

*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](http://www.ati.es)

Web de la revista: [www.ati.es/reicis](http://www.ati.es/reicis)

E-mail: [calidadsoft@ati.es](mailto: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<sup>a</sup> 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

<b>Editorial</b>	<b>4</b>
<i>Luís Fernández-Sanz, Juan J. Cuadrado-Gallego</i>	
<b>Presentación</b>	<b>5</b>
<i>Luis Fernández-Sanz</i>	
<b>Analizando el apoyo de marcos SPI a las características de calidad del producto ISO 25010</b>	<b>6</b>
<i>César Pardo, Francisco J. Pino, Félix García, Mario Piattini</i>	
<b>Generación automática de casos de prueba para Líneas de Producto de Software</b>	<b>17</b>
<i>Beatriz Pérez-Lamancha, Macario Polo</i>	
<b>Análisis de la calidad y productividad en el desarrollo de un proyecto software en una microempresa con TSPi</b>	<b>28</b>
<i>Edgar Caballero, José Antonio Calvo-Manzano, Gonzalo Cuevas, Tomás San Feliu</i>	
<b>Asegurar que el software crítico se construye fiable y seguro</b>	<b>38</b>
<i>Patricia Rodríguez</i>	
<b>Visión Innovadora de la Calidad del Producto Software</b>	<b>49</b>
<i>Antonio Calero, Paco Castro, Hugo Mora, Miguel Ángel Vicedo, David García</i>	
<b>El análisis de anomalías detectadas en las pruebas de software: una vía para mejorar el ciclo de vida</b>	<b>56</b>
<i>Ramón Enrique González</i>	
<b>Experiencias de una PYME en la mejora de procesos de pruebas</b>	<b>63</b>
<i>Antonio de Rojas, Tanja E.J. Vos, Beatriz Marín</i>	
<b>Procedimiento para pruebas de intrusión en aplicaciones Web</b>	<b>70</b>
<i>Delmys Pozo, Mairelis Quintero, Violena Hernández, Lisney Gil, Maria Felix Lorenzo</i>	
<b>La madurez de los servicios TI</b>	<b>77</b>
<i>Antoni Lluís Mesquida, Antònia Mas, Esperança Amengual</i>	
<b>Una aplicación de la norma ISO/IEC 15504 para la evaluación por niveles de madurez de Pymes y pequeños equipos de desarrollo</b>	<b>88</b>
<i>Javier Garzás, Carlos Manuel Fernández, Mario Piattini</i>	

# **Experiencias de una PYME en la mejora de procesos de pruebas**

Antonio de Rojas  
Clave Informática, S.L.  
Galileo Galilei, 12 - Elche Parque Industrial – España  
[aderojas@clavei.es](mailto:aderojas@clavei.es)

Tanja E.J. Vos, Beatriz Marín  
Centro de Investigación en Métodos de Producción de Software (ProS)  
Universidad Politécnica de Valencia - España  
{tvos, bmarin}@[pros.upv.es](mailto:pros.upv.es)

## **Resumen**

Las metodologías más habituales para la mejora de procesos de pruebas están orientadas a organizaciones grandes y son, por tanto, difícilmente aplicables al perfil de las PYME, con recursos limitados y poca madurez en estos procesos. En un trabajo anterior, se han propuesto una serie de acciones sencillas y concretas que las PYME pueden realizar para mejorar los procesos de testeo; sin dedicar muchos recursos, obteniendo resultados rápidamente y preparándolas para un posterior proceso de mejora más formal. En este trabajo, se presentan las experiencias que una PYME ha obtenido luego de implantar algunas de las acciones propuestas.

