

Revista
Española de
Innovación,
Calidad e
Ingeniería del Software



Volumen 6, Número 3 (especial XI JICS), noviembre, 2010

Web de la editorial: www.ati.es

Web de la revista: www.ati.es/reicis

E-mail: calidadsoft@ati.es

ISSN: 1885-4486

Copyright © ATI, 2010

Ninguna parte de esta publicación puede ser reproducida, almacenada, o transmitida por ningún medio (incluyendo medios electrónicos, mecánicos, fotocopias, grabaciones o cualquier otra) para su uso o difusión públicos sin permiso previo escrito de la editorial. Uso privado autorizado sin restricciones.

Publicado por la Asociación de Técnicos de Informática (ATI), Via Laietana, 46, 08003 Barcelona.

Secretaría de dirección: ATI Madrid, C/Padilla 66, 3º dcha., 28006 Madrid



Editor

Dr. D. Luís Fernández Sanz (director)

Departamento de Ciencias de la Computación, Universidad de Alcalá

Miembros del Consejo Científico

Dr. Dña. Idoia Alarcón

Depto. de Informática
Universidad Autónoma de Madrid

Dr. D. José Antonio Calvo-Manzano

Depto. de Leng y Sist. Inf. e Ing. Software
Universidad Politécnica de Madrid

Dra. Tanja Vos

Depto. de Sist. Informáticos y Computación
Universidad Politécnica de Valencia

Dña. M^a del Pilar Romay

CEU Madrid

Dr. D. Alvaro Rocha

Universidade Fernando Pessoa
Porto

Dr. D. Oscar Pastor

Depto. de Sist. Informáticos y Computación
Universidad Politécnica de Valencia

Dra. Dña. María Moreno

Depto. de Informática
Universidad de Salamanca

Dra. D. Javier Aroba

Depto de Ing. El. de Sist. Inf. y Automática
Universidad de Huelva

D. Guillermo Montoya

DEISER S.L.
Madrid

Dr. D. Pablo Javier Tuya

Depto. de Informática
Universidad de Oviedo

Dra. Dña. Antonia Mas

Depto. de Informática
Universitat de les Illes Balears

D. Jacques Lecomte

Meta 4, S.A.
Francia

Dra. Raquel Lacuesta

Depto. de Informática e Ing. de Sistemas
Universidad de Zaragoza

Dra. María José Escalona

Depto. de Lenguajes y Sist. Informáticos
Universidad de Sevilla

Dr. D. Ricardo Vargas

Universidad del Valle de México
México

Contenidos

REICIS

Editorial	4
<i>Luís Fernández-Sanz</i>	
Presentación	5
<i>Luis Fernández-Sanz</i>	
Taxonomía de factores críticos para el despliegue de procesos software	6
<i>Sussy Bayona, Jose Calvo-Manzano, Gonzalo Cuevas, Tomás San Feliu</i>	
Sistema de Gestión Integrado según las normas ISO 9001, ISO/IEC 20000 e ISO/IEC 27001	25
<i>Antoni Lluís Mesquida, Antònia Mas, Esperança Amengual, Ignacio Cabestrero</i>	
Implantación de CMMi nivel de madurez 2 en una PYME	35
<i>Fernando Ramos, Olimpia Torres, Nicolás Sánchez, Manuel Alba</i>	
Pruebas de Aceptación en Sistemas Navegables	47
<i>José Ponce, Francisco José Domínguez-Mayo, M. José Escalona, Manuel Mejías, Diego Pérez, Gustavo Aragón, Isabel Ramos</i>	
Análisis de métricas básicas y herramientas de código libre para medir la mantenibilidad	56
<i>Emanuel Irrazábal, Javier Garzás</i>	
Reduciendo distancia en proyectos de Desarrollo de Software Global Ágiles con técnicas de Ingeniería de Requisitos	66
<i>Mariano Minoli, Valeria de Castro, Javier Garzás</i>	
CMMI después de la certificación	76
<i>Vanesa Cabral y Juanjo Cukier</i>	
Comparando UML y OWL en la representación del conocimiento: correspondencia sintáctica	84
<i>Susana M. Ramírez, Yisel Alonso, Violena Hernández, Arturo Cesar Arias y Dayana La Rosa</i>	

Pruebas de aceptación en sistemas navegables

J. Ponce, F.J. Domínguez-Mayo, M.J. Escalona, M. Mejías,
D. Pérez, G. Aragón, I. Ramos

Departamento de Lenguajes y Sistemas Informáticos, Universidad de Sevilla
josepg@us.es

