

***Revista***  
***Española de***  
**Innovación,**  
**Calidad e**  
**Ingeniería del Software**

Volumen 1, No. 2, diciembre, 2005

**Web de la editorial: [www.ati.es](http://www.ati.es)**

**E-mail: [reicis@ati.es](mailto:reicis@ati.es)**

**ISSN: 1885-4486**

Copyright © ATI, 2005

Ninguna parte de esta publicación puede ser reproducida, almacenada, o transmitida por ningún medio (incluyendo medios electrónicos, mecánicos, fotocopias, grabaciones o cualquier otra) sin permiso previo escrito de la editorial.

Publicado por la Asociación de Técnicos en Informática

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

### **Editores**

**Dr. D. Luís Fernández Sanz**

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 Editorial**

**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

**Dña. Tanja Vos**

Instituto Tecnológico de Informática  
Universidad Politécnica de Valencia

**D. Raynald Korchia**

InQA.labs

**D. Rafael Fernández Calvo**

ATI

**Dr. D. Oscar Pastor**

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

**Dra. Dña. María Moreno**

Depto. de Informática  
Universidad de Salamanca

**Dr. D. Javier Aroba**

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

**D. Antonio Rodríguez**

Telelogic

**Dr. D. Javier Tuya**

Depto. de Informática  
Universidad de Oviedo

---

## 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</i>	
<b>La mejora de procesos de software en las pequeñas y medianas empresas. Un nuevo modelo y su aplicación a un caso real</b>	<b>7</b>
<i>Antonia Mas, Esperanza Amengual</i>	
<b>¿Cuál es la madurez que necesitarían los procesos para el desarrollo de sistemas de software crítico?</b>	<b>31</b>
<i>Patricia Rodríguez, Josefina Alonso, José C. Sánchez</i>	
<b>Un sondeo de la práctica actual de pruebas de software en España</b>	<b>43</b>
<i>Luis Fernández</i>	

---

## **Editorial**

The logo for REICIS, consisting of the word "REICIS" in a bold, white, serif font, centered within a solid black rectangular box.

---

REICIS comienza su andadura con dos propósitos principales: El primero, convertirse en una publicación académica de reconocido prestigio en el área de la Innovación, Calidad e Ingeniería del Software; y el segundo, e igual de importante, servir de medio de comunicación para la industria del software.

REICIS es editada por la Asociación de Técnicos en Informática (ATI), editora también de NOVATICA, la revista decana de la informática en España, con más de 30 años de andadura y un reconocido prestigio.

REICIS pretende seguir los pasos de NOVATICA y mantener un alto nivel académico y la máxima rigurosidad en sus publicaciones. Para esto contará con expertos prestigiosos para realizar la revisión de los trabajos publicados así como el continuo asesoramiento de los expertos de su consejo editorial.

REICIS invita desde estas líneas a todos los profesionales relacionados con el mundo de la Innovación, Calidad e Ingeniería del Software a que utilicen REICIS como el medio para dar a conocer sus trabajos e investigaciones teniendo las máximas garantías de la profesionalidad con que serán tratados sus trabajos. Podrán encontrar todas las instrucciones necesarias para el envío de sus contribuciones en la página web de la revista: [www.ati.es/reicis](http://www.ati.es/reicis).

Luis Fernández Sanz  
Juan J. Cuadrado-Gallego  
Editores

---

## **Presentación**

The logo for REICIS, consisting of the word "REICIS" in a bold, white, serif font, centered within a solid black rectangular box.

---

REICIS reúne en este número interesantes contribuciones sobre diversos aspectos de interés en el mundo de la ingeniería y calidad del software tanto desde el punto de vista de la investigación como de las aportaciones al mejor conocimiento de prácticas y técnicas disponibles y aplicables en la industria.

En el primer artículo, las profesoras Antonia Mas y Esperança Amengual de la Universitat de les Illes Balears nos presentan los resultados de la investigación sobre la aplicación de modelos de mejora de procesos y calidad de software sobre pequeñas y medianas empresas. El proyecto Quasar permitió a Pymes de Baleares abordar el camino de la mejora y a las autoras obtener valiosa información para personalizar y adaptar los modelos a este entorno. Los datos y experiencias recogidos en el artículo “La mejora de los procesos de software en las pequeñas y medianas empresas (pyme). Un nuevo modelo y su aplicación a un caso real” conforman la base de la tesis doctoral que la profesora Mas, por otra parte destacado miembro del Grupo de Calidad de Software de ATI, defendió en este año 2005 obteniendo la mejor calificación por su trabajo de definición de los modelos de mejora adaptados de los estándares ISO 15504 e ISO 9001:2000.

En “¿Cuál es la madurez que necesitarían los procesos para el desarrollo de sistemas de software crítico?”, los autores se plantean si los actuales procesos de certificación de sistemas software críticos son suficientes para evaluar todos los aspectos de seguridad y fiabilidad de este tipo de sistemas. Y presentan una propuesta de cómo y qué se debería evaluar para certificar sistemas software críticos. Este artículo corresponde a la ponencia mejor valorada en las pasadas IX Jornadas de Innovación y Calidad del Software de ATI (celebradas en julio de 2004 en la Universidad Europea de Madrid) en puntuación combinada de revisores y público asistente.

Por último el tercer artículo “Un sondeo sobre la práctica actual de pruebas de software en España” se presenta un estudio sobre el estado de las pruebas del software en España. La novedad y relevancia de este trabajo radica en el hecho de que aunque existen estudios similares a nivel europeo, éste es el primero que se presenta a nivel nacional español y se basa en los datos recogidos en cursos de formación y en eventos del Grupo de Calidad del Software de ATI.

Luis Fernández Sanz



# **La mejora de los procesos de software en las pequeñas y medianas empresas (pyme). Un nuevo modelo y su aplicación a un caso real**

Antònia Mas, Esperança Amengual

Departament de Ciències Matemàtiques i Informàtica, Universitat de les Illes Balears

{antonia.mas, eamengual}@uib.es

## **Abstract**

Interest in SMEs (*Small and Medium Enterprise*) sector has become widely known around the World. In consequence some works dealing with the application of software process improvement activities in this kind of companies have already been published. This article discusses the similarity between large organizations and small companies. A very exhaustive study about software process improvement and assessment models has been performed and it has been applied to the software SMEs sector in particular. The article presents a summary of adaptations made on the well-known models as well as on SMEs specially developed models. Moreover, the paper exposes a new SPICE (*Software Process Improvement and Capability dEtermination*) based model for a Quality Management System implementation on software SMEs and its application to some software SMEs in the Balearic Islands.

## **Resumen**

El interés por el sector de las pymes se ha generalizado en todo el mundo y se pueden encontrar ya algunos trabajos que relatan la aplicación de actividades relacionadas con la mejora de los procesos de software a este tipo de empresas. En este artículo se debate en primer lugar sobre la coincidencia de las características de las grandes organizaciones con las de las pequeñas empresas.

Se ha realizado un estudio muy exhaustivo sobre los modelos de evaluación y mejora de los procesos de software, se ha revisado cómo ha sido su aplicación en el caso particular de las pequeñas y medianas empresas de desarrollo de software, y se ofrece en el artículo un resumen, tanto de las adaptaciones que se han hecho de los modelos más conocidos, como de los que han sido desarrollados específicamente para este sector.

Además, se presenta un nuevo modelo para la implantación de un Sistema de Gestión de Calidad en Pymes de Desarrollo de Software basado en SPICE y se detalla su aplicación a un conjunto de pymes de las Islas Baleares.

**Palabras clave:** Mejora de procesos de software (SPI), CMM, SPICE (ISO/IEC 15504), Calidad de proceso software, Sistema de Gestión de la Calidad (SGC), ISO 9001:2000, Pyme, Certificación.

## **1. Las características de las pymes**

### **1.1. Estudios realizados en pymes**

Aunque hay un consenso generalizado en la literatura sobre la existencia de diferencias operacionales entre las grandes y las pequeñas empresas, no se han llevado a cabo iniciativas para verificarlo. En [6] se realiza un interesante estudio con el objetivo de determinar si el tamaño de una organización puede afectar a la estrategia de implantación de un programa de mejora de procesos y al grado de éxito que se alcance. Se selecciona una muestra muy amplia de empresas y de personal de las TIC (Tecnologías de la Información y de las Comunicaciones) de Noruega, se debate sobre el tipo y el tamaño de la muestra, para poder extraer resultados extrapolables estadísticamente. Se describen las variables, las medidas aplicadas, los cuestionarios de evaluación, los procedimientos de recolección de datos y las técnicas de análisis de datos utilizadas.

Se analizan los resultados y se concluye que hay diferencias fundamentales entre las grandes empresas (más de 200 desarrolladores) y las pymes (menos de 30 desarrolladores):

- Con respecto al rendimiento general, las grandes empresas obtuvieron niveles más altos de éxito en la implantación de programas de SPI que las pequeñas, algo que era de esperar, pero alcanzaron valores similares en cuanto al éxito empresarial.
- La diferencia fundamental se halló en las respectivas aproximaciones o visiones de los programas de SPI que se llevan a cabo en cada uno de los dos tipos de empresas. Mientras las grandes compañías despliegan sus buenas prácticas a través del establecimiento de modelos de procesos, procedimientos formales, guías, reglas, etc., las pequeñas obtienen sus mejores resultados derivados de la creatividad y de la dedicación de los recursos humanos implicados en el proyecto de mejora. Esto significa que para que las pequeñas



empresas puedan implantar programas de mejora de manera tan eficiente como las grandes, deben dedicar todo su esfuerzo en conseguir la participación de los empleados y en la exploración de nuevo conocimiento. En resumen, los procesos formales de las grandes organizaciones se suplen con la relación informal e interpersonal que se realiza en las pequeñas.

A pesar de estas diferencias, se ha podido demostrar cuantitativamente, a partir de todas las medidas de rendimiento realizadas durante el estudio, que las empresas de cualquier tamaño mejoran su rendimiento introduciendo programas de mejora de procesos. Es decir, en definitiva, el tamaño de una empresa no limita el éxito de los programas de mejora.

Según [27], el mayor error en la implantación de programas de SPI en pymes se da por la falta de seguimiento de los planes de acción y de los planes de implantación, debido fundamentalmente a que estas actividades son muy costosas de realizar en todos los sentidos, ya que consumen mucho tiempo y recursos. Como dice [28], un ciclo completo de SPI puede llevar entre 18 y 24 meses, demasiado tiempo para una pequeña empresa.

Otro problema añadido para las pymes y, que se menciona en [4], es que, aunque el retorno de la inversión de un programa de SPI está previsto que alcance entre 4 y 9 veces la inversión inicial, éste no se produce hasta un medio largo plazo, nuevamente demasiado para una pyme. Así pues, la dificultad de aplicación de los grandes modelos de SPI a las pymes es debido fundamentalmente a los costes asociados a su aplicación y al largo plazo de espera necesario para la observación de resultados.

Según [15] y [16], las pymes europeas, en general, no pueden permitirse grandes inversiones en sus procesos y lo que necesitan son aproximaciones que posibiliten un alineamiento de la mejora de sus procesos de software con sus objetivos de negocio.

Según estos autores, las pequeñas empresas:

- Son flexibles en su organización interna frente a la rigidez que suele acompañar a las grandes compañías.
- Tienen un estilo de gestión muy diferente de las grandes.
- No tienen departamentos especiales que les permitan realizar tareas complejas. Es decir, no disponen de especialistas en temas concretos.
- Gozan de recursos financieros limitados. No pueden permitirse el lujo de comprar o contratar todo lo que necesitan.

- Necesitan personal específico y especialmente formado para los programas de mejora de los procesos de software. También necesitan asesores, que son un personal muy costoso.

Según [26], para las pequeñas empresas el reto más difícil es ir modificando los procesos para ir adaptándose a las circunstancias cambiantes. Para que una pyme pueda gestionar un cambio de crecimiento inducido en sus procesos de desarrollo del producto, manteniendo al mismo tiempo la suficiente continuidad para que sigan siendo mínimamente predecibles y posibilitar la planificación, deben observarse ciertos aspectos:

- Un proceso es una herramienta más que un fin en sí mismo. Ningún proceso por sí solo puede transformar una organización indiferente en una organización efectiva.
- Los procesos deben ser simples. Los procesos complejos son difíciles de establecer, mantener y actualizar.
- Los procesos deben ser robustos, es decir, fáciles de aplicar y debe ser difícil equivocarse durante su aplicación.
- Los procesos deben adaptarse al entorno cambiante.

## **1.2. Características diferenciadoras de las pymes**

A partir de todos los estudios analizados, hemos podido observar que estas características diferenciadoras de las pequeñas organizaciones son debidas, entre otros factores a:

- **Los recursos humanos.**
  - Los equipos y el tamaño de los equipos. El número de empleados suele ser pequeño. Muchos de los empleados pertenecen simultáneamente a los diferentes grupos o equipos: de dirección, de gestión, de desarrollo, de SPI.
  - Los roles. No existe una especialización. La misma persona tiene que estar capacitada para llevar a cabo una gran variedad de actividades de desarrollo de software.
  - Las responsabilidades no suelen estar bien definidas.
  - Gran dependencia de los individuos [11]. Se refuerza la tarea individual y en muchos casos no hay una diferenciación clara de las funciones de cada

empleado. Los “héroes” tienen mucho protagonismo en este tipo de empresas.

