

Revista
Española de
Innovación,
Calidad e
Ingeniería del Software



Volumen 5, Número 2 (especial XI JICS), septiembre, 2009

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, 2009

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



Revista Española de Innovación, Calidad e Ingeniería del Software (REICIS)

Editores

Dr. D. Luís Fernández Sanz (director)

Departamento de Sistemas Informáticos, Universidad Europea de Madrid

Dr. D. Juan José Cuadrado-Gallego

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

Fundación Giner de los Ríos
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

Dr. D. José Ramón Hilera

Depto. de Ciencias de la Computación
Universidad de Alcalá

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, Juan J. Cuadrado-Gallego</i>	
Presentación	5
<i>Luis Fernández-Sanz</i>	
Analizando el apoyo de marcos SPI a las características de calidad del producto ISO 25010	6
<i>César Pardo, Francisco J. Pino, Félix García, Mario Piattini</i>	
Generación automática de casos de prueba para Líneas de Producto de Software	17
<i>Beatriz Pérez-Lamancha, Macario Polo</i>	
Análisis de la calidad y productividad en el desarrollo de un proyecto software en una microempresa con TSPi	28
<i>Edgar Caballero, José Antonio Calvo-Manzano, Gonzalo Cuevas, Tomás San Feliu</i>	
Asegurar que el software crítico se construye fiable y seguro	38
<i>Patricia Rodríguez</i>	
Visión Innovadora de la Calidad del Producto Software	49
<i>Antonio Calero, Paco Castro, Hugo Mora, Miguel Ángel Vicedo, David García</i>	
El análisis de anomalías detectadas en las pruebas de software: una vía para mejorar el ciclo de vida	56
<i>Ramón Enrique González</i>	
Experiencias de una PYME en la mejora de procesos de pruebas	63
<i>Antonio de Rojas, Tanja E.J. Vos, Beatriz Marín</i>	
Procedimiento para pruebas de intrusión en aplicaciones Web	70
<i>Delmys Pozo, Mairelis Quintero, Violena Hernández, Lisney Gil, Maria Felix Lorenzo</i>	
La madurez de los servicios TI	77
<i>Antoni Lluís Mesquida, Antònia Mas, Esperança Amengual</i>	
Una aplicación de la norma ISO/IEC 15504 para la evaluación por niveles de madurez de Pymes y pequeños equipos de desarrollo	88
<i>Javier Garzás, Carlos Manuel Fernández, Mario Piattini</i>	

Visión innovadora de la calidad del producto software

Antonio Calero, Paco Castro, Hugo Mora, Miguel Ángel Vicedo, David García
Company for Software and Development (CSD)
{acalero, fcastro, hmora, mavicedo, dgarcia}@csd.com.es

Resumen

El desarrollo y la ingeniería del software se han venido complicando día a día en aras de conseguir desarrollos y soluciones cada vez más completas, robustas y que respondan a los cada vez más exigentes requisitos de las organizaciones. El proceso de aseguramiento de calidad tiene como misión principal garantizar todos los requisitos de calidad establecidos, no sólo los que se refieren a un aspecto funcional, sino también a la fiabilidad, eficiencia, portabilidad, mantenimiento,... Para ello, los controles de calidad no deben aplicarse únicamente al código generado, sino que van mucho más allá, y deben recorrer elementos como las librerías de terceros, la arquitectura, la infraestructura tecnológica (servidor de aplicaciones, base de datos, integración), sin olvidar la seguridad, los procesos de construcción y la documentación. Para cada uno de estos elementos definiremos un conjunto de actividades a realizar, una serie de herramientas de soporte y un conjunto de resultados a obtener.

Palabras clave: calidad de software, aseguramiento de calidad, modelo de calidad, buenas prácticas, control de calidad.

Innovative vision of software product quality

Abstract

The development and engineering of software is becoming more complicated by the day, as we strive to achieve increasingly comprehensive and robust solutions which respond to companies' exacting requirements. The primary objective of the quality assurance process is to guarantee compliance with all of the established quality requirements, not only those which are related to functionality but also those that apply to reliability, efficiency, portability and maintainability. To attain this goal quality controls should not only be applied to source code but also must go far beyond, to encompass areas such as third-party libraries, architecture, infrastructure (application server, database server, integration environments), security, build, packaging and documentation. For each of these areas we define a set of quality assurance activities to be executed, a set of support tools and a specification of the results that should be obtained.

Key words: software quality, quality assurance, quality model, best practises, quality control.