Resumen

En los últimos años la importancia de la fase de pruebas ha ido creciendo hasta el punto de que se establece su necesidad como una clave para el aseguramiento de la calidad de los productos software. A pesar de este interés y necesidad, sin embargo, la fase de pruebas es compleja y aún hoy no existen protocolos generales o normas ampliamente aceptadas que certifiquen qué se entiende por una buena fase de pruebas o no. Si a esto lo acompañamos de los nuevos requerimientos que las nuevas tecnologías y desarrollos incluyen, como por ejemplo los sistemas navegables, nos encontramos con que existe poco soporte para los equipos de desarrollo a la hora de ofertar una estrategia de pruebas efectiva. Este trabajo trata de presentar un análisis del estado del arte de un conjunto de pruebas muy específicas: las pruebas de aceptación. Y, concretamente, se centra en estudiar qué ofertas hay para el tratamiento de este tipo de pruebas en sistemas navegables. El artículo pretende ser una visión general que permite establecer puntos de trabajos iniciales para la definición de nuevas estrategias de pruebas en este campo.

Palabras clave: D.2 Software Engineering, D.2.5.k Testing strategies, D.2.5.l Test design, D.2.19 SoftwareQuality/SQA

Acceptance Testing for Navigation Systems

Abstract

In recent years the importance of the testing phase has become a key factor to ensuring the quality of software products. However, despite this interest and need, the testing phase is a very complex phase and, even today, there are no general protocols and widely accepted standards to certify what is meant by a good phase of testing or not. If new requirements that include new technologies and developments, such as navigation systems, must be also considered we find that there is little support for development teams when they look for an effective test strategy. This paper aims to present an analysis of the state of the art of a very specific set of tests: acceptance tests. And, specifically, it is focused on studying recent solutions to deal with acceptance tests in navigable systems. The article is a general view of the state of the art and the base to the definition of new test strategies in this field.

Key words: D.2 Software Engineering, D.2.5.k Testing strategies, D.2.5.l Test design, D.2.19 SoftwareQuality/SQA

Ponce, J. Domínguez-Mayo, F.J., Escalona, M.J., Mejías, M., Pérez, D., Aragón, G. y Ramos, I., "Pruebas de aceptación en sistemas navegables", REICIS, vol. 6, no.3, 2010, pp.47-55. Recibido: 8-11-2010; revisado: 14-11-2010; aceptado: 22-11-2010

1. Introducción

Dentro de las fases que conforman el ciclo de vida, la fase de pruebas es fundamental a la hora de asegurar la calidad del producto entregado orientada, entre otras muchas cosas, a la validación y verificación del sistema. Por verificación se entiende que se esté construyendo el producto correctamente y por validación que se esté construyendo el producto correcto. En dicha fase se intenta que el software sea entregado con la mayor calidad posible, entendiendo como producto de calidad aquel que cumpla con las características marcadas por algunos estándares entre los que destaca ISO/IEC 9126 [1] siendo éste un estándar internacional para la evaluación de la calidad del producto software. Por otro lado, SQuaRE (Software Product Quality Requirements and Evaluation) es una revisión de ISO/IEC 9126 y tiene las mismas características de calidad del software pero extendidas con nuevas subcaracterísticas para las características ya definidas en ISO 9126. El objetivo general de la creación del estándar ISO/IEC 25000 SQuaRE es organizar, enriquecer y unificar las series que cubren dos procesos principales: especificación de requerimientos de calidad del software y evaluación de la calidad del software, soportada por el proceso de medición de calidad del software. La serie ISO/IEC 25000:2005 reemplaza a dos estándares relacionados: ISO/IEC 9126 e ISO/IEC 14598. Por tanto, el objetivo de la fase de pruebas es que el producto final acabe cumpliendo las expectativas creadas por el mismo, ciñéndose de esta manera al estándar establecido.

En cuanto a estándares de pruebas se refiere, existen algunos como BS 7925 o IEEE Std 1008 [2] los cuales están primordialmente enfocados a pruebas unitarias que son llevadas a cabo en fases tempranas dentro de la fase pruebas. Sin embargo, existen áreas no cubiertas por los estándares anteriores como son aspectos organizativos, proceso y gestión de las pruebas, técnicas funcionales y no funcionales, etc. El objetivo del nuevo estándar ISO/IEC 29119 [3] es, por tanto, el de unificar los anteriores estándares en uno solo para intentar cubrir la mayoría de áreas no cubiertas por los mismos hasta el momento.