- El esfuerzo estimado para la aplicación de grandes modelos de mejora es aproximadamente de una persona/mes dedicada al proyecto [9]. Este coste es difícilmente asumible en una pequeña empresa.
- Falta de formación de los empleados en los modelos.
- Desconocimiento por parte del líder de los beneficios que podría suponer la implantación de un programa de mejora. Falta de motivación.
- **Los aspectos económicos.**
  - En una pequeña empresa prima más la obtención de beneficios a corto que a largo plazo. Por otra parte, es muy importante que las medidas establecidas sean de aspectos tangibles y vitales para la empresa.
  - Los costes derivados de las auditorías externas, que se realizan normalmente con varios auditores, acostumbran a ser económicamente inviables para las pymes.
  - Una pyme dedica pocos recursos a programas de I+D. Concretamente las inversiones en SPI suelen ser pequeñas.
- **Los procesos.**
  - Es muy costoso para una pyme desarrollar y reflejar los resultados de la implantación de programas de mejora de procesos de software con el nivel de detalle y formalidad que acostumbran a exigir los grandes modelos.
  - Los resultados de las evaluaciones se expresan habitualmente en términos generales [28], esto significa que no pueden ser directamente traducibles a recomendaciones. Para las grandes empresas, puede ser posible realizar estas tareas, pero para las pequeñas, en muchas ocasiones, representa un esfuerzo imposible de asumir.
- **Los proyectos.**
  - Una pequeña empresa acostumbra a trabajar con proyectos no demasiado grandes y que no se extienden mucho en el tiempo. De la misma forma, suele también fijar sus objetivos. El trabajo, las previsiones, la situación a medio largo plazo, no es tan popular en las pymes como en las grandes compañías.

- El número de proyectos que suele manejar de manera simultánea una pequeña empresa acostumbra a ser pequeño. Si tiene problemas con alguno de ellos, representa problemas en un porcentaje muy alto del trabajo.

En resumen, se puede afirmar que casi todos los autores están de acuerdo en que las características especiales de las pequeñas empresas hacen que los programas de mejora de procesos deban aplicarse de un modo particular y visiblemente diferente a cómo se hace en las grandes organizaciones y que no es tan sencillo como considerarlas pequeñas versiones a escala de las grandes compañías [22]. El acuerdo ya no existe a la hora de afirmar si es mejor adaptar uno de los grandes modelos o es mejor utilizar uno diseñado específicamente para este sector.

Paulatinamente, las actividades de mejora de los procesos de software han ido penetrando en las empresas de tamaño mediano que, a efectos estadísticos, se incluyen bajo el nombre de pymes por el número de empleados que tienen, pero que difieren mucho de la estructura y funcionamiento de las empresas pequeñas o microempresas [5]. De las microempresas en general, podemos afirmar que no les ha llegado aún dicha inquietud por la mejora de sus procesos ya que, en muchos casos, incluso desconocen su existencia.

## **2. Modelos de mejora de procesos aplicados a pymes**

Los modelos más conocidos, CMM (*Capability Maturity Model*) y SPICE, han sido creados y tradicionalmente aplicados a las grandes compañías, con algunas iniciativas puntuales de adaptación de un cierto modelo a las pequeñas y medianas empresas. No se han encontrado apenas modelos basados en SPICE y especialmente adaptados a las necesidades de las pymes, pero sí diferentes iniciativas de aplicación del modelo SPICE en pequeñas empresas de desarrollo de software. Se han encontrado algunas adaptaciones realizadas a partir del modelo CMM, así como diversas aplicaciones efectuadas en pequeñas organizaciones.

Por otra parte, se han producido algunas aportaciones de métodos propios, algunos basados en un modelo ya existente y otros creados específicamente para el sector de las pymes.

En la tabla 1 se presenta un resumen de las aplicaciones realizadas de los diferentes modelos de mejora de procesos de software al caso específico de pymes.

Año	Organismo/Autor	Iniciativa/proyecto	Descripción
1993	<p><b>ESSI</b> (<i>European Software and System Initiative</i>)                      Desde que se inició la fase piloto en 1993 [8], la ESSI ha promovido más de 450 proyectos relacionados con SPI, todos ellos dirigidos a estimular el desarrollo, a experimentar y difundir conceptos, buenas prácticas, métodos, etc. que permitan la mejora de los procesos de software y que estimulen su aplicación en las empresas de desarrollo del software de toda Europa.</p>	<p>Marco <b>ESPRIT</b>                      (<i>European Commission's Research and Development Programme for Information Technologies</i>)</p> <p><b>Esprit Project 23873 - SPIRE</b> (<i>Software Process Improvement in Regions of Europe</i>) [24]</p>	<p>Ayudar a las pequeñas empresas y unidades de desarrollo de software con un máximo de 50 empleados, a:</p> <ul style="list-style-type: none"> <li>✓ Mejorar sus procesos de software.</li> <li>✓ Obtener beneficios tangibles a partir de las inversiones en programas de SPI.</li> <li>✓ Formar a sus empleados en todos los aspectos de SPI.</li> <li>✓ Compartir las experiencias del proyecto con otras empresas de las mismas características.</li> </ul>
1997	<p>Grünbacher, P. [9]</p>	<p><b>Un proceso de evaluación basado en CMM y adaptado a pymes</b></p>	<p>Grünbacher describe un proceso de evaluación basado en CMM y adaptado a pequeñas empresas de desarrollo de software, utilizando una herramienta de soporte a la evaluación.</p>
1999	<p><b>ESSI</b> (<i>European Software and System Initiative</i>)                      Con el fin de estimular, soportar y coordinar todo el esfuerzo llevado a cabo en estos experimentos, la Comisión Europea dentro del programa ESPRIT-ESSI estableció una red de 18 participantes-socios denominada ESSI PIE Nodes, denominándose también proyectos ESPINODE [7].</p>	<p><b>TOPS</b> (<i>Toward Organised Software Processes in SMEs</i>) [25]</p>	<p>Es el ESPINODE que promovió la iniciativa de mejora de procesos en pymes de desarrollo de software en el Centro de Italia. Se ofrecieron evaluaciones de los procesos gratuitas para todas las empresas que se adscribieron al proyecto. Esto permitió recopilar una gran cantidad de datos derivados de estas evaluaciones y elaborar un informe sobre la madurez de los procesos en las pymes italianas.</p>
1999	<p><b>NRC</b> (<i>Norwegian Research Council</i>)</p>	<p><b>SPIQ</b> (<i>Software Process Improvement for better Quality</i>)</p>	<p>En [18] se describe una experiencia de mejora de procesos llevada a cabo en una compañía noruega, dentro del programa SPIQ subvencionado por el NRC, con el objetivo principal de definir un marco genérico y sistemático para la mejora del proceso de software que conduzca a un incremento de competitividad de las empresas noruegas.</p>
1999	<p>Calvo-Manzano, J. [3]</p>	<p><b>MESOPYME</b>                      (<i>Método/Modelo de mEjora del proceso SOfware orientado a la Pequeña Y Mediana Empresa</i>)</p>	<p>MESOPYME proporciona un Método de mejora formado por 5 fases y un paquete de actuación, que es un conjunto de componentes organizativos, técnicos y de gestión que ayudan a dar una solución concreta a un problema software bien definido.</p>

Año	Organismo/Autor	Iniciativa/proyecto	Descripción
			Esta arquitectura proporciona una guía para soportar, analizar, diseñar y gestionar los procesos software, de forma que sean consistentes con CMM, ISO/IEC 15504 o el modelo de procesos de referencia elegido.
2000	SEI (Software Engineering Institute)	<b>IDEAL</b> (Initiating, Diagnosing, Establishing, Acting, Leveraging) [20]	En [14] se describe la investigación realizada para determinar la aplicabilidad del modelo a pequeñas empresas, así como las adaptaciones que han sido realizadas en cada una de las cinco fases del modelo IDEAL para posibilitar su uso en empresas danesas.
2000	Batista, J., Dias de Figueiredo, A. [1]	<b>Aplicación del modelo CMM a una pyme portuguesa</b>	Se describe la aplicación del modelo CMM a una empresa portuguesa con un equipo inferior a 10 personas y con unos recursos muy limitados. El principal objetivo del estudio era ver si el modelo podía ser aplicado, utilizado y adaptado para conseguir la mejora real de los procesos de software en equipos muy pequeños.
2000	Horvat, R.V., Rozman, I. and Gyorkos, J. [11]	<b>PROCESSUS</b>	Formado por un conjunto de Procedimientos Estándares y otro de Documentos Estándares junto con una metodología que nace de la integración del modelo CMM y de las Normas ISO 9001:1994 e ISO 9000-3:1997, todo ello orientado a la aplicación de la mejora de procesos de software en pequeñas empresas.
2001	Leung, H., Yuen, T. [17]	<b>Small process o Small Project lifecycle</b>	Se muestra el desarrollo de un modelo, denominado <i>Small process o Small Project lifecycle</i> , basado en ocho áreas clave de proceso de CMM y especialmente creado para la aplicación a pequeños proyectos. Un Proyecto pequeño es aquel cuyo esfuerzo es inferior o igual a 3 personas/mes, considerando asimismo que a mayor número de miembros del equipo, mayor incremento del esfuerzo en la comunicación.
2002	Richardson, Ita [23]	<b>SPM</b> (Software Process Matrix)	Es un modelo adaptado a pequeñas empresas de desarrollo de software denominado basado en QFD ( <i>Quality Function Deployment</i> ). Existen algunas variantes del modelo QFD, pero el <i>Tour-phase model</i> , adaptado

Año	Organismo/Autor	Iniciativa/proyecto	Descripción
			por el <i>American Standards Institute</i> y que contiene cuatro matrices, es el más utilizado.
2003	Beecham, S., Hall, T., Rainer, A. [2]	<b><i>Mejora del proceso de software. Problemas en doce compañías. Un análisis empírico.</i></b>	Se realiza el estudio de los problemas detectados durante la implantación de un programa de mejora de procesos según el modelo CMM en 12 empresas de desarrollo de software. En este trabajo se observa que hay una gran relación entre el nivel de madurez de la empresa y los tipos de problemas encontrados
2004	Guerreo, F., Eterovic, Y. [10]	<b><i>Los 10 factores de éxito para una pyme</i></b>	Se describe una iniciativa de aplicación del modelo CMM a una empresa chilena de desarrollo de software, basada en considerar los 10 factores relacionados con el entorno organizacional que pueden contribuir al éxito de un programa de SPI en una pyme.

Tabla 1. Resumen de la aplicación de modelos de SPI a pymes.

### 3. Modelo

En [19] se presenta un nuevo modelo para la implantación de un sistema de gestión de calidad que soporta todo el proceso de mejora y que está orientado específicamente a las pequeñas y medianas empresas de desarrollo de software. La aportación en este caso reside, en que se trata de una adaptación para pymes de la Norma internacional ISO/IEC 15504 [13], que lidera junto con el modelo CMM-SW [21] el sector de la mejora de procesos de software.

En el modelo se recogen todas las actividades relacionadas con la gestión de la calidad en una pyme de desarrollo de software, para que pueda utilizarse como marco de referencia en cualquier empresa interesada en la calidad desde el punto de vista del proceso.

En el modelo, se propone un método formado por un conjunto de actividades que, a su vez, se dividen en tareas. Para cada tarea, y siguiendo la estructura de la Metodología MÉTRICA V.3, se identifican los productos de entrada y los productos de salida. También se establecen las técnicas y prácticas útiles en el desarrollo de cada tarea así como los participantes en la misma.

El método propuesto divide las actividades de gestión de la calidad en dos grandes grupos:

### **3.1. Actividades de Implantación del SGC**

- Este primer grupo engloba las actividades relacionadas con el establecimiento del SGC una vez que la empresa ha decidido iniciarse en el camino de la mejora de sus procesos. En este primer grupo pueden diferenciarse tres bloques:
- El primero está compuesto por actividades orientadas a iniciar e implicar a toda la empresa hacia la mejora continua. Por una parte, dicho bloque está dedicado a establecer los parámetros generales del Sistema de Gestión de Calidad, desde la definición de la política, objetivos y plan de calidad de la empresa, hasta el establecimiento de los estándares de trabajo. Por la otra, se ha considerado que la motivación a todo el personal de la empresa y la formación específica para el grupo de calidad son actividades esenciales que deben realizarse en la primera fase de Implantación del SGC.
- En un segundo bloque, se incluyen un conjunto de actividades que concentran todas las tareas propias de la realización de una evaluación y mejora. Se realiza la identificación, el análisis y el estudio detallado de todos los procesos de la empresa, así como el cálculo de la capacidad de los mismos. Posteriormente, se mejoran los procesos seleccionados y una vez comunicada la mejora se procede a su implantación.
- En el tercer y último bloque, se incluyen las actividades de auditoría de la mejora. Se recogen todas las tareas necesarias para la certificación de la empresa según la Norma ISO 9001:2000 [12] y todas las relacionadas con la realización de una segunda evaluación de los procesos específicos del ciclo de vida del software según SPICE.

### **3.2. Actividades de Mantenimiento del SGC**

En el segundo grupo se recogen todas las actividades relacionadas con el mantenimiento del sistema implantado y con la mejora de los procesos. Se trata de actividades de seguimiento y control del plan de mejora continua establecido en la empresa.