Calero, A., Castro, F., Mora, H., Vicedo, M.A. y García, D., "Visión innovadora de la calidad del producto software", REICIS, vol. 5, no.2, 2009, pp.49-55. Recibido: 22-6-2009; revisado: 6-7-2009; aceptado: 31-7-2009

1. Introducción

El software es cada día mucho más complejo, a todos los niveles. A nivel funcional, cada vez le pedimos que haga más cosas, y más importantes, y a nivel técnico, la evolución de las tecnologías implican la adaptación de nuestro software en todo momento. Pero no sólo se trata de complejidad sino también de criticidad. El software es cada vez más crítico, automatizando aspectos más importantes del negocio, y eso tiene como consecuencia que cualquier error o problema que ocasione el software más impacto tendrá sobre las organizaciones. La tendencia actual en las organizaciones es la centralización del software y el empleo cada vez mayor de aplicaciones con arquitectura web, esta característica que permite usar las aplicaciones en cualquier momento y lugar obliga casi siempre a contemplar aspectos como la alta disponibilidad, tolerancia a fallos y el rendimiento. Esto requiere un control absoluto de todos los procesos implicados y además a todos los niveles, ya que cualquier interacción con el sistema podría desestabilizarlo. Lo habitual es que el proveedor de software garantice la calidad de su desarrollo, sin embargo esto no es suficiente, ya que hay que garantizar el correcto funcionamiento de todas las soluciones implantadas (infraestructura, comunicaciones, puestos, etc.).

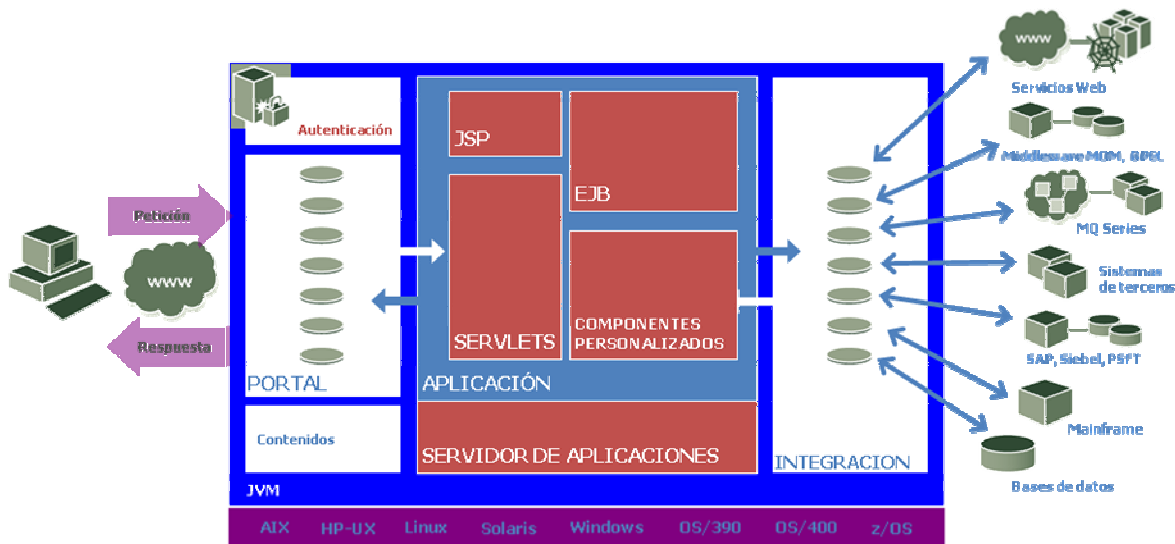


Figura 1. Elementos que participan en la puesta en marcha de los sistemas de información actuales.

2. Puntos de fallo potenciales

Entre los elementos clave de un sistema de información [1] consideramos como puntos de fallo potenciales los siguientes elementos:

- Balanceador de carga de las peticiones entre los servidores
- Servidores web de la capa frontal
- Servidores de autenticación y validación.
- Servidores de aplicaciones de la capa de negocio
- Servidores de base de datos
- Entornos de red y comunicaciones
- Servidores de backup y recuperación
- Cualquier otro sistema de información externo con el que exista una dependencia

A continuación reflejamos algunos ejemplos de puntos de fallo críticos.

2.1. Servidores Web

Los servidores web son un elemento fundamental en una arquitectura centralizada ya que son el punto de entrada a nuestro sistema. En prácticamente la totalidad de los casos el síntoma de la aparición de un problema es la pérdida de rendimiento. Las causas pueden venir por: consumo excesivo de recursos (CPU, memoria), servidores de aplicaciones o sistemas externos no disponibles, acceso a ficheros estáticos muy grandes y pesados, sistemas de ficheros que alcanzan el límite de tamaño, peticiones repetidas o “ecos” (cuando el usuario pulsa varias veces el mismo botón porque no responde el sistema).

