

Revista
Española de
Innovación,
Calidad e
Ingeniería del Software



Volumen 6, No. 1, abril, 2010

Web de la editorial: www.ati.es

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

E-mail: calidadsoft@ati.es

ISSN: 1885-4486

Copyright © ATI, 2010

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

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

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



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

Editores

Dr. D. Luís Fernández Sanz (director)

Departamento de Ciencias de la Computación, Universidad de Alcalá

Dr. D. Juan José Cuadrado-Gallego

Departamento de Ciencias de la Computación, Universidad de Alcalá

Miembros del Consejo Científico

Dr. Dña. Idoia Alarcón

Depto. de Informática
Universidad Autónoma de Madrid

Dr. D. José Antonio Calvo-Manzano

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

Dra. Tanja Vos

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

Dña. M^a del Pilar Romay

CEU Madrid

Dr. D. Alvaro Rocha

Universidade Fernando Pessoa
Porto

Dr. D. Oscar Pastor

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

Dra. Dña. María Moreno

Depto. de Informática
Universidad de Salamanca

Dra. D. Javier Aroba

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

D. Guillermo Montoya

DEISER S.L.
Madrid

Dr. D. Pablo Javier Tuya

Depto. de Informática
Universidad de Oviedo

Dra. Dña. Antonia Mas

Depto. de Informática
Universitat de les Illes Balears

Dr. D. José Ramón Hilera

Depto. de Ciencias de la Computación
Universidad de Alcalá

Dra. Raquel Lacuesta

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

Dra. María José Escalona

Depto. de Lenguajes y Sist. Informáticos
Universidad de Sevilla

Dr. D. Ricardo Vargas

Universidad del Valle de México
México

Contenidos

The logo for REICIS (Revista Española de Innovación, Calidad e Ingeniería del Software) is displayed in white capital letters on a black rectangular background.

Editorial	4
<i>Luís Fernández-Sanz, Juan J. Cuadrado-Gallego</i>	
Presentación	5
<i>Luis Fernández-Sanz</i>	
Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías Ágiles y Software Libre	6
<i>José Manuel Navarro y Javier Garzás</i>	
Metodología para la evaluación de la calidad en los modelos UML	16
<i>Moisés Rodríguez, Javier Verdugo, Ramón Coloma, Marcela Genero y Mario Piattini</i>	
Sección Actualidad Invitada:	36
Sostenibilidad del software: más allá de Green IT	
<i>Guillermo Montoya, DEISER</i>	

Editorial

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

La calidad del software sigue siendo un elemento fundamental para la competitividad de las empresas. Constituye uno de los tres ejes esenciales de la gestión junto con el tiempo y los recursos económicos. Aunque son variadas las iniciativas que desde hace muchos años han permitido avanzar en la mejora de las prácticas de desarrollo, mantenimiento y gestión de software, también es cierto que existen numerosos campos dentro de la calidad de las aplicaciones que no han atraído tanto interés como los ya conocidos de mejora de procesos, métricas, aspectos técnicos de las pruebas, etc.; por ejemplo, el análisis cuantificado y empírico de los recursos humanos en la calidad del software o la adaptación de los modelos generales de calidad a entornos específicos como los ERP, los sistemas móviles, los entornos de e-learning, etc. Otra de las necesidades más importantes es contar con información, no sesgada por intereses comerciales, de la implantación real de sistemas y tecnologías así como de los problemas que se experimentan diariamente en los proyectos de software y de los factores que influyen realmente en la productividad, la calidad y en la sostenibilidad del desarrollo de software. Desde REICIS, con la colaboración del grupo de calidad del software de ATI (www.ati.es/gtcalidadsoft), procuraremos que este tipo de estudios y análisis se pueda realizar y que sea publicado tanto en la revista como en la sección de estudios y encuestas de dicho grupo (<http://www.ati.es/spip.php?rubrique336>).

Dentro de esta línea de potenciación de la actividad de estudios y análisis, acogemos con gran esperanza la creación de la red especial de interés en la calidad del software (Software Quality Special Interest Network) por parte de CEPIS (www.cepis.org) para contribuir a que estos aspectos aún poco explorados de esta disciplina puedan ser estudiados gracias a la cooperación europea. Este enfoque proporciona no sólo un mayor potencial de conocimiento sino también la necesaria perspectiva internacional imprescindible para obtener resultados significativos para el panorama competitivo actual. Esperamos que esta iniciativa (<http://www.cepis.org/index.jsp?p=827&n=1141>) proporcione resultados de interés en este año 2010 que transmitiremos puntualmente a REICIS.

Luis Fernández Sanz
Director

Este número de REICIS publicados contribuciones remitidas directamente por sus autores a nuestra revista para su revisión por el comité editorial que decidió finalmente aceptarlas para publicación.

En el trabajo “Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías ágiles y software libre”, José Navarro y Javier Garzás nos presentan una alternativa de implementación de las prácticas determinadas por el modelo CMMI-DEV basada en la aplicación de métodos ágiles con el soporte de herramientas y soluciones de software libre. Este tipo de experiencias puede ayudar a las pequeñas y medianas empresas a afrontar con mayor facilidad el paso a un esquema maduro de procesos de software.