## **4. Aplicación del modelo a un caso real: QuaSAR**

Dentro del ámbito del trabajo presentado en [19] se ha llevado a cabo el proyecto QuaSAR (*Qualitat de Software a les baleARs*), iniciativa que ha partido de la UIB (*Universitat de les Illes Balears*) y que ha permitido realizar un estudio de la situación actual del sector de las empresas de desarrollo de software en el entorno Balear, además



de ofrecer soporte en el camino hacia la mejora de sus procesos de software y hacia la certificación según la Norma ISO 9001:2000. El proyecto QuaSAR también ha permitido aplicar los resultados de esta investigación y así refinar el método genérico para la implantación de un SGC en pymes, a partir de los puntos débiles que se han detectado en su primera aplicación.

#### **4.1. Participantes de QuaSAR**

En una fase previa al inicio del proyecto QuaSAR, y como punto de partida del mismo, se iniciaron los contactos con el Departamento de calidad del IDI (*Institut d'Innovació Empresarial de les Illes Balears*), organismo público cuya misión es fomentar la calidad en las empresas y organizaciones de las Baleares, para que actuara como intermediario entre nuestro grupo de investigación de la UIB y las empresas de desarrollo de software que pudieran tener inquietudes de mejora de sus procesos.

Una parte importante del éxito que tuvo el proyecto QuaSAR es gracias a sus participantes, que provinieron de entidades bien diferentes y con motivaciones completamente distintas. Así pues, además de la UIB, que actuó como promotor del proyecto, los participantes fueron:

1. **El IDI, organismo público** que actuó como entidad coordinadora y de soporte logístico entre todas las partes implicadas. También se encargó de gestionar todos los aspectos económicos, incluyendo la petición de subvenciones a través de proyectos financiados por el *Govern de les Illes Balears*.
2. Una **asociación**, GRUPSoftBALEAR que agrupaba a todas estas empresas y que debería actuar como motor de difusión de las buenas prácticas en futuras actuaciones entre las otras empresas asociadas, pero que en esta ocasión no se habían sumado al proyecto.
3. **Pymes de desarrollo de software** que deseaban tanto obtener una certificación según la Norma ISO 9001 para que les ofreciera una ventaja competitiva, como una mejora de los procesos de desarrollo de software. En el inicio del proyecto QuaSAR no había ninguna empresa del sector que gozara de este reconocimiento en Baleares. Las 7 empresas privadas participantes en el proyecto son pequeñas compañías que tienen entre 5 y 30 empleados dedicados al desarrollo de software. Están principalmente orientadas al desarrollo de aplicaciones a medida para el sector servicios o

para el sector turístico, muy presente en la economía Balear, y algunas de ellas están especializadas en los sectores hotelero y de restauración. También participó en el proyecto una empresa pública.

4. **Una empresa externa de consultoría** experta en la implantación de ISO 9001:2000 en empresas de desarrollo de software.
5. **Becarios.** Alumnos de último curso de la Ingeniería Superior en Informática de la *Universitat de les Illes Balears*, dando soporte tanto a la implantación del sistema de gestión de calidad como participando en las mejoras de los procesos propios de software.
6. **Una entidad de certificación.**

#### **4.2. Bloque inicial de actividades**

El primer bloque está compuesto por actividades orientadas a iniciar e implicar a toda la empresa hacia la mejora continua. Comprende las actividades GC 1 *Establecimiento de los parámetros generales del Sistema de Gestión de Calidad* y GC 2 *Formación*.

Se establecieron unos estándares de trabajo comunes a todo el proyecto QuaSAR que serían adoptados por todas las empresas participantes en el mismo.

En cada empresa se constituyó el equipo de trabajo y se determinó el plan de calidad específico para la organización. Las empresas que no tenían fijados la política y los objetivos de calidad los establecieron. En cada una de ellas se determinaron los sistemas de información que serían objeto de la gestión de calidad.

Una vez realizado todo el proceso de selección, tanto de la empresa consultora como de los becarios, para cada una de las empresas adscritas al proyecto, se inició la actividad de formación que cubrió los aspectos siguientes:

- ✓ Motivación a todo el personal de la organización, que se realizó de manera conjunta para las 8 empresas participantes.
- ✓ Formación a los responsables de calidad y a los becarios en los procesos del ciclo de vida del software y en las bases del modelo SPICE. También se realizó de forma agrupada.

#### **4.3. Segundo bloque: evaluación y mejora de los procesos**

Este segundo bloque cubre las actividades orientadas a la evaluación y mejora de los procesos de la empresa, tanto en el ámbito de la gestión como en el de los procesos específicos. Comprende las actividades GC 3 *Realización de la primera evaluación SPICE*, GC 4 *Análisis y documentación de los procesos* y GC 5 *Mejora de los procesos*.

#### 4.3.1. Realización de la primera evaluación SPICE

Para conocer el estado actual de cada empresa y poder efectuar las evaluaciones que se utilizarían como línea de base para medir su progreso se realizó, por parte de la empresa consultora, una valoración de la gestión empresarial y, por parte de la Universidad, una evaluación de la capacidad de los procesos del ciclo de vida del software.

Las evaluaciones SPICE se realizaron según las tareas descritas en la actividad GC 3 del método propuesto en [19]. Una vez establecido el calendario de las evaluaciones, se mantuvieron sesiones de trabajo de unas 4 horas de duración con cada una de las empresas participantes con la finalidad de recoger los datos sobre el estado de todos sus procesos ya que la evaluación se realizó a nivel global y no para un subconjunto determinado de procesos.

Como resultado de esta primera evaluación se emitió un informe sobre la situación de todos los procesos contemplados por la Norma ISO/IEC 15504, con el valor y el nivel de capacidad alcanzados, para cada una de las empresas participantes en el proyecto.

En el informe se sugirieron propuestas de mejora para los indicadores de realización (Nivel 1 de capacidad) que habían obtenido un valor inferior al 50%. Se definieron claramente los objetivos de la mejora y se indicaron las actividades concretas a realizar para alcanzar dichos objetivos. También se proporcionó una planificación detallada con hitos bien definidos. Se necesitó en este punto una fuerte colaboración de la empresa, tanto para mostrar el acuerdo con las puntuaciones obtenidas en la evaluación, como para seleccionar y priorizar las mejoras, pues no es realista pensar que pueden abordarse todas ellas de manera simultánea.

#### 4.3.2. Presentación de resultados y priorización de procesos a mejorar

A partir de este informe de evaluación cada empresa seleccionó 3 procesos para su mejora. La coincidencia entre las 8 empresas participantes fue casi plena ya que, entre todas ellas, se seleccionaron los siguientes 5 procesos:

- **El proceso de pruebas.** Lo que la empresa denomina de manera genérica “pruebas del software” abarca todas las pruebas realizadas durante el desarrollo de un proyecto software, desde las pruebas unitarias hasta las pruebas del sistema una vez integrado con el hardware.
- **El proceso de gestión de la configuración.**
- **Los procesos de análisis y diseño.**

- **El proceso de gestión de proyectos.**
- **El proceso de medición, análisis y mejora.**

#### 4.3.3. Análisis y documentación de los procesos

Para implantar el sistema de gestión de calidad marcado por la Norma ISO 9001:2000 se definieron el Catálogo de procesos, la Matriz de procesos de la empresa y la Ficha de cada proceso. Al mismo tiempo que se iban inventariando los procesos reales que había establecidos en ese momento en la empresa, se iban introduciendo todas las mejoras que eran aplicables a corto plazo. Hay que recordar que el proceso que se describe es el que se somete a la revisión de los auditores de certificación. Una empresa no puede superar con éxito una auditoría si no tiene definidos sus procesos principales, si bien, la descripción de un proceso no debe mostrar intenciones sino realidades.

Debido a que la obtención de la certificación está ligada a la implantación de los procesos, la empresa empezó a trabajar en este sentido. La implantación en proyectos reales de la empresa resultó la más costosa del proyecto

#### 4.3.4. Mejora de los procesos seleccionados

Por razones temporales se creyó conveniente centrar la mejora únicamente en tres procesos y los 5 procesos citados anteriormente se sometieron a votación entre las 8 empresas participantes.

- El primer lugar, con 7 votos, lo obtuvo el proceso de pruebas.
- El segundo lugar, con 6 votos, fue ocupado por el proceso de medición, análisis y mejora.
- El tercer lugar, con 4 votos, fue para el de gestión de la configuración.

Se planificaron y llevaron a cabo tres sesiones de mejora de procesos. La Universidad fue la encargada de organizar cada una de estas jornadas. El objetivo principal era, además de transmitir a la empresa los conocimientos necesarios para posibilitar la mejora del proceso, poder ofrecer una visión práctica de su implantación: tratamiento real seguido en otras empresas, aplicaciones informáticas de soporte al proceso existentes en el mercado, etc.

Una vez establecido el contacto con profesionales especialistas en cada una de las áreas y de fijar el calendario, se procedió a la celebración de las sesiones en las que participó el personal que creyó conveniente cada empresa.

Así pues, en realidad, paralelamente a la implantación marcada por la Norma, se trabajó en la mejora de los procesos, algunos de ellos totalmente nuevos para las

empresas. Fueron especialmente útiles tanto las indicaciones sugeridas a partir de la evaluación SPICE, como las derivadas de las sesiones de mejora realizadas. Las empresas redefinieron los procesos y establecieron al mismo tiempo una planificación para su puesta en marcha.

#### **4.4. Tercer bloque: auditoría de la mejora**

Este último bloque cubre las actividades destinadas a comprobar que se ha realizado una mejora de los procesos de la empresa. Comprende las actividades GC 6 *Certificación según la Norma ISO 9001:2000* y GC 7 *Segunda evaluación SPICE*.

El IDI fue la entidad encargada de gestionar todo el proceso de selección y contratación de la certificadora. Se establecieron unos contactos iniciales con un conjunto de empresas certificadoras que realizaron una oferta económica para abaratar notablemente los costes del proceso de certificación de cada empresa, en el caso de que se contratara el servicio para las ocho empresas. A partir de todas las propuestas y de mutuo acuerdo entre todas las pymes participantes, se seleccionó una empresa para la realización de la certificación.

Todas las empresas adheridas al proyecto QuaSAR optaron por la realización de una **auditoría previa** a la auditoría de certificación. En esta visita previa habitualmente el auditor revisa la documentación del Sistema de Gestión de la Calidad, audita un proyecto de desarrollo de software y se reúne con la dirección de la empresa, tanto para presentar al futuro equipo auditor como para conocer y comprobar el grado de implicación de la dirección con el sistema, que es uno de los requisitos de la Norma. Con ello el auditor comprueba que la empresa está preparada para pasar la auditoría. En general, el resultado de estas visitas fue satisfactorio. El auditor sólo detectó no-conformidades menores y dio el visto bueno para la auditoría. Además, como la visita previa no se considera formalmente como una auditoría, el auditor pudo realizar actividad de consultoría y propuso soluciones a los problemas detectados.

##### **4.4.1. Auditoría de certificación según la Norma ISO 9001:2000**

En la auditoría de certificación se audita el cumplimiento de todos los puntos de la Norma. En las posteriores, las auditorías de mantenimiento, ya no es necesario revisarlos todos.

Hay una serie de puntos que se deben verificar en todas las auditorías de forma obligatoria, mientras que el resto de puntos únicamente se revisan si el auditor lo cree conveniente. La primera tarea a realizar en una auditoría es la planificación de la misma.

Se realiza una reunión con el responsable de calidad para acordar cuándo se auditará cada punto de la Norma y para saber cuál será el personal que se verá afectado. Durante la auditoría la empresa no debe detener su funcionamiento. El auditor tiene que adaptar su planificación a la disponibilidad del personal.

En promedio, la auditoría se realizó en tres días y fue realizada por un único auditor. La duración de una auditoría es directamente proporcional al número de empleados afectados por el SGC.

- El primer día, se realizó la reunión de planificación entre el auditor y el responsable de calidad. Ese mismo día, el auditor comprobó el punto 4 *Sistema de gestión de la calidad*, el punto 5 *Responsabilidad de la dirección* y los procesos de comercialización junto con el personal de calidad: el directivo responsable de calidad y los otros dos miembros del departamento de calidad.
- El segundo día, el auditor revisó cinco proyectos de la empresa. Los proyectos fueron seleccionados de forma que hubiera como mínimo un proyecto de cada departamento. Para cada proyecto, el auditor entrevistó al jefe del proyecto en su puesto de trabajo. En cada proyecto se revisaron una serie de aspectos para comprobar el punto 7 de la norma *Realización del producto*, que fueron:
  - Los Requisitos del proyecto. Se revisó tanto su correcta especificación como el control de los cambios en los requisitos. La revisión afectó a los requisitos funcionales y también a restricciones o requisitos de sistema.
  - La Gestión de los proyectos. La revisión afectó tanto a los aspectos de planificación: tareas a realizar, plazos, personal encargado de realizarlo, recursos utilizados, como a los de seguimiento.
  - La Gestión de la configuración de los proyectos.
  - Los aspectos de Verificación de requisitos. Se comprobó que se realiza la verificación de que el producto desarrollado cumple con sus requisitos y que no tiene errores antes de la entrega al cliente.
- El último día se auditó el punto 8 *Medición, análisis y mejora*. El auditor comprobó los temas referentes a la satisfacción de cliente y a las acciones de mejora. La satisfacción de los clientes se midió utilizando encuestas. Ese mismo día el auditor redactó el informe de auditoría y dio lectura al informe en una reunión con el director general y el personal de calidad de la empresa. El resultado fue, en general, muy satisfactorio, únicamente aparecieron

diversas no-conformidades menores que afectaban a diversos departamentos de las distintas empresas auditadas.