La idea general que se tiene para validar aplicaciones o sistemas informáticos son realizar un conjunto de pruebas de diferente índole y tipo, desarrolladas por un conjunto de personas pertenecientes a unas y otras entidades, ejecutadas por personas que de nuevo pueden pertenecer o no, a esas mismas o diferentes entidades, que por uno u otro lado componen toda esa maraña que rodea a el complejo proceso de creación de software, donde

una de las fases más importantes del ciclo de vida por donde va pasando dicho software, es el estudio, diseño, ejecución y validación de estas pruebas que van a dar una visión o valoración, muy importantes al software que finalmente se está construyendo. Dentro de este trabajo, tras una introducción a la fase de pruebas, nos centraremos primeramente dentro de toda la tipología de pruebas que tenemos en las de aceptación del sistema, pruebas que se encargan de realizar una muestra de individuos pertenecientes al colectivo de usuarios finales, bajo la supervisión de miembros de otros grupos de trabajo pertenecientes al proyecto. Abordaremos dichas pruebas, para finalmente centrarlas, en la aplicación de pruebas para validar sistemas navegables, poniendo como ejemplo de este tipo de sistemas cualquier sistema Web, haciendo un estudio de las características más importantes a cubrir intentando reflejar su repercusión en la mejora de calidad.

El hecho de centrarnos en las pruebas de aceptación, surge de intentar reforzar la visión de que el usuario integrado en dicha fase, de forma temprana, ayudaría a mejorar dicho proceso desde la fase de planificación y diseño de las pruebas, con las consiguientes mejoras de muchos aspectos cuantificables y deseables como pueden ser:

- Aumento en la calidad del software integrado
- Minimización de costes
- Aumento de la fiabilidad en los resultados del proyecto
- Se produce un incremento de la satisfacción del cliente al utilizar un software con una cantidad de errores inferior.
- Se incrementa la eficiencia del proceso de desarrollo.

En el apartado 2, se presenta de forma general la situación actual en las Pruebas de Aceptación así como las diferentes modalidades de pruebas. Seguidamente, en el apartado 3 se describen las diferentes Pruebas de Aceptación pero orientadas a Sistemas Navegables. Finalmente, en el apartado 4 se sintetizan una serie de conclusiones a tener en cuenta en base a las diferentes características deseables en sistemas Web.

2. Estudio de la situación actual en Pruebas de Aceptación

Dentro del tipo de pruebas que tenemos para validar el software, una de las más importantes son las de aceptación (user's tests). Son aquellas pruebas que son diseñadas por el propio equipo de desarrollo en base a los requisitos funcionales especificados en la fase

de análisis para cubrir todo ese espectro, y ejecutadas por el propio usuario final, no por todos evidentemente, pero sí por una cantidad de usuarios finales significativo que den validez y conformidad al producto que se les está entregado en base a lo que se acordó inicialmente.

Dependiendo de la complejidad del sistema a probar, si está o no dividido por módulos, etc. la realización de dichas pruebas se ejecuta de forma diferente. Si estuviera una aplicación dividida en módulos, se tratarían esto como subsistemas y estudiando si tienen o no la suficiente complejidad como para tratarlos de forma diferente, habría que realizar sesiones de prueba de aceptación diferentes. Las pruebas de aceptación, tienen dos tipos de procedimiento para realizarlas, las llamadas pruebas alfa y beta.

2.1. Pruebas alfa

Es aquella en la que se le entrega a un usuario final todo el producto final, junto a su documentación correspondiente para que este, en presencia del desarrollador y en entornos previamente preparados para el proceso de dichas pruebas, vaya informando de todo lo que vea que no está bien, que no se cumple, etc.

2.2. Pruebas beta

Es la que se le proporciona a usuarios finales, situados en lugares concretos de los puestos de trabajo donde finalmente será el software implantado, para que sea de nuevo el usuario y sin estar nadie presente del resto de grupos de trabajo, el que de nuevo vuelva a emitir unos informes de resultados e impresiones de la aplicación o sistema software desarrollado.

Independiente del proceso seguido, también tendríamos que prestar especial atención a la documentación entregada al cliente para su posterior evaluación, ya que la entrega de aplicación y documentación, debería tener la misma calidad.