**Palabras clave:** Pruebas, Software, PYME, Mejora de Procesos.

## **SME experiences in testing process improvement**

### **Abstract**

Common methodologies for test process improvement are oriented towards big companies, and so they are more difficult to apply to SMEs that have limited resources and immature processes. In a previous work, a set of concrete and simple actions were proposed in order to SMEs can improve their test processes without dedicating too many resources. The underlying idea is that they will quickly obtain results that will prepare them for a more formal improvement process. These results will come from real needs of the specific companies and not from artificial needs listed in process improvement books. In this paper, experiences from an SME following recommendations of previous work will be presented,

**Key words:** Testing, Software, PYME, Process Improvement.

*De Rojas, A., Vos, T. y Marin, B., " Experiencias de una PYME en la mejora de procesos de pruebas", REICIS, vol. 5, no.2, 2009, pp.63-69. Recibido: 22-6-2009; revisado: 6-7-2009; aceptado: 31-7-2009*

## **1. Introducción**

Existen bastantes metodologías y técnicas para la mejora de procesos de pruebas, como por ejemplo Test Improvement Model (TIM) [2], Testing Organization Maturity Model

(TOM)<sup>2</sup>, Testing Maturity Model (TMM) [1], o Test Process Improvement (TPI) [3]. Estas metodologías han sido diseñadas para organizaciones más grandes, y no son fácilmente adaptables a estructuras mucho más pequeñas, como las PYME. El principal problema es que el coste y la duración de un proceso de mejora son desproporcionados respecto a los recursos disponibles en una PYME.

Este artículo describe los resultados en la mejora de los procesos de pruebas que ha obtenido la empresa Clave Informática, una PYME de la Comunidad Valenciana (España) que se dedica a diseñar y comercializar aplicaciones software de Producción, Gestión Comercial y Financiera, todas ellas integradas con Servicios ISP/IDC (Internet Service Provider / Internet Data Center) de alta disponibilidad y seguridad. Las acciones de mejora en Clave Informática se han basado en un trabajo anterior de uno de los autores de este artículo [4], donde se propone una serie de acciones sencillas y concretas que las PYME pueden realizar sin dedicar muchos recursos, obteniendo resultados rápidamente y preparándolas para un posterior proceso de mejora más formal.

El resto del artículo está organizado como sigue: la Sección 2 describe cómo preparar a una PYME para la mejora de procesos de pruebas, la Sección 3 presenta las experiencias de Clave Informática, y la Sección 4 presenta algunas conclusiones.

## **2. Preparando a las PYME para la Mejora de Procesos de Pruebas**

Antes de aplicar alguna metodología para la mejora de los procesos de pruebas en una PYME, es necesario primero aplicar acciones más sencillas para preparar a las PYME para mejorar estos procesos. En [4], se proponen 3 acciones muy prácticas y sencillas:

1. Formación y concienciación de la importancia de las pruebas del software.
2. Los programadores deben realizar algún tipo de pruebas unitarias sobre sus desarrollos, dejando a su elección qué y cómo hacerlo. Es necesario confiar en el conocimiento de los programadores: ellos saben lo que quieren programar, por lo tanto, que sean ellos los que comprueben si lo hacen correctamente.
3. Asignar (o contratar) al menos 1 persona dedicada exclusivamente a realizar pruebas de alto nivel, es decir, a realizar pruebas de aceptación e integración.

---

<sup>2</sup> Gerrald Consulting web site, <http://www.gerraldconsulting.com/default.asp?page=/tomoverview.html>

Para empezar, en [4] no se define un proceso estructurado de pruebas, simplemente se debe probar de la mejor forma posible, con el objetivo de encontrar errores y con la libertad para realizar las pruebas según se crea conveniente. Así, además de encontrar errores, se sacarán a la luz necesidades para realizar más eficazmente el trabajo (p. ej.: definición de requisitos, gestión de defectos, análisis de riesgos, planificación, etc.). Estas necesidades se transformarán en acciones que poco a poco mejoran los procesos de pruebas hasta llegar a procesos estructurados, sin depender de un modelo formal y la implantación de acciones artificiales que éste dicte.

### **3. Las Experiencias de Clave Informática**

La producción de software de Clave Informática gira alrededor de dos ERPs, que son aplicaciones cliente-servidor diseñadas en arquitectura de 3-capas. Los ERP han sido desarrollados con tecnología Microsoft y utilizan SQL Server como base de datos.