Una vez finalizada la auditoría y entregado el informe, la empresa certificadora dio un plazo de dos meses a las empresas para que realizaran las acciones correctoras que se habían propuesto y para que enviaran el informe de dichas acciones. En este informe, cada empresa debería comunicar las acciones que se habían llevado a cabo para solucionar las no-conformidades detectadas o para evitar que se volvieran a producir. En el caso de no-conformidades graves, en el informe se deberían incluir los documentos del SGC que probaran su corrección.

Las no-conformidades detectadas en la auditoría se solucionaron y se envió el informe de acciones correctoras junto con la documentación administrativa necesaria. El informe fue revisado en primer lugar por el mismo auditor que visitó la empresa. Una vez que dio su visto bueno, el informe de auditoría y el informe de acciones correctoras fueron analizados por un comité de la empresa certificadora. Este comité fue el que al final concedió el certificado a cada una de las empresas.

#### **4.5. Resultados obtenidos**

Una vez finalizado el proyecto QuaSAR se han analizado los resultados de la evaluación en las 8 empresas, se han extraído conclusiones sobre el grado de madurez de los procesos evaluados y se han realizado comparativas entre los datos obtenidos. Para facilitar la exposición e interpretación de los resultados, se ha dividido el análisis derivado de las evaluaciones realizadas en estas 8 empresas en diferentes apartados.

##### 4.5.1. Procesos que han obtenido nivel 1, candidatos a evaluar según niveles superiores

<b>Procesos de la empresa</b>		<b>E1</b>	<b>E2</b>	<b>E3</b>	<b>E4</b>	<b>E5</b>
CUS.1.1	Proceso de preparación de la adquisición				X	X
CUS.1.2	Proceso de selección del proveedor					X
CUS.1.4	Proceso de aceptación del cliente	X				
CUS.2	Proceso de suministro		X	X		X
CUS.3	Proceso de obtención de requisitos	X				X
ENG.1.1	Proceso de análisis y diseño de requisitos de sistema					X
ENG.1.2	Proceso de análisis de requisitos del software					X
ENG.2	Proceso de mantenimiento del software y del sistema					X
SUP.1	Proceso de documentación					X
SUP.2	Proceso de gestión de la configuración					X

Tabla 2. Matriz de procesos – empresas evaluables a nivel 2

La tabla 2 muestra los procesos que han obtenido una puntuación superior al 85% en la evaluación del atributo de realización del proceso y que son candidatos a ser evaluados según el nivel 2 de SPICE.

Si se analiza la tabla por columnas, se puede observar como la empresa E5 destaca en todas sus puntuaciones, que son superiores al 85% para todos los procesos evaluados, excepto para *CUS.1.4 Proceso de aceptación del cliente*. Las empresas que no aparecen en la tabla no tienen ningún proceso a nivel 1 (Proceso realizado).

#### 4.5.2. Los dos procesos con mayor puntuación

Como se puede apreciar en el gráfico de la figura 1, *CUS.2 Proceso de suministro* y *CUS.3 Proceso de obtención de requisitos*, son los dos procesos que han obtenido mayor puntuación. El primero de ellos, *CUS.2*, se encarga de suministrar el software al cliente, mientras que el segundo, *CUS.3*, es el *proceso de identificación y seguimiento de requisitos*. Puesto que el objetivo principal de toda empresa es satisfacer a sus clientes, no es de extrañar que estos procesos, que implican al cliente de manera directa, se consideren esenciales y, como consecuencia de ello, hayan obtenido una mayor puntuación con respecto a los demás procesos evaluados.

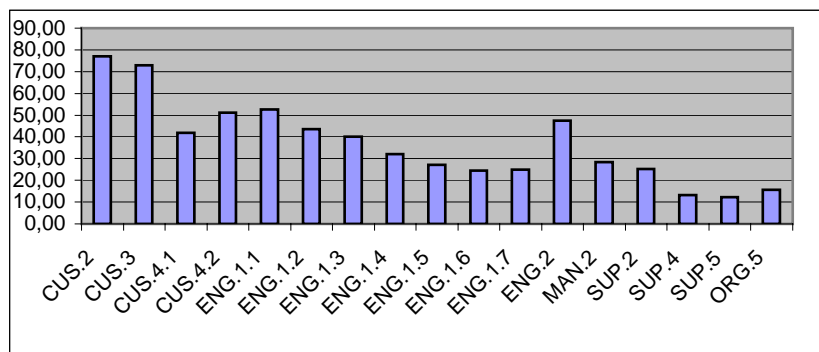


Figura 1. Puntuaciones medias obtenidas

Tomemos *CUS.3* como ejemplo de proceso del que en general se ha obtenido una buena puntuación. Nos interesa priorizar este proceso *CUS.3*, por encima de *CUS.2*, porque es con el que se inicia el desarrollo de un nuevo producto software y por tanto condiciona todas las etapas del ciclo de vida. El gráfico de la figura 2 muestra las puntuaciones obtenidas para el proceso *CUS.3* en cada una de las empresas evaluadas.



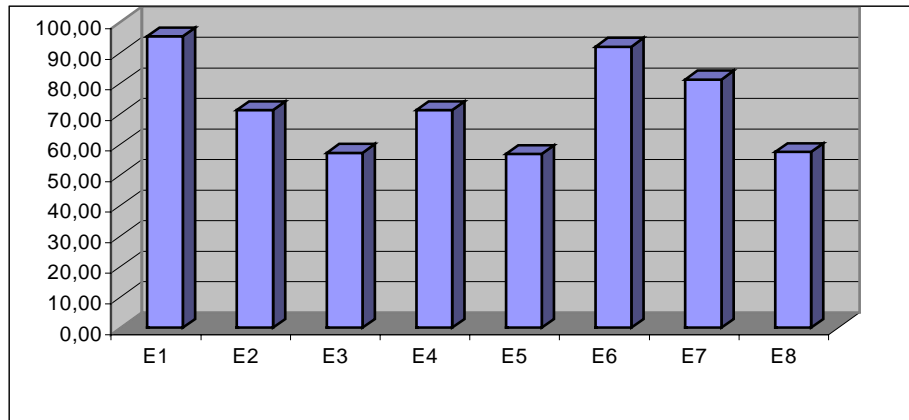


Figura 2. Puntuaciones obtenidas para el proceso CUS. 3, Proceso de Obtención de requisitos

Tal y como se muestra en la figura 2, todas las empresas han obtenido una puntuación superior al 50% para el proceso *CUS.3*. En dos de ellas, en E1 y en E6, la puntuación obtenida es superior al 85%, lo cual significa que solamente en estas dos empresas se podría realizar una evaluación de nivel 2. E2, E4 y E7 han obtenido puntuaciones superiores al 70%. En estas tres empresas una pequeña mejora les permitiría optar a la evaluación de nivel 2. Las empresas E3 y E5, con puntuaciones inferiores al 60%, deberían plantearse la mejora de este proceso.

#### 4.5.3. Procesos seleccionados para la mejora

Tal y como ya se ha mencionado en el apartado 4.3.4 de este artículo, como resultado de las votaciones entre todas las empresas participantes en el proyecto QuaSAR, se eligieron tres procesos a mejorar: el proceso de Pruebas, el proceso de Medición análisis y mejora y el proceso de Gestión de la configuración.

Tomemos el proceso de Pruebas, que según SPICE, abarca los procesos:

- ENG.1.6 Proceso de pruebas de software.
- ENG.1.7 Proceso de pruebas e integración del sistema.
- SUP.4 Proceso de verificación.
- SUP.5 Proceso de validación.

El gráfico de la figura 3 muestra las puntuaciones obtenidas en cada empresa para el proceso de pruebas, considerado éste como la suma de los cuatro procesos de SPICE mencionados anteriormente. Se puede observar que las puntuaciones obtenidas para los dos procesos de Soporte, *SUP.4* y *SUP.5* son inferiores a las obtenidas para los dos procesos de la categoría de Ingeniería, *ENG.1.6* y *ENG.1.7*.

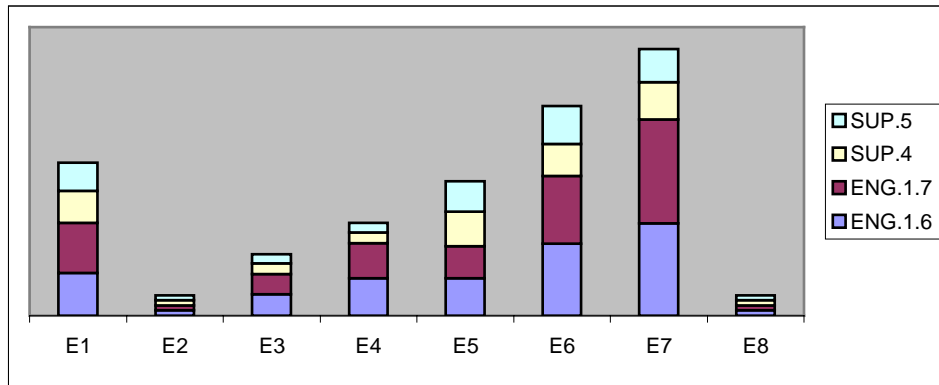


Figura 3. Puntuaciones obtenidas para cada proceso del proceso de pruebas

#### 4.6. Conclusiones del proyecto QuaSAR

Una vez concluido el proyecto podemos afirmar que se han cumplido todos los objetivos previstos al inicio del mismo y que cada una de las partes integrantes ha obtenido un resultado positivo y ha alcanzado sus expectativas.

- Todas las empresas han implantado un sistema de gestión de calidad y han obtenido la certificación según la Norma ISO 9001:2000. Este hecho les ha permitido una mejora a corto plazo de algunos de sus procesos, así como la identificación y planificación de futuras mejoras en otros procesos de la organización, iniciando de esta manera el camino hacia la mejora continua.
- El *Institut d'Innovació Empresarial de les Illes Balears*, ha conseguido fomentar la calidad en empresas del sector específico de desarrollo del software, iniciando una primera experiencia que esperamos haya servido de modelo para futuras colaboraciones, tanto en las *Illes Balears* como en otras comunidades autónomas.
- La empresa de consultoría externa, ha participado en un proyecto innovador en el territorio español y se ha abierto mercado en las Islas.
- Los becarios han tenido la oportunidad de adquirir conocimientos sobre gestión de calidad y de vivir la realidad empresarial. Además, algunos han realizado el proyecto de final de carrera en temas relacionados con la calidad de software y otros han continuado con contrato laboral en la empresa que los acogía como becarios.
- Las participantes de la UIB hemos podido aplicar nuestras investigaciones a un caso real, lo que nos ha permitido refinar los resultados y adaptarlos a la realidad de las pymes.

## 5 Conclusiones

En este artículo se han descrito un conjunto de características que diferencian las pequeñas y medianas empresas de desarrollo de software de las grandes compañías. Se ha presentado un resumen de las principales aportaciones habidas en el campo de la evaluación y mejora de procesos de software a las pymes y se ha descrito la aplicación de un modelo propio, específico para pymes y basado en SPICE, a un conjunto de 8 pymes de Baleares.

## **Agradecimientos**

Este trabajo cuenta con el soporte de los proyectos:

- ✓ CICYT TIC2001-1143-C03-01 "Mejora de los procesos para la toma de decisiones en la gestión de proyectos de ingeniería del software" ARGO.
- ✓ CICYT TIN2004-06689-C03 "Innovación e Integración de Métodos para el Desarrollo y Gestión Cuantitativa de Proyectos Software" IN2GESOFT.

## **Referencias**

- [1] Batista, J., Dias de Figueiredo, A. "SPI in a Very Small Team: a Case with CMM". *Software Process: Improvement and Practice*, vol. 5, nº 4, diciembre 2000, pp. 243-250.
- [2] Beecham, S., Hall, T., Rainer, A. "Software Process Improvement Problems in Twelve Software Companies: An Empirical Analysis". *Empirical Software Engineering*, vol. 8, nº 1, marzo 2003, pp. 7-42. Kluwer Academic Publishers.
- [3] Calvo-Manzano, J. A. Método de mejora del proceso de desarrollo de sistemas de información en la pequeña y mediana empresa. Ph. D. Thesis, Universidad de Vigo, 1999.
- [4] Calvo-Manzano, J. A., Cuevas, G., San Feliu, T., De Amescua, A. Pérez, M. "Experiences in the Application of Software Process Improvement in SMES". *Software Quality Journal*, vol. 10, nº 3, noviembre 2002, pp. 261-273. Kluwer Academic Publishers.
- [5] Cuevas, G., Amescua, A., San Feliu, T., Arcilla, M., Cerrada, J. A., Calvo-Manzano, J. A., García, M. *Gestión del Proceso Software*. Centro de Estudios Ramón Areces, 2003.
- [6] Dyba, T. "Factors of Software Process Improvement Success in Small and Large Organizations: An Empirical Study in the Scandinavian Context". *Proceedings of the European Software Engineering Conference and ACM SIGSOFT Symposium on the Foundations of Software Engineering, 2003*, pp. 148-157.