3. Pruebas de Aceptación en Sistemas Navegables

La prueba de una determinada aplicación Web, ejemplo tomado dentro de los múltiples tipos de sistemas navegables debido en parte por la gran demanda de los mismos que hay, así como el enfoque que se han encauzado en los últimos años hacia la aplicación software basado en la Web, es una colección de actividades para descubrir errores en el contenido, la funcionalidad, utilidad, navegabilidad, rendimiento, seguridad de dicha aplicación. Dentro

de las pruebas de aceptación se intentará cubrir revisiones en una o varias dimensiones de calidad mostradas a continuación (basándonos en [4][5]):

- El contenido, evaluándose tanto a nivel sintáctico hasta el nivel semántico. En el primero se harán revisiones de vocabulario, puntuación, gramática, etc. y en el segundo se valora la corrección de la información presentada, la consistencia, así como la falta de ambigüedad en lo aportado.
- La función se probará para descubrir errores que indiquen la falta de conformidad por parte del cliente. Cada función de la aplicación se valorará en su corrección, inestabilidad así como la conformidad con diferentes estándares de programación.
- La estructura se valorará que garantice adecuadamente el contenido y la función de la aplicación, que es extensible y que puede soportarse conforme se agregue nuevo contenido o funcionalidad.
- La usabilidad, para asegurarse que la aplicación es capaz de soportar a cada tipo de usuario que se pueda conectar a ella pudiendo aprender y aplicar toda la sintaxis y semántica de la navegación requerida.
- La navegabilidad se probará para que toda la estructura de navegación esté correcta y descubrir errores tan simples como usuales de enlaces muertos, inadecuados o erróneos.
- El rendimiento, la compatibilidad, seguridad son otros aspectos a tener en cuenta en dichas revisiones para que la satisfacción del cliente sea máxima y la confianza del producto la más alta posible.

Cuando finalizan las pruebas de integración es el momento de comenzar dichas pruebas. La idea es elaborar un plan de pruebas que agrupe la mayor parte posible de las dimensiones de calidad anteriormente expuestas y que siempre busquen demostrar la conformidad del cliente con respecto a los requerimientos inicialmente definidos basándose en la medida de lo posible, en los estándares de pruebas comentados arriba

Para cada una de las pruebas que mencionamos a continuación, se realizarían técnicas tanto de caja blanca, aquellas que fueran más unitarias, así como de caja negra con vistas a probar de mejor forma la funcionalidad del sistema.

3.1. Pruebas de contenido

En relación a las pruebas de contenido que pueden ser tan triviales como errores tipográficos menores o tan significativos como información incorrecta, organización inadecuada o violación de leyes de la propiedad intelectual, siendo la misión de las pruebas de contenido intentar descubrirlos y muchos más antes que el usuario final los encuentre. Dichas pruebas tienen por objetivo, descubrir errores sintácticos, errores gramaticales, etc. en documentos de texto, representaciones gráficas y otros medios, descubrir errores semánticos, errores en la precisión o completitud de la información y encontrar errores en la organización o estructura del contenido presentado al usuario.

3.2. Pruebas de interfaz de usuario

La verificación y validación de una interfaz de usuario de una aplicación Web ocurre dentro de tres puntos distintos dentro del proceso de software, al nivel de análisis, al de diseño y al de pruebas. Durante la prueba, la atención es centrada en la ejecución de aspectos específicos de la aplicación de la interacción con el usuario, conforme salgan por la sintaxis y la semántica. Una de las funciones será dar validez vínculos, formularios, HTML dinámico, contenido de streaming, etc.

Del mismo modo, el usuario implícitamente irá realizando pruebas para medir la usabilidad del software, donde se indicará el grado en el que el usuario podrá interactuar efectivamente con la aplicación, es decir, en cuánto facilita la vida la aplicación al usuario. Características como la interactividad, legibilidad, estética, accesibilidad son aquellas que son lo más deseable y valorables dentro de las categorizaciones de usabilidad.

3.3. Pruebas de navegación

Este conjunto de pruebas tiene dos vertientes ya que por un lado, puede ser predecible los objetivos del usuario que llega a un determinado punto, y por otro resultar impredecible ya que el usuario puede elegir otro camino al previsto inicialmente, con lo cual, la misión de estas pruebas van a derivar en garantizar que son funcionales todos los mecanismos que permiten al usuario recorrer la aplicación.

Para ello, podrían ser probado muchos de los elementos que existen en la aplicación, como vínculos de navegación, anclas de páginas que nos redireccionan a algún lugar dentro de la misma, redirecciones hacia enlaces caídos, mapas de sitios, motores de búsqueda internos, etc.