A partir de finales de 2003, Clave adoptó una serie de medidas en el proceso de desarrollo de software (vea la tabla 1). Estas medidas se tradujeron en instrucciones que dieron consistencia al proceso de desarrollo de software y entendimiento a todos los colaboradores. Luego, se establecieron los primeros indicadores para evaluar el proceso: Incidencias por Versión, Incidencias por Módulo, e Incidencias por Aplicación. De esta manera, Clave introdujo procesos de pruebas en la fase de desarrollo, pero a nivel de detalle todavía se carecía de procesos estructurados que permitiesen mejorar las pruebas del software y resolver dudas en cuanto a: (1) ¿Cómo empezar a probar las aplicaciones que ya tenían desarrolladas (con miles y miles de líneas de código)? y (2) ¿Cómo especificar la forma en que se debían probar los nuevos desarrollos? Además, el analista (o incluso el jefe de producto) ya no era la persona idónea para realizar las pruebas cuando el objetivo era disminuir el número de incidencias en las versiones finales.

Hacia finales de 2006, estos inconvenientes impulsaron a Clave Informática a establecer un “Comité de Calidad del Software”. Hablando con los autores del artículo [4], el comité decidió contar con personal exclusivo para la realización de las pruebas de alto nivel (aceptación e integración), y buscar ayuda con la formación y concienciación de la importancia de las pruebas del software. Gracias a convenios entre Universidad y Empresa, dos estudiantes de Ingeniería Técnica en Informática de la Universidad de Alicante

realizaron sus prácticas en Clave. Ellos tenían como cometido la puesta en marcha de procesos de pruebas bajo la tutela de la Universidad Politécnica de Valencia. Al final del período, Clave decidió contratar a tiempo completo a uno de los estudiantes.

<b>Descripción Medida</b>
<b>1.</b> Procedimientos que refuercen las pruebas unitarias por parte del propio programador mediante la creación de una plantilla de chequeos básicos por tipo de aplicación, que le sirvieran como puntos de verificación de cuestiones básicas a contemplar en el desarrollo. Por ejemplo, chequeos de la declaración de variables, chequeos de los colores de la interfaz, chequeos de nombres de procedimientos, etc.
<b>2.</b> Definición de un flujo de trabajo hacia el analista del sistema, que fue asignado como el encargado de realizar la verificación y validación de las modificaciones realizadas por un programador. La verificación se basaba en pruebas exploratorias a alto nivel con el objetivo de detectar errores en el desarrollo, los cuales generaban otro flujo de trabajo de vuelta al programador para su resolución, y así sucesivamente.
<b>3.</b> Desarrollo de módulos personalizados dentro de los ERPs, para incorporarles herramientas que facilitasen la gestión de todos los procesos. Se trataba de la implementación de un módulo de Gestión de proyectos y un módulo de Incidencias de Software para la captura y resolución de errores de programa.

Tabla 1. Medidas adoptadas por Clave Informática para conseguir la certificación ISO 9001

Las pruebas de aceptación resultaron un gran avance en Clave, puesto que están muy enfocadas tanto a la verificación y la validación de modificaciones y adaptaciones personalizadas del software a sus clientes. Consecuentemente, durante 2007 se prepararon procedimientos más estructurados de pruebas. Estos procedimientos fueron diseñados en base a las experiencias y necesidades que el testeador a tiempo completo había averiguado durante sus actividades de pruebas. Las pruebas de aceptación se integraron en del ciclo de desarrollo en dos fases: (1) Una vez que el programador finalizaba su desarrollo de manera de verificar naturalmente las modificaciones (así el flujo de trabajo iba del programador hacia el testeador), y (2) Luego de la puesta en marcha del sistema, se decide realizar las pruebas de aceptación al comienzo del proceso de desarrollo (antes de que el programador finalice su trabajo). Esto presenta muchas ventajas, como por ejemplo:

- Permite diseñar las pruebas en total colaboración con el cliente. Al mismo tiempo que se conocen los cambios pedidos, se va diseñando la forma de probarlos.
- Las pruebas no están viciadas por el desarrollo ya realizado por el programador, que puede influenciar la forma de realizar las pruebas.
- El programador puede verificar sus desarrollos siguiendo el diseño de las pruebas de aceptación ya realizado. Así, al testeador llega el trabajo mucho más depurado.