La solución en la mayoría de organizaciones pasa por la ampliación de los recursos hardware, sin embargo, aunque parezca la solución sólo lo será puntualmente, ya que la clave es optimizar el rendimiento de las aplicaciones [2] y aplicar los parches correspondientes en el software de base.

2.2. Servidores de aplicaciones

Los servidores de aplicaciones son el núcleo de ejecución fundamental de nuestros sistemas de información por lo que cualquier problema que surja afectará de manera significativa a todos los componentes de nuestro sistema. Al igual que con el resto de componentes el síntoma principal es la pérdida de rendimiento, y viene ocasionado por: problemas de memoria de las aplicaciones y liberación de recursos, mala gestión de los drivers de acceso a base de datos, de la sesión de usuario y del pool de conexiones, acceso a disco elevado, tratamiento erróneo de cadenas de caracteres, transacciones largas e innecesarias.

Una vez más la solución habitual es ampliar los recursos hardware. Esto mejorará el rendimiento puntualmente pero tarde o temprano los problemas volverán a aparecer. De nuevo la clave es optimizar los sistemas y mantener el software de base actualizados.

2.3. Servidores de base de datos

Las bases de datos son otro elemento fundamental en nuestro sistema de información, y también suelen experimentar la mayor parte de nuestros problemas. Algunas de las posibles causas de problemas son: consultas pesadas, interbloqueos, descontrol en el número de sesiones abiertas, acceso a disco, falta de índices, elevado número de cursores abiertos.

De nuevo el síntoma experimentado en el sistema es la pérdida de rendimiento, y de nuevo la solución que se suele tomar es la ampliación de recursos hardware. En su lugar, la experiencia nos dice que en la mayoría de veces basta con optimizar los desarrollos y mantener al día el software de base de datos.

3. Visión y Modelo de Calidad de Software

Llegados a este punto nos damos cuenta de que problemas supuestamente resueltos no lo están, y que tenemos arquitecturas muy complejas en las que detectar el problema es muy difícil, y se hace necesario y fundamental establecer una monitorización continua [3] y la ejecución de actividades de aseguramiento de la calidad.

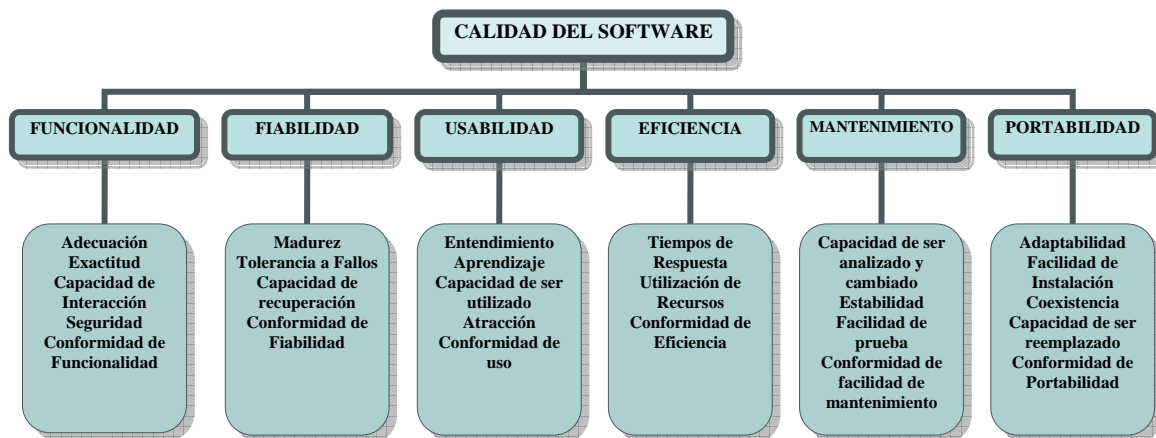


Figura 2. Propiedades de la calidad del software según ISO 9126

El término calidad se define como “propiedad o conjunto de propiedades inherentes a una cosa, que permiten juzgar su valor”. Por tanto, la calidad del software estará definida por el conjunto de propiedades del software que nos permitirán conocer si un software es

mejor o peor que otro, y concretamente estas propiedades vienen definidas por la norma ISO 9126 (estándar internacional para la evaluación de la calidad del software [4]).

El aseguramiento de la calidad del software conformará el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto cumple con los requisitos de calidad establecidos. El proceso de aseguramiento de calidad debe garantizar no sólo los aspectos que se refieren a cuestiones funcionales, sino también los que se refieren al resto de propiedades definidas (eficiencia, usabilidad, fiabilidad, portabilidad, etc.).