- [7] *ESPINODE: ESSI (European Software and System Initiative) PIE (Process Improvement Experiments) Nodes.* <<http://www.cordis.lu/esprit/src/stessi.htm>>
- [8] *ESSI: European Software and System Initiative.*  
<[http://www.cordis.lu/esprit/src/essi.htm#ch1\\_1](http://www.cordis.lu/esprit/src/essi.htm#ch1_1)>
- [9] Grunbacher, P. “A software assessment process for small software enterprises”. *Proceedings of the EUROMICRO 97: New Frontiers of Information Technology, conference*, septiembre 1997, pp. 123-128.
- [10] Guerreo, F., Eterovic, Y. “Adopting the SW-CMM in a Small IT Organization”. *IEEE Software*, vol. 21, nº 4, julio-agosto 2004, pp. 29-35.
- [11] Horvat, R.V., Rozman, I. and Gyorkos, J. “Managing the Complexity of SPI in Small Companies”. *Software Process: Improvement and Practice*, vol. 5, nº 1, marzo 2000, pp. 45-54.
- [12] UNE-EN ISO 9001:2000. *Sistemas de gestión de la calidad. Requisitos.* AENOR, 2000.
- [13] ISO/IEC 15504:2004. *Information technology – Process assessment –* International Organisation for Standardization.
- [14] Kautz, K., Westergaard, H. Thaysen, K. “Applying and Adjusting a Software Process Improvement Model in Practice: The Use of the IDEAL Model in a Small Software Enterprise”. *Proceedings of the International Conference on Software Engineering*, junio 2000, pp. 626-633.
- [15] Kilpi, T. “Product management challenge to software change process: preliminary results from three SMEs experiment”. *Software Process: Improvement and Practice*, vol. 3, nº3, septiembre 1997, pp. 165-175.
- [16] Kuvaja, P., Messnarz, R. “BootStrap – a modern software process assessment and improvement methodology”, *Proceedings of the Fifth European Conference on Software Quality*, septiembre 1996, pp. 194-207.
- [17] Leung, H., Yuen, T. “A Process Framework for Small Projects”. *Software Process: Improvement and Practice*, vol. 6, nº 2, junio 2001, pp. 67-83.
- [18] Lied, H. J. “Experience from process improvement in a SME”. *Proceedings of the European Software Process Improvement conference*, octubre 1999.
- [19] Mas, A. Un Nuevo Modelo para la Implantación de un Sistema de Gestión de Calidad en Pymes de Desarrollo de Software basado en SPICE (ISO/IEC 15504). Ph. D. Thesis, Universitat de les Illes Balears, 2005.

- [20] McFeeley, Bob. *IDEAL<sup>SM</sup>: A User's Guide for Software Process Improvement*. Software Engineering Institute, CMU/SEI-96-HB-001, febrero 1996.
- [21] Paulk, M. C., Curtis, B., Chrissis, M. B., Weber, C. *Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute, CMU/SEI-93-TR-24, febrero 1993.
- [22] Richardson, Ita. "Software Process Matrix: A Small Company SPI Model". *Software Process: Improvement and Practice*, vol. 6, n° 3, septiembre 2001, pp. 157-165.
- [23] Richardson, Ita. "SPI Models: What Characteristics are Required for Small Software Development Companies?". *Software Quality Journal*, vol. 10, n° 2, septiembre 2002, pp. 101-114. Kluwer Academic Publishers.
- [24] *Software Process Improvement in Regions of Europe, SPIRE*.  
<<http://www.cse.dcu.ie/spire>>
- [25] TOPS: Toward Organised Software Processes in SMEs. 27977 TOPS – ESPINODE for Central Italy. "Rapid Software Process Assessment to Promote Innovation in SME's".
- [26] Ward, R. P., Fayad, M. E., Laitinen, M. "Software Process Improvement in the Small". *Communications of the ACM*, vol. 44, n° 4, abril 2001, pp. 105-107.
- [27] Wiegers, Karl E. Sturzenberger, D. C. "A Modular Software Process Mini-Assessment Method", *IEEE Software*, vol. 17, n° 1, enero/febrero 2000, pp. 62-69.
- [28] Zahran, S. *Software Process Improvement. Practical guidelines for Business Success*. Addison-Wesley, 1998.



## **¿Cuál es la madurez que necesitarían los procesos para el desarrollo de sistemas de software crítico?**

Patricia Rodríguez Dapena, Josefina Alonso Nocelo, José Carlos Sánchez Domínguez

SoftWcare, S.L

C/. Serafín Avendaño 18 interior, oficina 13

36201 Vigo (Pontevedra)

rodriguezdapena@softwcare.com, alonsonocelo@softwcare.com, sanchezdominguez@softwcare.com

### **Abstract**

Critical systems which function conditions our life need to be verified and certified before its use. The criticism of these systems is founded in the software products related with. The current certification process are, sometimes, insufficient to evaluate all the security and reliability aspects related.

Some international certification organism deals with the development process certification instead of the product or the system. This paper presents how and what must be evaluated respect the development process related with the certification of critics software systems.

### **Resumen**

Los sistemas críticos cuyo funcionamiento condiciona nuestra vida cotidiana necesitan ser verificados, certificados u homologados, antes de su puesta en funcionamiento.

La criticidad de estos sistemas reside cada vez más en los productos software que contienen. Los actuales procesos de certificación resultan en ocasiones insuficientes para evaluar todos los aspectos de seguridad y fiabilidad.

Algunos organismos internacionales de certificación pretenden certificar el proceso de desarrollo en vez del producto o el sistema en sí. Este trabajo presenta cómo y qué se debería evaluar respecto a los procesos de desarrollo para certificar sistemas de software críticos.

**Palabras clave:** Sistemas, software crítico, evaluación de procesos, seguridad, fiabilidad, proceso, certificación, homologación.

## **1 Introducción**

Las entidades de certificación y homologación (FAA [1], etc.) son conscientes de la necesidad de un nuevo enfoque en la certificación para agilizar la evaluación de la seguridad y fiabilidad de sus

sistemas, para que la incorporación de nuevas tecnologías no retrase la puesta en marcha del sistema (pues resulta compleja y difícil la certificación) y sea una ventaja y no un inconveniente.

Para agilizar este proceso se plantea la certificación del proceso en vez del producto. Podría certificarse que las organizaciones que desarrollan estos sistemas dispongan de procesos internos de diseño, pruebas y aseguramiento de la calidad cuyos resultados sean productos con el nivel de seguridad (*safety*) requerido. Este nuevo enfoque permitiría a las entidades certificadoras centrarse más efectivamente en los aspectos críticos de un sistema.

Para lograr esto, deberían relacionarse los modelos existentes de evaluación y/o certificación de la madurez de los procesos software (tales como SPICE o CMM/CMMI para software) con los requisitos de seguridad (*'safety'*) del sistema y definir las exigencias según los niveles de criticidad del software —*Safety Integrity Levels (SILs)*—, definiendo así los 'perfiles de madurez' de los procesos respecto a los diferentes niveles de criticidad de productos software.

## **2 Evaluación de procesos**

El objetivo buscado en la certificación u homologación de sistemas críticos es el aseguramiento de un mínimo riesgo de fallo del sistema (o al menos, un nivel de riesgo aceptable), una vez puesto en funcionamiento.

A principios de los 80, los militares de EE.UU. y del Reino Unido se propusieron mejorar el mecanismo de selección de proveedores de software con el objetivo de detener el creciente costo de software, reducir riesgos en su desarrollo y mejorar la calidad de los productos de software.

En EE.UU., se creó el *Software Engineering Institute (SEI)*, con el objetivo de desarrollar el mecanismo de selección de proveedores. El modelo CMM [7] [8] (cuya primera versión se obtuvo en el año 1991) y el trabajo e impacto de este instituto son bien conocidos ([www.cmu.sei.edu](http://www.cmu.sei.edu))

Por su parte, más adelante, teniendo origen en el Reino Unido se reconoció la necesidad de abordar con mayor rigor el problema de selección de proveedores para los sistemas que dependen en gran medida de software. Se revisaron muchos modelos y métodos existentes en ese momento (Bootstrap, Trillium, etc.), llegándose a un consenso internacional sobre la necesidad y los requisitos para un modelo y método de referencia de evaluación de procesos, constituyendo éste el origen de lo que sería posteriormente la norma ISO 15504.

### **La Norma SPICE (ISO/IEC 15504)**

Hoy día ya existe un estándar ISO, el ISO/IEC TR 15504:1998 [2], que detalla un modelo y un método de referencia para la evaluación de procesos software. Este estándar se encuentra en fase de revisión para su segunda versión [3], y será un modelo de referencia para la evaluación de procesos en general



(no sólo de software) La norma ISO/IEC 15504 [2] [3] (también denominada SPICE —*Software Process Improvement and Capability dEtermination*—, por el proyecto que dio origen a la norma) se caracteriza por:

- Ser aplicable a cualquier organización o empresa.
- Ser independiente de la organización, el modelo del ciclo de vida, la metodología y la tecnología.
- Ser un marco para métodos de evaluación, no un método o un modelo en sí.
- Cubrir diferentes objetivos para la evaluación de procesos:
  - Determinación de la capacidad (niveles de capacidad o de madurez);
  - Mejora de procesos
  - Evaluar el cumplimiento de determinados requisitos del ciclo de vida de desarrollo de software.

La parte 5 de la norma proporciona la guía para un modelo de evaluación de procesos software, de acuerdo con la norma ISO/IEC 12207 [4] [5] para procesos del ciclo de vida del software. La arquitectura SPICE tiene dos dimensiones: procesos y niveles de madurez o capacidad (figura 1).

#### *Dimensión de los Procesos*

Contiene los procesos que se han de evaluar y que, en la parte 5 del estándar, se corresponden con los procesos del ciclo de vida del software, definidos en el estándar ISO/IEC 12207 [4] [5].

Los procesos se agrupan en categorías, en función del tipo de actividad al cual se aplican: Clientes y proveedores (CUS), Ingeniería (ENG), Soporte (SUP), Gestión (MAN) y Organización (ORG).

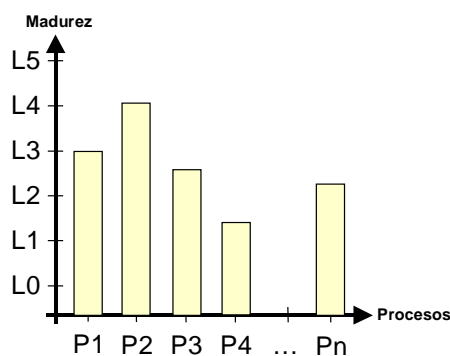


Fig. 1 Dimensiones de SPICE

#### *Dimensión de Niveles de Madurez o Capacidad*

Define una escala de medida para determinar la madurez de un proceso. Cuenta con seis niveles de capacidad o madurez —numerados de 0 a 5— asociados a nueve atributos de procesos (ver tabla 1).

Nivel de capacidad		Atributos de los procesos
0	Incompleto	
1	Realizado	Grado de realización
2	Gestionado	Gestión de la realización Gestión de productos de trabajo
3	Establecido	Grado de definición Grado de institucionalización
4	Predecible	Medida del proceso Control del proceso
5	Optimizado	Grado de gestión del cambio Grado de optimización

Tabla 1 Niveles de capacidad y atributos de los procesos de ISO 15504 [2] [3]

Para evaluar los procesos se utilizan estos atributos para cada proceso que, de cumplirse, le proporcionan mayor o menor nivel de madurez al proceso. Y para evaluar estos atributos se utilizan indicadores que detallan mucho más los atributos facilitando la evaluación de cada proceso. Una vez evaluados los procesos se obtiene lo que se conoce como perfiles de madurez.

Existen otros proyectos a escala internacional para la evaluación de procesos (como CMM y CMMI, desarrollados por el *Software Engineering Institute*) que, al igual que SPICE, están más orientados a la selección o evaluación de proveedores que a la certificación de productos críticos.

A continuación, se presentan los diferentes requisitos de seguridad ('safety') y fiabilidad en diferentes dominios, para analizar qué se certifica respecto a los diferentes niveles de criticidad, para luego compararlos con los modelos de evaluación de procesos existentes y definir sus diferencias.

### 3 Niveles de criticidad de productos software y estándares de desarrollo

Los niveles de criticidad de un sistema software se asignan según la severidad y frecuencia del mal funcionamiento del software durante su operación. Son los denominados riesgos del producto o sistema, de forma que cuanto más severos y frecuentes sean los efectos de sus fallos más altos es el riesgo y más alto será su nivel de criticidad del mismo.

La clasificación del software en diferentes niveles de criticidad puede proporcionar una base para la definición de exigencias más o menos estrictas respecto a los procesos y actividades de desarrollo de software: a mayor criticidad, mayor exigencia.

Los objetivos de esta clasificación son los siguientes:

- Asegurar que los procesos y actividades que se utilizan son lo suficientemente completos para que se adecuen a las necesidades de desarrollo de software más o menos crítico.
- Eliminar costes innecesarios, de modo que el software de baja criticidad no sea desarrollado con excesivos, costosos y sofisticados procesos.
- Que los procesos de software usados en distintos proyectos sean lo más uniformes posibles, para cada nivel de criticidad.

Los niveles de criticidad son diferentes según cada dominio de aplicación y las características de seguridad que se apliquen respectivamente.

A continuación se hace un pequeño análisis de los estándares más representativos de los distintos dominios de aplicación (sistemas electrónicos, aviónica, etc.) con el objetivo de realizar un estudio de las distintas exigencias que estos recomiendan en función de los diferentes niveles de criticidad específicos de cada uno de ellos.

#### *Sistemas Electrónicos*

Como cualquier estándar general de seguridad (*safety*), el estándar IEC 61508 [10] define una serie de niveles de criticidad a partir de la probabilidad media de fallos y recomienda el uso de métodos y técnicas más o menos estrictos según el nivel.