En último lugar, para que las pruebas de aceptación sean realmente efectivas, se ha detectado que es necesario poner en marcha un sistema de integración continua, que permita tener una versión “Beta” disponible todos los días, sobre la cual se realizasen las pruebas en las mismas condiciones que si se tratase de la versión final instalada en los clientes. Siempre existe la posibilidad de realizarlas sobre el código fuente, pero en Clave son conscientes de que el comportamiento puede variar. Finalmente, una vez que los programadores terminan su trabajo, deben actualizar el controlador de versiones. De esta manera, en Clave se ha ideado un proceso de integración propio, que se ejecuta por la noche y que permite la generación de la versión “Beta” para verificar y también la actualización de la base de datos dependiente de esos cambios. Así, el testeador tiene cada mañana una versión totalmente funcional para realizar las pruebas de aceptación.

Además, se introdujo una nueva métrica: el Índice de Errores no Detectados (EnD) para proporcionar una idea de la efectividad de las pruebas. La fórmula usada es:

$$\text{Índice Errores No Detectados} = \frac{\text{Errores Detectados por Cliente}}{\text{Errores Detectados por Cliente} + \text{Errores Detectados por el Proceso de Pruebas}}$$

Este indicador no sólo contribuye en la evaluación de la efectividad de las pruebas actuales, sino que también ayuda a establecer objetivos de mejora para el futuro. Los resultados de la aplicación de la métrica EnD fueron concluyentes: existe una gran diferencia entre las aplicaciones sometidas a los procesos de pruebas, de las que se habían dejado fuera deliberadamente porque su funcionamiento no era tan crítico. Así, las percepciones internas de la empresa y de los clientes se trasladaron a datos objetivos.

La gráfica en Figura 1, muestra la evolución de todo lo comentado hasta este punto, reflejando el número de incidencias detectadas en los procesos de pruebas desde 2005 hasta 2008. Estas incidencias no llegaron a manos de los clientes debido a que se solucionaron antes de las versiones finales. En esta grafica se comparan los resultados de dos aplicaciones: una sometida a los procesos de pruebas, y otra que se decidió dejar fuera. La aplicación probada fue desarrollada por 5 programadores a tiempo completo y tiene alrededor de 800.000 líneas de código, en cambio, la aplicación sin pruebas fue desarrollada por 3 programadores y tiene alrededor de 500.000 líneas de código. El esfuerzo real de pruebas requirió aproximadamente el 30% del tiempo de desarrollo. El 21% de las incidencias detectadas se clasificaron como graves, es decir: (1) Alta

repercusión económica en el cliente, debido a resultados erróneos o por una posible parada de negocio; (2) Alta cantidad de clientes afectados cuando la incidencia está contenida en un módulo de uso común. El tiempo de resolución para las incidencias graves es de un día, mientras que para las incidencias leves es de tres días. El 10% de las incidencias corresponde tienen un impacto mínimo en la funcionalidad del programa, o su resolución implica grandes cambios en la estructura de la aplicación. Estas incidencias se planifican para su tratamiento en el siguiente trimestre.