El modelo de calidad presentado establece que los elementos a revisar van desde el núcleo con el código fuente hasta la documentación pasando por la infraestructura de ejecución o los sistemas dependientes externos.

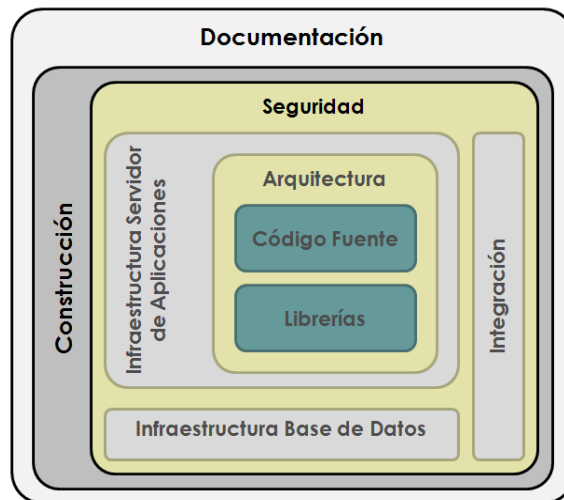


Figura 3. Visión conceptual del modelo de calidad de software

A continuación resumimos algunas de las revisiones que forman parte de nuestro modelo de aseguramiento de la calidad:

- Código fuente. Revisiones de convenciones de código, de pruebas unitarias y cobertura, trazas, control de versiones, complejidad, etc.
- Librerías. Revisiones de licencias y versiones de librerías utilizadas, dependencias y solapamientos, uso de estándares, etc.
- Arquitectura. Revisiones del modelo de arquitectura, consistencia, escalabilidad, mantenibilidad y rendimiento, etc.

- Infraestructura. Revisión de cachés y dependencias, revisiones de configuración en los servidores de aplicaciones y servidores web, parametrización, ficheros de trazas, etc.
- Base de datos. Uso de estándares, revisión de índices, particionamiento, históricos, procesos en segundo plano, controles de versiones del modelo, etc.
- Seguridad. Seguridad en los puestos, seguridad en las integraciones y comunicaciones, planes de contingencia y continuidad, etc.
- Construcción. Revisión de los modelos de construcción de los desarrollos (integraciones, migraciones, etc.), automatización de reglas, empaquetado y ejecución, etc.
- Documentación. Creación de sitios web para los desarrollos, automatización y revisión de la documentación, etc.

Un aspecto muy importante a tener en cuenta en todos los procesos de desarrollo de los productos es la integración continua. La integración continua tiene como finalidad prevenir errores muy comunes, adelantando todo aquello que pueda ser automatizado en el proceso de construcción de un producto. El objetivo es que con una frecuencia determinada se lleven a cabo esas tareas y en caso de detectar errores se proceda a su resolución cuanto antes, evitando que los errores aparezcan el día de publicación de una nueva versión. La integración continua se basa en un motor de reglas que permite automatizar la ejecución de todas esas tareas comunes y contribuye enormemente a mejorar la calidad del software.

3. Conclusiones

El aseguramiento de la calidad del software es una actividad continua durante todo el ciclo de vida de un sistema de información. El ámbito es global, desde el código fuente hasta la infraestructura, seguridad y documentación, por lo que requiere un equipo técnico multidisciplinar y herramientas de soporte específicas.

No sólo hemos de garantizar que el código fuente es de calidad, sino también como este código enlaza con las librerías de terceros, con la base de datos, con los servicios de integración, etc. etc. Sólo así podremos tener la confianza en que todos los requisitos de calidad se están cumpliendo.

Referencias

- [1] Casanova, J.C., Calero, A., Devesa, J., “Arquitectura de Sistemas y Aplicaciones en Entornos Críticos: La experiencia de la Conselleria de Sanidad”, *IX Congreso Nacional de Informática de la Salud, INFORSALUD 2006, Madrid.*
- [2] Calero, A., Gómez, J., “Gestión del Rendimiento y Calidad en Aplicaciones J2EE”, *X Congreso Nacional de Informática de la Salud, INFORSALUD 2007, Madrid.*
- [3] Calero, A., Casanova, J.C., Devesa, J., “Entornos de Producción Críticos en los Sistemas de Información Sanitarios. Monitorización. Diagnóstico”, *IX Congreso Nacional de Informática de la Salud, INFORSALUD 2006, Madrid.*
- [4] ISO (International Organization for Standardization), Software Product Evaluation. Quality Characteristics and Guidelines for their Use. ISO/IEC Std 9126, ISO, 2001.