño durante la evaluación.

ESCENARIOS DE PRUEBA

Para facilitar el proceso de evaluación de calidad se cuenta con el uso de herramientas libres las cuales sistematizan este proceso, estas herramientas son:

- QaBook: Esta herramienta permite hacer la gestión de pruebas a través de la creación de entornos, componentes, requerimientos, casos de prueba, determinar las pruebas de funcionamiento que se van a ejecutar, almacenar los defectos encontrados durante la evaluación y finalmente realiza reportes en los que es posible ver gráficamente los informes en Word, mostrando gráficas de los requerimientos del proyecto de acuerdo a su estado, prioridad (alta, media, baja), tipo de requerimiento (funcional - no funcional); así mismo es posible observar los casos de prueba y los defectos que se presentan durante la evaluación determinando de esta manera su prioridad, gravedad o estado del defecto encontrado.
- Selenium IDE: Esta herramienta está diseñada para Firefox cuenta con una interfaz sencilla para manejar y ejecutar las pruebas funcionales, a través de la creación de casos de prueba en los cuales se puede

establecer un proceso de ejecución, este se puede reanudar, pausar o ejecutar paso a paso, es posible grabar los pasos para realizar una ejecución repetitiva.

- LoadUI: Esta herramienta permite medir la capacidad máxima que puede soportar un sitio web. Lo que permite conocer si el software está preparado para soportar una exigencia máxima que se puede presentar cuando la cantidad de usuarios se presenta en masa. Esta herramienta permite almacenar las distintas pruebas realizadas, y observar las estadísticas de las pruebas en tiempo real. Está orientada a las pruebas de carga y rendimiento.

III. RESULTADOS

Para garantizar el uso de la metodología se realizó la evaluación al proyecto "Modulo web para la gestión de usuarios del sistema de información de autoevaluación de la Universidad de Cundinamarca", el cual cuenta con módulos del sistema de información, estos son: fuentes primarias, fuentes secundarias y plan de mejoramiento.

En el caso del módulo administrador se verificó que se registraran los usuarios, sedes, facultades, programas que tiene la Universidad y la posibilidad de crear nuevos módulos.

Una de las deficiencias encontradas en este proyecto es que cuenta solamente con 3 diagramas de los que son tenidos en cuenta para la evaluación: diagrama de colaboración, diagrama de secuencia y diagrama de actividad, en este caso no se cuenta con diagramas de interacción. Para determinar la ponderación de la evaluación de calidad se tuvo en cuenta la participación del comité de evaluación para asignar porcentajes tanto al modelado como a la aplicación de pruebas.

En el momento de realizar las pruebas funcionales en los resultados se pudo identificar un error por retardo en el tiempo de respuesta desde que se registran los datos con respecto al inicio de sesión. Como esta herramienta (Selenium IDE) tiene la opción de grabar los

científicos

pasos que se han ejecutado y repetirlos, durante múltiples ejecuciones sólo presentó ese error, en términos generales la aplicación funciona correctamente, ocurrió un caso similar al ejecutar la aplicación varios usuarios a la vez el tiempo de respuesta es un poco lento.

Como seguimiento de esta actividad, la gestión de pruebas ayuda a determinar la prioridad (atención especial, urgente, alta, media o baja) de los requerimientos a través de la herramienta QABook y posteriormente esta ayudará a especificar si fueron aprobados los requerimientos asociados a los diferentes casos de prueba. El proyecto "Modulo web para la gestión de usuarios del sistema de información de autoevaluación de la Universidad de Cundinamarca" cuenta con varios requerimientos de prioridad media, cuando se presentó un defecto fue posible establecer si se podía continuar con la prueba o si la evaluación finalizaría en ese punto y es posible visualizar los resultados en el reporte final.

Aunque no es posible automatizar la evaluación del lenguaje de modelado universal (UML) debido a que no existe una herramienta que realice dicho proceso, para lograr una evaluación adecuada se tendrán los diferentes componentes y sus usos especificados en la Tabla1. Clasificación modelado UML presentada en la página 2 del presente artículo.

Finalmente el uso de herramientas libres para la evaluación de calidad de software permite garantizar las especificaciones necesarias con las cuales se debe contar para realizar un testeo de software adecuado en el momento de elaborar un proyecto informático, logrando perfeccionar la etapa de desarrollo y puesta en marcha.

IV. CONCLUSIONES Y TRABAJOS FUTUROS

Considerando las necesidades encontradas acerca de la evaluación de productos software desarrollados en la Universidad de

Cundinamarca y falta de estándares que sustenten y acompañen el proceso de creación del mismo desde el diseño hasta el desarrollo y ejecución de la aplicación. Al establecer y aplicar herramientas libres que permitan consolidar una metodología, es posible concertar la capacidad competitiva y la calidad de software desarrollado en la Universidad de Cundinamarca, así mismo teniendo en cuenta la labor realizada por el comité de evaluación logrando mejorar y ofreciendo ventajas a dicho proceso, con la posibilidad de migrar estas herramientas a entornos empresariales y de esta manera crear una cultura organizacional al reemplazar este proceso que se viene realizando de manera "artesanal" respecto a la evaluación de calidad de un producto software en un entorno cotidiano en el que se presentan ciertas exigencias que muchas veces en las prácticas educativas no son tenidas en cuenta.

Para realizar una nueva versión del presente proyecto se pretende construir una plataforma que integre las tres herramientas libres utilizadas para automatizar el proceso de evaluación y sus diferentes fases de ejecución.

Posteriormente se pretende generar un modelo matemático que sustente la precisión de un proyecto en determinado tiempo y según los criterios de evaluación que se han planteado en la presente metodología.

Un trabajo futuro a realizar para la metodología de evaluación de calidad de software en la Universidad de Cundinamarca consta de la seguridad informática que debería cumplir cualquier producto software para así garantizar el manejo adecuado de la información siguiendo los lineamientos normativos existentes para esta investigación, además de expandir esta evaluación a aplicaciones móviles y así lograr mayor auge en el desarrollo de productos software en cualquier tipo de plataforma.

V. REFERENCIAS

[1] Colombia líder en la región en la producción

de software de calidad. (Marzo, 2015).

Recuperado de:

<http://www.mintic.gov.co/portal/604/w3-article-8571.html>

[2] Calero C, Moraga M, Piattini M, (2010). *Calidad del producto y proceso software*. Edit. RA-MA.

[3] Carvallo, J. P. (2015, Mayo). *CALIDAD DE COMPONENTES SOFTWARE*. Recuperado de: <http://www.essi.upc.edu/~franch/papers/libro-calidad-cap-10-jpc-xf-cq-10-version-preliminar.pdf>

[4] González L, (2009). *Método para generar casos de prueba funcional en el desarrollo de software*. Revista ingenierías Universidad de Medellín. Vol. 8 N° 15.

[5] Kevitt, M. (2008). *Best Software Test & Quality Assurance Practices in the project life-cycle*. Dublín: Dublín City University.

[6] Kimmel P, (2010). *Manual de UML*. Edit. Mac Graw Hill.

[7] *Normas ISO/IEC/IEEE 29119 parte 1: Definiciones y vocabulario, parte2: Proceso de pruebas, parte 3: Documentación de pruebas* (2013).

[8] Riquelme J, (2006). *TESTEO DE SOFTWARE CON DOS TÉCNICAS METAHEURÍSTICAS*. Barcelona: CIMNE.

[9] Rueda G, Black R, (2011). *Fundamentos de pruebas de software (edición español)*. Edit. Rocky Nook.