En la tabla siguiente se presenta un ejemplo de técnicas y recomendaciones asociadas a los diferentes niveles de criticidad definidos en este estándar.

<b>Método o técnica</b>	<b>Categoría</b>	<b>SIL1</b>	<b>SIL2</b>	<b>SIL3</b>	<b>SIL4</b>
Uso de estándares de código	Estándares de código	HR	HR	HR	HR
Clases de equivalencia y pruebas de partición	Pruebas de caja negra	R	HR	HR	HR
Pruebas de estrés	Pruebas de funcionamiento	R	R	HR	HR
Análisis de flujo de control	Inspecciones	---	R	R	HR
<i>Leyenda: 'HR' – altamente recomendado; 'R' – recomendado; '---' no es necesario</i>					

Tabla 2 Técnicas y recomendaciones para cada nivel de criticidad para sistemas electrónicos[10]

#### *Locomoción*

El estándar EN 50128 [9] introduce también diversos aspectos de criticidad para el desarrollo de software en el dominio ferroviario. Está basado en el IEC 61508 anteriormente mencionado.

Aviónica

La actividad de desarrollo del software en el dominio de la aviónica según el estándar DO-178B [6] empieza con una evaluación de la seguridad (*safety*) del sistema. Durante esta evaluación, el efecto de un fallo sobre el funcionamiento total del avión es estudiado y analizado, clasificando al software en cinco niveles de criticidad (A, B, C, D, E) según el tipo de efecto del fallo (catastrófico, arriesgado, mayor envergadura, menor envergadura, sin efecto). Además, para cada una de estas clases de software se indican las actividades que se tienen que satisfacer para conseguir la certificación según el estándar DO-178B [6] (ver ejemplo en tabla 3)

Actividad	Clases de software			
	A	B		
El código fuente cumple los requisitos de bajo nivel.	●	●	○	
El código fuente implementa la arquitectura del software.				
El código fuente es verificable.	○	○		
El código fuente es acorde a los estándares.	○	○	○	
La cobertura de las pruebas de todas las estructuras internas del software es completa	●	●		
<b>LEYENDA</b>	● → La actividad debería realizarse por personas u organizaciones independientes. ○ → La actividad debería satisfacerse. (en blanco) → El cumplimiento de la actividad queda a elección del usuario.			

Tabla 3 Ejemplo de actividades del proceso de codificación y pruebas del software en el dominio de la aviónica [10]

Automóvil

En la industria del automóvil, el estándar MISRA [14] define los niveles de control (*controllability categories*) como la capacidad de los ocupantes, y no sólo del conductor, de preservar la seguridad (*safety*) después de un fallo en el vehículo.

MISRA clasifica el software en 4 niveles de criticidad (1-4) y, a su vez, para cada una de estas clases de software define las actividades que se tienen que satisfacer [6] (ver tabla 4).

### Espacio

Cualquier desarrollo de sistemas espaciales en Europa sigue los estándares ECSS (*European Cooperation for Space Standardization*). El estándar ECSS-Q-80B [11] contiene requisitos para la calidad de software y está apoyado por los estándares ECSS-Q-40B [12] y ECSS-Q-30B [13] en cuestiones de seguridad (*safety*) y fiabilidad respectivamente. Este estándar contiene requisitos específicos para software crítico. Es importante destacar que no se refieren a clases de software en particular, aunque sí especifican requisitos para software crítico, añadidos a los demás requisitos para cualquier clase de software.

El estándar ECSS-Q-80B [11] propone una serie de actividades que permitan asegurar la fiabilidad del software crítico, como por ejemplo el uso de técnicas de programación defensivas, la inspección completa del código fuente o la prohibición del uso de características que puedan proporcionar resultados impredecibles.

Como se ha visto en los apartados anteriores cada estándar en cada dominio define exigencias con diferentes grados de rigor según los niveles de criticidad del software. La pregunta ahora radica en cómo relacionar estas exigencias de diferente índole con los perfiles de madurez de los procesos.

## 4 Perfiles de madurez y criticidad del producto software

El alcance y los objetivos de los modelos de evaluación de procesos y de los diferentes estándares de desarrollo de software crítico son muy diferentes. Por tanto es necesario ahondar más en la adaptación de los métodos de evaluación de procesos software, para dar lugar a los perfiles de madurez para la evaluación /certificación de software crítico.

Actividad	Clases de software			
	1	2	3	4
Lenguajes de codificación y compiladores	Utilización de un lenguaje estructurado	Utilización de un subconjunto restringido de lenguaje estructurado. Utilización de compiladores validados	Como para el nivel 2	Uso de compiladores certificados independientemente con reglas formales de sintaxis y semántica
Pruebas	Mostrar que se cumplen los requisitos. Plan de pruebas repetible.	Pruebas de caja negra	Pruebas de caja blanca a todos los módulos de código – midiendo la cobertura. Pruebas de estrés. Análisis estático de sintaxis.	100% pruebas de caja blanca a los módulos. 100% pruebas de los requisitos. 100% pruebas de integración.

Tabla 4 Ejemplo de actividades del proceso de codificación y pruebas en el ámbito del automóvil [16]

Existen opiniones contrapuestas respecto al valor añadido de relacionar los perfiles de madurez respecto a los niveles de criticidad. Según el Departamento de Defensa de los EE.UU. el alcanzar un determinado nivel de madurez de procesos no garantiza un nivel adecuado de calidad en el desarrollo de los sistemas críticos. Sin embargo, los dominios del automóvil y del espacio tienen planes para establecer una relación entre los niveles de criticidad y los niveles de madurez.

En los últimos años, la implantación de sistemas de calidad basadas en la norma ISO9001 en diferentes empresas y organizaciones ha crecido, siendo ahora orientada a procesos. Sin embargo, para aquellas organizaciones que desarrollan sistemas de software crítico, esta norma presenta una serie de problemas pudiendo resultar insuficientes ya que:

- a) Disponer de un sistema de calidad del software acorde con la ISO 9001 no garantiza más allá de poder desarrollar software con el nivel más bajo de criticidad (SIL1)
- b) Obtener una certificación ISO 9001 en una organización correspondería a una madurez de procesos al nivel 3 para todos los procesos del ciclo de vida (tal y como se menciona en diferentes artículos y literatura existente)
- c) La norma ISO 9001 no es específica para software, aunque se dispondrá de la ISO 90003; sin embargo, ésta tampoco es específica para software crítico, resultando todavía insuficiente.

Tanto ISO/IEC TR 15504 [2] como otros modelos de evaluación de procesos software, inciden en la determinación de la capacidad y la mejora de los procesos software en detalle, pero no aseguran las condiciones suficientes para garantizar el nivel de fiabilidad y seguridad (*safety*) necesarios del producto. Se haría pues necesario, modificar estos modelos para poder garantizar los aspectos relacionados con la criticidad del software. Estas adaptaciones son necesarias en relación con:

- a) actividades adicionales para el desarrollo y la verificación de software crítico,
- b) más detalle de los documentos y productos resultantes de cada actividad y,
- c) exigencias más específicas según el nivel de criticidad respecto al personal que desarrolle el sistema,
- d) exigencias específicas respecto a los métodos, técnicas y herramientas que se utilicen,
- e) exigencias para la organización que asegure la fiabilidad y seguridad (*'safety'*) del sistema.

Estas modificaciones afectan a casi todos los procesos y resultan ser exigencias a diferentes niveles de madurez y no sólo referentes a la definición de las actividades del proceso en sí.

## 5 Definición de los perfiles de madurez

La parte 8 de ISO/IEC TR 15504 [2] describe un método general para definir perfiles de madurez. Este método consiste en vincular los procesos con los niveles de criticidad mediante un análisis de la relación existente entre el riesgo de los procesos y el éxito de los proyectos. En la tabla 6, mostrada a continuación, se muestra un ejemplo de estos perfiles de madurez para los procesos de clientes y proveedores según el nivel de criticidad. Como puede comprobarse, para la clase A (la de mayor criticidad) se exige el nivel de madurez 4 que resulta, en cualquier caso, muy exigente. Su cumplimiento puede resultar por tanto inviable.

Proceso		Criticidad			
		A	B	C	D
CUS.2.2	Entrega	4	3	3	1
CUS.3	Análisis de Requisitos	4	3	3	1
CUS.4	Operación	4	4	3	2
CUS.4.1	Uso operacional	4	4	3	2
CUS.4.2	Suporte al cliente	4	4	3	2

Tabla 6 Ejemplo de definición de perfiles de madurez para diferentes niveles de criticidad [15]

El borrador del estándar ECSS-Q-80-02 [15] contendrá un ejemplo de los perfiles de madurez de referencia, para los diferentes proyectos de desarrollo de software de sistemas espaciales en Europa.

## 6 Conclusiones

Existen iniciativas para redirigir la certificación del software como parte de sistemas críticos desde la evaluación del producto a la de sus procesos de desarrollo para así agilizar estos tediosos procesos de certificación especialmente cuando los productos software de estos sistemas son nuevos y cada vez más complejos.

A la vista de lo expuesto anteriormente, para el desarrollo de sistemas de software crítico:

- a) ni la familia de normas ISO 9000 ni los modelos de evaluación de procesos (Ej. ISO/IEC 15504 [2] [3]) resultan suficientes para asegurar la madurez necesaria de los procesos en las organizaciones;
- b) las adaptaciones necesarias a los modelos de evaluación de procesos existentes no son despreciables, incluyendo la dificultad de tener que armonizar las exigencias en los diferentes dominios de aplicación;

- c) es previsible la exigencia de un nivel de madurez elevado en diferentes procesos, pudiendo resultar inasumible para las organizaciones;
- d) tener el nivel necesario de madurez de los procesos no asegura el nivel final requerido de seguridad y fiabilidad en los productos.

La literatura actual muestra un amplio abanico de posibilidades en cuanto a la relación que debe establecerse entre los niveles de madurez de los procesos y la criticidad del software.

A día de hoy, se sigue investigando en la relación existente entre los niveles de madurez del software y la criticidad del software en distintos dominios. Y aunque resulta imprescindible seguir estudiando esta relación con el fin de poder facilitar la certificación de estos sistemas, hay que mantener ciertas reservas respecto a que la certificación de estos sistemas sea posible sólo a través de la evaluación de sus procesos de ciclo de vida.

## **7 Bibliografía**

- [1] Commission on the Future of the U.S. Aerospace Industry. FAA. November 2002. FAA. [www.faa.gov](http://www.faa.gov)
- [2] ISO/IEC TR 15504:1998 *Information Technology – Software Process Assessment*.
- [3] ISO/IEC 15504:2003. *Information Technology – Process Assessment*.
- [4] ISO/IEC 12207:1995/Amd 1:2002. *Information technology -- Software life cycle processes*. December 2002.
- [5] ISO/IEC 12207:1995/Amd 1:2002 Corrigenda. *Information technology -- Software life cycle processes*.
- [6] RTCA/DO-178B. *Software Considerations in Airborne Systems and Equipment Certification*. December 1992.
- [7] *CMMI<sup>SM</sup> for Systems Engineering and Software Engineering (CMMI-SE/SW, V1.1) continuous representation (CMU/SEI-2002-TR-001) and staged representation (CMU/SEI-2002-TR-002)* Software Engineering Institute. December 2001.
- [8] *CMMI<sup>SM</sup> for Software Engineering (CMMI-SW, V1.1) continuous representation (CMU/SEI-2002-TR-029) and staged representation (CMU/SEI-2002-TR-028)*. Software Engineering Institute. August 2002.
- [9] EN 50128:2001. *Railway applications. Communications, signalling and processing systems. Software for railway control and protection systems*. (debe leerse conjuntamente con EN50126 y EN



50129).

- [10] *IEC 61508 Functional safety of electrical/electronic/programmable electronic safety-related system. Parts 1-7.* International Electro-technical Commission. IEC 61508. 1999.
- [11] *ECSS-Q-80B Space product assurance – Software product assurance.* 10 October 2003.
- [12] *ECSS-Q-40B Space product assurance – Safety.* 17 May 2002.
- [13] *ECSS-Q-30B Space product assurance – Dependability.* 8 March 2002.
- [14] *Draft ECSS-Q-80-03 Space product assurance – Guidelines for methods and techniques to support the verification and validation of software dependability and safety.* Draft D1.5, 25 July 2003
- [15] *Draft ECSS-Q-80-02 Space product assurance – Software process assessment and improvement.* Draft 1.5.1. ECSS-Q-80-02 working group <http://www.ecss.nl>
- [16] Jesty, K. Mobley, R. Evans, I. Kendall, “Safety Analysis of Vehicle-Based Systems”. PMISRA. <http://www.misra.org.uk/papers/SCSC00-SA.PDF>

# **Un sondeo sobre la práctica actual de pruebas de software en España**

Luis Fernández Sanz

Depto. de Sistemas Informáticos, Universidad Europea de Madrid

luis.fernandez@uem.es

## **Abstract**

Testing is one of the basic activities for software development and it usually represents the only technique used for verification and validation. There are different international studies about real practice in industry projects and also general analyses of the whole set of software process, but it is difficult to locate specific surveys about testing in the context of Spain. Data of daily practice in the area of testing in software and system development projects collected from professional practitioners in different surveys during the last years are presented in this paper.

Keywords: software testing, software process, education.