3.4. Pruebas de componente

Las pruebas a nivel de componente se hacen al nivel de realizar las llamadas pruebas de función donde generalmente son las respuestas de nuestro sistema ante la entrada de datos en un determinado formulario, donde tras introducirlos, seleccionamos pulsando algún botón el inicio de la ejecución del proceso. Buscan encontrar errores dentro de estas funcionalidades. Aquí son usuales métodos de pruebas de caja negra tal y como particiones de equivalencia, análisis de valores límite, etc.

3.5. Pruebas de configuración

Dentro de la variabilidad de configuraciones posibles dentro de aplicaciones Web, tales como hardware, sistemas operativos, navegadores, capacidad de almacenamiento, velocidades de comunicación de red y otros muchos dentro del lado cliente, el error dependiendo de la configuración de un determinado cliente a otro pueden ser desde muy sutiles a muy significativos. Normalmente el usuario valida algunas configuraciones del lado cliente para dar conformidad a lo que está probando.

3.6. Pruebas de rendimiento

Cuando un usuario se pone delante de una aplicación Web y al cargar la página el tiempo es frustrante, el usuario está probando el rendimiento de la aplicación ante las diferentes peticiones. El usuario validará en cuestión de tiempos si son razonables o no los diferentes accesos a las diferentes secciones de la aplicación, si determinadas consultas a bases de datos en caso de existir son lo suficientemente rápidas, si no habría que optimizarlas. Para ello un concepto importante es el de carga, que es el nivel de peticiones que recibe el servidor concurrentemente y valorar el tiempo de respuesta para cada uno de ellos.

3.7. Pruebas de seguridad

El usuario tendrá que validar que el sistema cumple con todas las restricciones de seguridad exigidas en los requerimientos del mismo. Para ello comprobará los errores con los diferentes firewalls instalados, errores en las autenticaciones, encriptado así como en las autorizaciones. Todas ellas con el fin de descubrir vulnerabilidades o huecos en la seguridad de la aplicación.

4. Conclusiones y trabajo futuro

De todos los estudios revisados hasta el momento en la bibliografía existente, aparece latente la necesidad de la realización del tipo de pruebas anteriormente definidos, con la suficiente calidad para poder ajustarse a todas y cada una de las especificaciones que nos marcan los estándares anteriormente mencionados así como otros específicos como W3C [6], WAI[7], etc. como podemos observar en proyectos reales [9]. Como trabajos futuros, podríamos por un lado, plantear la relación de cada una de las pruebas del tipo anteriormente citadas, en qué atributos de calidad termina redundando de forma positiva para que la calidad final del producto mejore y por otro, la de realizar un profundo estudio de herramientas para el tipo de pruebas de aceptación como Canoo WebTest, JBehave, Concordion, Fitnessse, etc.

Agradecimientos

Esta investigación ha sido apoyada por el proyecto QSimTest (TIN2007-67843-C06_03) y por el proyecto Tempros (TIN2010-20057-C03-02) del Ministerio de Educación y Ciencia, España.

Referencias

- [1] ISO/IEC 9126-1: Software engineering - Product quality - Part 1: Quality model. First edition, 2001.
- [2] http://standards.ieee.org/reading/ieee/std_public/description/sc/1008-1987_desc.html
- [3] Tuya, J., “Hacia el nuevo estándar de pruebas ISO 29119”. *XI Jornadas de Innovación y Calidad del Software (JICS)*, Alcalá de Henares, 2009.
- [4] Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., and Fernandez, L., “Análisis y diseño de aplicaciones informáticas de gestión, una perspectiva de ingeniería del software”, Alfaomega, pp. 419-469, 2004.
- [5] *Ingeniería del software: Un enfoque práctico. Séptima Edición.* Roger S. Pressman, McGrawHill. Capítulos 17, 18 y 19.
- [6] <http://www.w3c.es/>
- [7] <http://www.w3.org/WAI/>
- [8] Méndez, E., Pérez, M., Mendoza, L.: *Aplicación de un Método para Especificar Casos de Prueba de Software en la Administración Pública. PRIS 2007: Taller sobre Pruebas*

en Ingeniería del Software. Actas de Talleres de Ingeniería del Software y Bases de Datos (JISBD 2007), Zaragoza, España, 2007.

- [9] Ponce, J., Escalona, M.J., Gómez, A., Luque, M. y Molina, A. “Definición de una política de pruebas en la gestión cultural: aplicación al desarrollo del proyecto Mosaico”, REICIS Vol. 6, No. 2, 2010