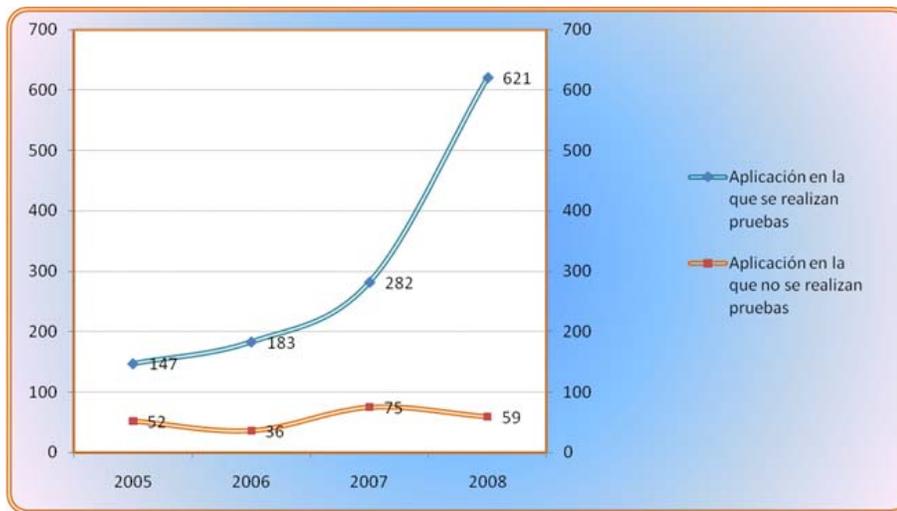


Figura 1: Numero de incidencias detectadas en dos aplicaciones

El análisis de la gráfica en la Figura 1 permite obtener las siguientes conclusiones:

- La introducción de procesos de pruebas mejora los resultados del desarrollo de software (Vea la diferencia de incidencias de ambas aplicaciones en 2005 y 2006).
- Con la incorporación de personal dedicado a las pruebas la mejora es considerable (En 2007 las incidencias son muy superiores a la tendencia de los años anteriores).

La mayor diferencia se produjo entre 2007 y 2008, donde se produjo una mejora exponencial debido a la implantación de las necesidades detectadas por el testeador para realizar eficazmente su trabajo. Por razones de espacio, no se han detallado estas necesidades. Sin embargo, éstas no serían útiles a otras PYME, ya que cada PYME tiene diferentes necesidades dependiendo de su personal, procesos y productos.

## **4. Conclusiones**

En este artículo, hemos descrito el éxito que Clave Informática ha tenido estructurando paulatinamente los procesos de pruebas basándose en las necesidades internas de la empresa detectadas por empezar a probar con el objetivo de encontrar errores. Hoy por hoy, Clave Informática sigue trabajando en la mejora continua, destacando lo siguiente:

- Automatización de Pruebas. Hay procesos que se están realizando de forma manual y repetitivamente, que al automatizarlos se ganaría en productividad y eficiencia.
- Pruebas unitarias. El objetivo es extender como una práctica habitual entre los programadores, la realización de pruebas unitarias para los nuevos desarrollos.

En cuanto al impacto sobre el personal de Clave, las mejoras aquí introducidas han contribuido a: Mejorar la seguridad y bienestar de los desarrolladores, ya que saben que sus resultados están verificados y validados por el testeador; Mejorar la confianza de los consultores y el personal de soporte, ya que saben que los procesos de pruebas están permitiendo liberar software con menos errores; y Mejorar la satisfacción de los clientes, ya que se les está dando lo que necesitan y libre de errores. Por ello, invitamos a otras PYME a leer el artículo [4] y lograr el éxito que ha tenido Clave.

## **Referencias**

- [1] Burnstein, I., Suwanassart, T., Carlson, C.R., “The Development of a Testing Maturity Model”, En: *Procs of the 9<sup>th</sup> Int Quality Week Conf, 21-24 de mayo de 1996*.
- [2] Ericson, T., Subotic, A., Ursing, S., “TIM-a test improvement model”, *Software Testing, Verification and Reliability*, vol. 7, n° 4, pp. 229-246, 1997.
- [3] Koomen, T.; Pol, M., *Test Process Improvement: A practical step-by-step guide to structured testing*. Addison-Wesley, 1999.
- [4] Vos, T.E.J., Sanchez, J., Mannise, M., “Mejorando el testeo en las PYME ¿Cómo empezar?”, En: *Proc de la 5a Ed de las JTS2008, 2-4 de abril de 2008, Valencia*.