## **Resumen**

Las pruebas representan una actividad fundamental en el desarrollo de software y, en muchos casos, suponen prácticamente el único medio empleado en los proyectos para la verificación y validación del software. Existen estudios de ámbito internacional sobre las prácticas reales de pruebas en proyectos de la industria y también análisis generales del conjunto de procesos de software, no se localizan estudios específicos sobre las pruebas del software en el ámbito español. En este artículo se presentan los resultados de las encuestas realizadas en los últimos años a profesionales del desarrollo de aplicaciones y sistemas sobre la aplicación real de las pruebas en su práctica diaria.

**Palabras clave:** pruebas de software, procesos de software, formación.

## **1. Introducción**

Las pruebas de software son seguramente la actividad más común de control de calidad realizada en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas. Aunque un aseguramiento de calidad de software más eficaz debería incluir otras técnicas como, por ejemplo, inspecciones y revisiones (automatizadas o no) de modelos y documentos no ejecutables de las primeras fases de desarrollo, no existe proyecto de desarrollo que no realice de manera más o menos exhaustiva y formal pruebas de software. Las pruebas de software<sup>1</sup> se definen como “una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan y registran y se realiza una evaluación de algún aspecto” [1].

En <http://www.csc.ncsu.edu/faculty/xie/testingresearchsurvey.htm> se puede consultar una amplia variedad de encuestas sobre pruebas de software publicados en revistas científicas si bien esta lista no es exhaustiva ya que tenemos otros ejemplos claros, por ejemplo, en [2] o en encuestas del Quality Assurance Institute (<http://www.qai.com>). También existen propuestas sobre mejora de procesos en pruebas [3], modelos específicos de mejora como TPI [4] y datos sobre las prácticas de las pruebas en los estudios generales sobre mejoras de procesos, por ejemplo, a través del área clave de proceso de Quality Assurance [5]. También existen estudios para evaluar las implicaciones económicas de las técnicas (y sus mejoras) relacionadas con las pruebas y la calidad (puede consultarse la web <http://www.davidfrico.com>) así como ejemplos como el del NIST [6].

## **2. Encuesta**

Los datos que presentamos a continuación se basan en la realización de diferentes encuestas desde 1999 a 2005 realizadas a asistentes tanto a cursos de formación sobre pruebas como a sesiones técnicas y conferencias del grupo de Calidad del Software de la Asociación de Técnicos de Informática ([www.ati.es/gtcalidadsoft](http://www.ati.es/gtcalidadsoft)). Por tanto, inicialmente, las respuestas proceden de personas que ya tienen un cierto interés y motivación por la calidad y las pruebas del software.

---

<sup>1</sup> Esta definición es importante ya que algunos autores hablan de pruebas dinámicas (las pruebas reales) y estáticas (otros controles que no implican ejecución real) lo que puede llevar a confusión.

El cuestionario utilizado se inspiró en modelos utilizados por el Quality Assurance Insitutue ([www.qai.com](http://www.qai.com)) en algunas de sus encuestas sobre buenas prácticas de proceso de pruebas, añadiendo datos básicos de población y situación del individuo que responde. El núcleo de la encuesta se apoya en 20 preguntas que permiten analizar la madurez del proceso de pruebas en base al número de respuestas negativas. Se proponen 5 niveles de clasificación en orden ascendente de madurez y buenas prácticas: arte, habilidad individual, proceso definido, organización avanzada de pruebas y calidad de primera clase. La descripción de estos niveles se incluye en el anexo 1.

A partir de 2003, se incluyeron algunas preguntas adicionales sobre conceptos relacionados con buenas prácticas en las pruebas de software. El modelo usado a partir de 2003 se incluye en el anexo 1.

En total se han obtenido respuestas de 102 profesionales del desarrollo de software que incluyen una amplia variedad de perfiles: desde directores o gerentes a analistas y programadores y técnicos y expertos de calidad<sup>2</sup>.

Los datos de población básicos de quienes de los entrevistados son los siguientes:

- Corresponden a expertos que trabajan en desarrollo para una gran variedad de sectores (aunque un 40% de encuestados no aclara este dato):
  - 17,65% corresponde a banca y finanzas
  - 10,78% a informática y telecomunicaciones
  - 8,82% a industria y energía
  - 6,86% a turismo y hostelería
  - Una variedad de sectores como I+D, empresas públicas, seguros, aeroespacial, etc. totalizan el 16,67%.
- Los lenguajes o entornos en los que trabajan también incluyen una gran variedad de plataformas y tecnologías (aunque un 24,5% de las respuestas no aclara el dato)<sup>3</sup>:
  - Tecnología Java: 19,61%
  - COBOL: 13,73%
  - C++: 11,76%
  - Tecnología .net y ASP: 10,78%
  - Tecnología Visual Basic: 9,80%

---

<sup>2</sup> Este dato se empezó a solicitar oficialmente en 2005 por lo que la estadística es limitada en cuanto a datos formales ya que existía información informal en las encuestas anteriores que confirman la gran variedad de puestos abordados.

<sup>3</sup> La suma de los porcentajes no totaliza el 100% sino una cifra mayor ya que las opciones a escoger no son excluyentes: por ejemplo, hay profesionales que deben trabajar en dos o más de las tecnologías indicadas.

- Natural: 7,84%
- Oracle: 6,86%
- Powerbuilder: 5,88%
- PHP: 5,88%
- Otros (Delphi, C, Access, Ada, ERPs, etc.): 9,80%
- Los entornos de infraestructura en la que deben abordar el trabajo de pruebas es también variado<sup>4</sup>:
  - Grandes ordenadores: 20,59%
  - Estaciones de trabajo o PCs: 47,06%
  - Entornos Cliente/Servidor: 30,39%
  - No contesta: 16,67%

Lamentablemente, la muestra, aunque significativa para aportar ideas sobre la situación de la práctica real en pruebas de software en España, no constituye una base numérica suficientemente amplia como para sustentar el análisis, con garantías, de la evolución de los resultados a lo largo de los años de aplicación de la encuesta. De hecho, los años 2003 y 2005 fueron los que registraron más participantes con 27 y 39 respectivamente.

### **3. Resultados**

A continuación se presentan los resultados estadísticos elementales que se pueden deducir de los datos recogidos en las encuestas. Obviamente, la información principal recogida por la encuesta se refiere a la valoración de la madurez y buenas prácticas del proceso de pruebas aplicado en la organización correspondiente. Desde este punto de vista, los datos revelan la siguiente clasificación en los niveles fijados según las respuestas negativas a los 20 ítems planteados en el cuestionario:

- Proceso como arte (17-20 respuestas negativas): 14,71%
- Habilidad individual (13-16 negativas): 46,08%
- Proceso definido (9-12 negativas): 28,43%
- Organización avanzada (5-8 negativas): 8,08%
- Calidad de primera clase (0-4 negativas): 1,96%

---

<sup>4</sup> La suma de los porcentajes no totaliza el 100% sino una cifra mayor ya que las opciones a escoger no son excluyentes: por ejemplo, hay profesionales que deben trabajar en dos o en los tres entornos indicados.

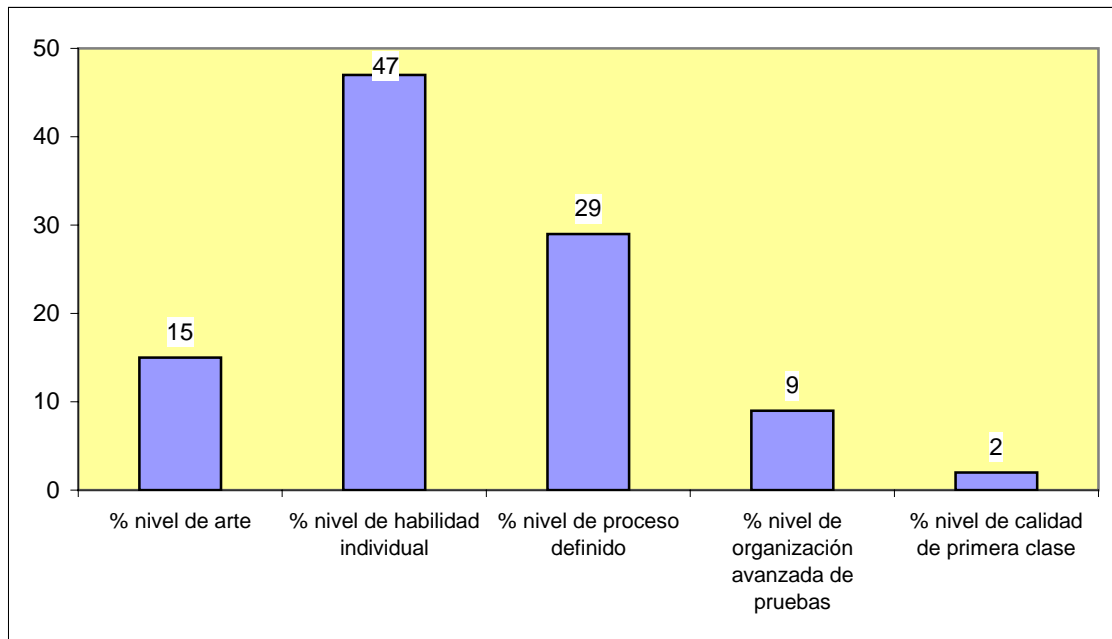


Figura 1. Clasificación de resultados por niveles de proceso

En cuanto a medidas de tendencia sobre el número de respuestas positivas (es decir, que confirman que las buenas prácticas se han implantado), son las siguientes:

- Promedio: 6,76
- Varianza: 3,64

Por tanto, de 20 buenas prácticas identificadas para el proceso de pruebas, como media, menos de 7 están implantadas en las organizaciones de los encuestados.

En cuanto a la distribución de respuestas positivas por cada pregunta del cuestionario, los detalles se encuentran en el Anexo 2. Los ítems que revelan una mayor implantación (mayor o cercana al 50% de las respuestas) en la práctica de las organizaciones de desarrollo son:

- 7 (¿Se valida que, además de que están bien implementadas, se cumplen las expectativas del cliente?): 65,69%.
- 6. (¿Se valida que las especificaciones están correctamente especificadas?): 49,02%.
- 9. (¿El personal de pruebas informa de los defectos al equipo de desarrollo y no a otras partes como la dirección?): 49,02%.

Por el contrario, son numerosas las prácticas que tienen una implantación débil en el conjunto de encuestados ya que no llegan al 20% de las respuestas:

- 3. (¿Existe y se usa un estándar de pruebas de unidad?): 19,61%.
- 15. (¿Hay un proceso de mejora de procesos de pruebas?): 18,63%.
- 20. (¿Supone el uso de herramientas automatizadas de prueba un componente significativo del proceso de pruebas?): 18,63%.
- 11. (¿Se establecen objetivos mensurables de pruebas para cada sistema sometido a pruebas?): 16,67%.
- 16. (¿Existe un nombre o código de identificación los defectos?): 16,67%.
- 10. (¿El personal de pruebas identifica los riesgos de negocio antes de desarrollar el plan de pruebas?): 15,69%
- 18. (¿Se usan métricas (p.ej., defectos/KLOC) para planear y evaluar los procesos de pruebas?): 7,84%.

En general, se percibe que los ítems relacionados con medición y registro de datos tienen tendencia a situarse en los puestos más bajos de implantación, confirmando que éstos son aspectos en general descuidados en las organizaciones. Así mismo, la mejora de proceso de prueba, normalmente inscrita en procesos de mejora general, también se sitúa en niveles bajos coherentes con la implantación actual de este tipo de iniciativas en España.

Por último, podemos indicar que los datos recogidos a partir de 2003 sobre el conocimiento de conceptos básicos recomendados para las pruebas revelan que existe una clara tendencia a no aplicar (e incluso desconocer) la manera recomendada y más eficaz de realizar las pruebas. Así, se plantearon las distintas preguntas del cuestionario sobre aspectos muy prácticos de actitud ante las pruebas mediante dos opciones claramente poco apropiadas (pero que se intuían bastante generalizadas por experiencia previa del autor en formación de pruebas) y una última opción libre para indicar una alternativa correcta<sup>5</sup> (ver anexo 1). Para evitar ambigüedades en la comprensión de las distintas alternativas se dieron las aclaraciones necesarias en cada caso. Los resultados obtenidos fueron los que se presentan a continuación, agrupados por cada una de las preguntas realizadas:

---

<sup>5</sup> Todas las respuestas alternativas recogidas fueron correctas en las cinco preguntas realizadas.

- ¿Cuándo diseño las pruebas?
  - Justo antes de ejecutarlas: 13,64%.
  - Cuando ya tengo código: 45,45%.
  - Otro enfoque (respuesta correcta)<sup>6</sup>: 34,85%.
  - No contesta: 6,06%.
- ¿Qué enfoque aplico a las pruebas?:
  - Simplemente ir ejecutando: 25,76%.
  - Diseñar un poco antes de ejecutar: 45,45%.
  - Otro enfoque (respuesta correcta)<sup>7</sup>: 15,15%.
  - No contesta: 13,64%.
- ¿Cómo planifico las pruebas?
  - Dejo algo de tiempo al final: 27,27%.
  - Aplico todo el tiempo que me queda tras el desarrollo: 28,79%.
  - Otro enfoque (respuesta correcta)<sup>8</sup>: 33,33%.
  - No contesta: 10,61%.
- ¿Cómo planteo las pruebas?
  - Como medio para demostrar que el software funciona: 43,94%.
  - Como manera de comprobar si he olvidado algo: 24,24%.
  - Otro enfoque (respuesta correcta)<sup>9</sup>: 22,73%.
  - No contesta: 9,09%.
- ¿Cómo gestiono las pruebas?
  - Simplemente voy ejecutando las pruebas: 25,76%.
  - Procuero anotar las pruebas para no repetirlas: 43,94%.
  - Otro enfoque (respuesta correcta)<sup>10</sup>: 21,21%.
  - No contesta: 9,09%.

En cualquier caso, sobre la muestra de 39 personas de 2005 se les preguntó si habían recibido formación específica en pruebas de software. El 23,08% declaró que sí

---

<sup>6</sup> La respuesta correcta debería comentar que las pruebas deben diseñarse en cuanto se tiene la especificación o análisis de la aplicación.

<sup>7</sup> La respuesta correcta debe centrarse en la generación de un diseño previo muy anterior a la ejecución.

<sup>8</sup> La respuesta correcta debe recoger la idea de una planificación cuidadosa teniendo en cuenta que en muchos proyectos se consume porcentajes del 30-40% del esfuerzo en el ciclo de pruebas y depuración.

<sup>9</sup> La filosofía correcta debe centrarse en las pruebas como medio para detectar defectos de todo tipo [7].

<sup>10</sup> La respuesta correcta debe recoger de una lista diseñada de casos, el registro de resultados y la medición de la cobertura de pruebas alcanzada.



frente a un 61,54% que indicó que no había recibido tal formación (el 15,38% no contestó).

Realizando una correlación básica entre formación y aciertos en las preguntas de conceptos de buenas prácticas se obtiene la siguiente tabla:

	Formación previa	Sin formación previa
Correcta nº1	7 (77,7%)	6 (25%)
Correcta nº2	4 (44,4%)	2 (8,3%)
Correcta nº3	4 (44,4%)	5 (20,8%)
Correcta nº4	5 (55,5%)	3 (12,5%)
Correcta nº5	4 (44,4%)	3 (12,5%)

Tabla 1. Correlación entre formación y respuestas correctas

Dado que se trata de muestras pequeñas, no se recomienda aplicar técnicas de correlación más formales para un análisis más riguroso. No obstante, se percibe que quienes tenían formación previa han obtenido proporcionalmente puntuaciones mejores que los que no la tenían.

#### **4. Conclusiones**

El presente estudio pretende aportar información sobre la realidad de la práctica de las pruebas de software en España. Las muestras recogidas, aunque son reducidas, cuentan con suficiente variedad de entornos, tecnologías y sectores como para aportar una visión significativa de la realidad diaria en este aspecto.

A través del estudio de procesos de software siguiendo modelos conocidos como CMMi o ISO 15504, se suele tener conciencia de lo mucho que aún hay que hacer para contar con métodos de trabajo apropiados para el desarrollo de software relevante. Por ello, no sorprende demasiado que el área de las pruebas ( con las 20 preguntas sobre el proceso) también esté pendiente de esta mejora aunque quizás suele haber menos conciencia de lo que se puede hacer y menor motivación ya que se considera una actividad tradicionalmente desagradable y poco creativa en comparación con el desarrollo. Lo que quizás resulta especialmente llamativo son las carencias conceptuales reveladas en el estudio de las actitudes básicas ante las pruebas, pero que también es coherente con otros estudios realizados sobre la actitud ante la calidad del software [8].

A través de distintas iniciativas se está tomando conciencia e incrementando la formación sobre pruebas donde podemos resaltar las actividades del Grupo de Calidad del Software de ATI. Además, actualmente se está estableciendo con ayuda del Ministerio de Educación y Ciencia una red de trabajo para progresar en la I+D y en la difusión sobre las pruebas del software que podrá aportar más acciones y recursos. Se cuenta con la participación de representantes de universidades, institutos tecnológicos así como empresas comercializadoras de soluciones tecnológica y de consultoría.

## **5. Referencias**

- [1] IEEE, “IEEE Std. 610 Computer dictionary”, IEEE Computer Society, 1990.
- [2] Geras, A.M.; Smith, M.R; Miller J., “A survey of software testing practices in Alberta”, Canadian Journal of Electrical and Computer Engineering, vol. 29, nº 3, 183–191, 2004.
- [3] Kit, E., “Software testing in the real world : improving the process”, Addison-Wesley, 1999.
- [4] Koomen, T.; Pol M., “Test Process Improvement: A practical step-by-step guide to structured testing”, ACM Press, 1999.
- [5] SEI, “Software CMM® CBA IPI and SPA Appraisal Results 2003 Mid-Year Update”, Software Engineering Institute, Carnegie Mellon University, 2003.
- [6] NIST, “Planning Report 02-3. The Economic Impacts of Inadequate Infrastructure for Software Testing”, National Institute of Standards & Technology, 2002.
- [7] Myers, G.J., “The art of software testing”, John Wiley & sons, 1979.
- [8] Fernández, L., “Análisis de las expectativas de los alumnos en los cursos sobre calidad del software”, Actas de las III Jornadas de Informática, 177-186, 1997.

## Anexo 1

El siguiente modelo de cuestionario se empleó a partir de 2003. Sólo se reproducen las preguntas sobre pruebas (no las de población y complementarias ).

- **¿Cuándo diseño las pruebas?**  Justo antes de ejecutarlas  Cuando ya tengo código  Otro: \_\_\_\_\_
- **¿Qué enfoque aplico en las pruebas?**  Simplemente ir ejecutando  Diseñar un poco antes de ejecutar  Otro enfoque: \_\_\_\_\_
- **¿Cómo planifico las pruebas?**  Dejo algo de tiempo al final para hacer algunas pruebas  Aplico todo el tiempo que me queda desde que acaba el desarrollo  Otro: \_\_\_\_\_
- **¿Cómo planteo mis pruebas?**  Como un medio para demostrar que el software funcionan  Como una manera de ver si he olvidado comprobar algo  Otro: \_\_\_\_\_
- **¿Cómo gestiono mis pruebas?**  Simplemente voy ejecutando pruebas  Procuro anotar las pruebas durante la sesión para no repetirlas  Otro: \_\_\_\_\_

PREGUNTA	RESPUESTA			Notas o comentarios
	SÍ	NO	N/A	
1. ¿Hay un responsable de procesos de prueba en la organización?				
2. ¿Existe y se usa un estándar de planes de prueba?				
3. ¿Existe y se usa un estándar de pruebas de unidad?				
4. ¿Existe y se usa un estándar de informes de prueba?				
5. ¿El proceso de planificación y ejecución de pruebas es paralelo al proceso entero de desarrollo de software (es decir, las actividades de prueba empiezan y terminan con el ciclo)?				
6. ¿Se valida que las especificaciones están correctamente especificadas?				
7. ¿Se valida que, además de que están bien implementadas, se cumplen las expectativas del cliente?				
8. ¿El personal de pruebas comprueba la precisión y la compleción de documentos de desarrollo (ej. requisitos y diseño)?				
9. ¿El personal de pruebas informa de los defectos al equipo de desarrollo (y no a otras partes como la dirección)?				
10. ¿El personal de pruebas identifica los riesgos de negocio antes de desarrollar el plan de pruebas?				
11. ¿Se establecen objetivos mensurables de pruebas para cada sistema sometido a pruebas?				
12. ¿Están los objetivos asociados a riesgos de negocio?				
13. Los defectos descubiertos en las pruebas ¿se registran, resumen y utilizan para mejorar los procesos de desarrollo y de prueba?				
14. ¿El personal de pruebas ha definido expectativas de defectos basadas en experiencias previas?				
15. ¿Hay un proceso de mejora de procesos de pruebas?				
16. ¿Existe un nombre o código de identificación los defectos?				
17. ¿La organización registra, resume y utiliza los datos de fallos para evaluar la efectividad de los procesos de pruebas para producir software de calidad?				
18. ¿Se usan métricas (p.ej., defc/KLOC) para planear y evaluar los procesos de pruebas?				
19. ¿Existe procesos de formación del personal de pruebas?				
20. ¿Supone el uso de herramientas automatizadas de prueba un componente significativo del proceso de pruebas?				

Para permitir la auto evaluación rápida de las respuesta para los profesionales que contestaron el cuestionario, se indican las reglas para clasificar la organización de pruebas en función de las respuestas al misma.

### **Reglas para la interpretación de resultados**

Número de respuestas negativas	Evaluación	Explicación
17-20	Pruebas artísticas	En la organización, las pruebas dependen mucho de la habilidad y creatividad individual del personal. No existen guías. El resultado debería ser unas pruebas inconsistentes (en algunos casos, son excelentes y en otros deficientes). Los clientes y usuarios no pueden confiar en la efectividad de las pruebas para asegurar la calidad de los sistemas.
13-16	Pruebas como destreza	Existe un proceso de pruebas. Existen algunos procedimientos, algunos estándares y, normalmente, un plan de pruebas generalista. No obstante, el personal tiende a desviarse del plan y se centra en descubrir defectos e informa de ellos a los desarrolladores. Los usuarios quieren pruebas de aceptación porque no pueden confiar en el proceso de pruebas del desarrollador.
9-12	Pruebas con un proceso de pruebas definido	El proceso de pruebas está bien definido pero no se ejecuta apropiadamente. Es típica el caso de que si hubiera más tiempo, las pruebas serían más efectivas. Las pruebas están todavía basadas en especificaciones y el énfasis se pone en la prueba de los requisitos. Normalmente no hay informe definitivo al final de las pruebas sobre el estado del software. La conclusión general suele ser si el software está listo o no para explotación.
5-8	Organización avanzada de pruebas	La organización se centra en objetivos de prueba claros y se optimizan el uso de recursos de prueba para el logro de dichos objetivos. Se pone énfasis en las deficiencias del proceso de pruebas a través del análisis de resultados de pruebas y el proceso se mejora en consecuencia.
0-4	Organización de calidad de nivel mundial	El personal de pruebas hacen casi todo bien. Las pruebas se basan en la reducción de riesgos, se someten a medición, el proceso está bien definido y los defectos se registran, se resumen, se analizan y se utilizan para la mejora del proceso. El coste de las pruebas puede disminuir considerablemente y los usuarios/clientes confían en el proceso de pruebas para la calidad del software más que basarse en las pruebas de aceptación para determinar si el software funciona.

## Anexo 2

A continuación se ofrece la tabal detallada de respuestas obtenidas para cada ítem del cuestionario referido a buenas prácticas de proceso de pruebas.

Detalles de encuesta	Sí	No	NA	%Sí	%No	%NA
1. ¿Hay alguien responsable de los procesos de prueba en la organización?	38	40	24	37,25%	39,22%	23,53%
2. ¿Existe y se usa un estándar de planes de prueba?	23	55	24	22,55%	53,92%	23,53%
3. ¿Existe y se usa un estándar de pruebas de unidad?	20	58	24	19,61%	56,86%	23,53%
4. ¿Existe y se usa un estándar de informes de prueba?	26	51	25	25,49%	50,00%	24,51%
5. ¿El proceso de planificación y ejecución de pruebas es paralelo al proceso entero de desarrollo de software (es decir, las actividades de prueba comienzan y terminan con el inicio y fin del ciclo)?	37	42	23	36,27%	41,18%	22,55%
6. ¿Se valida que las especificaciones están correctamente especificadas?	50	27	25	49,02%	26,47%	24,51%
7. ¿Se valida que, además de que están bien implementadas, se cumplen las expectativas del cliente?	67	12	23	65,69%	11,76%	22,55%
8. ¿El personal de pruebas comprueba la precisión y la compleción de documentos de desarrollo como requisitos y diseño?	24	42	36	23,53%	41,18%	35,29%
9. ¿El personal de pruebas informa de los defectos al equipo de desarrollo (y no a otras partes como la dirección)?	50	21	31	49,02%	20,59%	30,39%
10. ¿El personal de pruebas identifica los riesgos de negocio antes de desarrollar el plan de pruebas?	16	51	35	15,69%	50,00%	34,31%
11. ¿Se establecen objetivos mensurables de pruebas para cada sistema sometido a pruebas?	17	54	31	16,67%	52,94%	30,39%
12. ¿Están los objetivos asociados a riesgos de negocio?	22	43	37	21,57%	42,16%	36,27%
13. Los defectos descubiertos en las pruebas ¿se registran, resumidos y utilizados para mejorar los procesos de desarrollo y de prueba?	28	47	27	27,45%	46,08%	26,47%
14. ¿El personal de pruebas ha definido expectativas de defectos basadas en experiencias previas?	24	52	26	23,53%	50,98%	25,49%
15. ¿Hay un proceso de mejora de procesos de pruebas?	19	56	27	18,63%	54,90%	26,47%
16. ¿Existe un nombre o código de identificación los defectos?	17	56	29	16,67%	54,90%	28,43%
17. ¿La organización registra, resume y utiliza los datos de fallos para evaluar la efectividad de los procesos de pruebas para producir software de calidad?	21	50	31	20,59%	49,02%	30,39%
18. ¿Se usan métricas (p.ej., defc/KLOC) para planear y evaluar los procesos de pruebas?	8	64	30	7,84%	62,75%	29,41%
19. ¿Existe procesos de formación del personal de pruebas?	21	54	27	20,59%	52,94%	26,47%
20. ¿Supone el uso de herramientas automatizadas de prueba un componente significativo del proceso de pruebas?	19	53	30	18,63%	51,96%	29,41%