En el segundo trabajo, un equipo de investigadores y profesionales de Alarcos Quality Center y de la Universidad de Castilla-La Mancha presentan un marco de trabajo para la evaluación de calidad del software basada en los modelos UML asociados a su desarrollo. El trabajo incorpora las necesarias referencias a las normas ISO relevantes en el campo de la evaluación de la calidad de software.

Finalmente, en la columna de Actualidad Invitada, Guillermo Monotoya, CEO de DEISER y miembro del grupo de Calidad del Software de ATI, presenta una idea novedosa: la sostenibilidad del software y su desarrollo. Siguiendo la idea de la Green IT donde gran parte del carácter verde y sostenible de la informática reside sobre todo en la optimización del consumo energético de los ordenadores, la sostenibilidad del software apela a la conjunción de factores que cuiden el aspecto económico, social y ecológico involucrado en el software.

Luis Fernández Sanz

Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías ágiles y software libre

José Manuel Navarro

Unkasoft

jmnavarro@unkasoft.com

Javier Garzás

Kybele Consulting y Universidad Rey Juan Carlos

javier.garzas@kybeleconsulting.com / javier.garzas@urjc.es

Resumen

En la actualidad, la industria del software está formada principalmente por pymes (pequeñas y medianas empresas) y micropymes (Pymes de aproximadamente 20 empleados). En este tipo de empresas la calidad del software es esencial, siendo la mejora de procesos software una actividad que desean implementar con el objetivo de incrementar la calidad y capacidad de sus procesos y, en consecuencia, la calidad de sus productos y servicios. Sin embargo, la aplicación de los modelos de mejora de procesos referentes en la actualidad (CMMI e ISO 15504) en pymes es muy difícil ya que supone para estas una gran inversión en dinero, tiempo y recursos. Este tipo de empresas necesita prácticas de ingeniería del software adaptadas a su tamaño y tipo de negocio. En este contexto, el objetivo de este artículo es plantear dicha problemática y presentar una experiencia sobre como metodologías ágiles, software libre y CMMI pueden integrarse.

Palabras clave: mejora de procesos software, PYME, CMMI, métodos ágiles.

Experience of implementation of CMMI-DEV v1.2 in a SME with agile methodologies and free software

Abstract

Nowadays, the software industry consists mainly of SMEs (small and medium enterprises) and micro-SMEs (SMEs approximately 20 employees). In this type of enterprise software quality is essential, and software process improvement is an essential activity to increase the quality and capability of its processes and, consequently, the quality of their products and services. However, the implementation of process improvement models (CMMI and ISO 15504) in SMEs is very difficult and involved for such a large investment in money, time and resources. Such enterprises need software engineering practices appropriate to their size and type of business. In this context, this paper is aimed at outlining this problem and presenting an experience on how agile methods, open source and CMMI can be integrated.

Keywords: Software Process Improvement, SMEs, CMMI, Agile Methods.

Navarro, J.M., Garzás, J., "Experiencia en la implantación de CMMI-DEV v1.2 en una micropyme con metodologías Ágiles y Software Libre", REICIS, vol. 6, no.1, 2010, pp.6-16. Recibido: 25-5-2009; revisado: 19-10-2009; aceptado: 3-11-2009

1. Introducción

La industria del software en España está formada principalmente por pymes y micropymes, empresas que suponen cerca del 80% del sector, donde incluso el 85% tienen menos de diez empleados¹, porcentaje que está aumentando cada vez más, en parte debido a la actual tendencia hacia la externalización y el “nearshoring” [1].

La mejora de procesos software es una actividad que las pymes desean implementar con el objetivo de incrementar la calidad y capacidad de sus procesos [2] y, en consecuencia, la calidad de sus productos y servicios. Y para mejorar sus procesos las empresas están utilizando modelos como CMMI-DEV [3] e ISO 15504 [4], ambos modelos de referencia en España [5], contando, en el caso de CMMI, con 105 evaluaciones realizadas en España².

Sin embargo, numerosos estudios [6-8] muestran que la aplicación de estos modelos en las pymes es muy difícil ya que supone para estas una gran inversión en dinero, tiempo y recursos. Este tipo de empresas necesita prácticas de ingeniería del software adaptadas a su tamaño y tipo de negocio [9] [10] [11]. Y en este sentido, para apoyar a las pequeñas empresas en la mejora de procesos, se están desarrollando varias iniciativas, como la ISO/IEC 29110 para micropymes y pequeños grupos [12], que no estará lista hasta 2010; u otras como ITmark, Competisoft, etc., que por la mayor difusión de CMMI [5] son menos demandadas.

Y por otro lado, uno de los paradigmas que más se ha adaptado y ha sido adoptado por pymes y micropymes es el Ágil. Y, sin embargo, metodologías ágiles y CMMI siempre han sido difíciles de unir³ [13].

Este artículo presenta como metodologías ágiles, software libre y CMMI pueden integrarse, y la experiencia de una micropyme, Unkasoft, en la certificación del nivel 2 de CMMI con dichos elementos.

2. Motivación para la certificación de UNKASOFT en el nivel 2 de CMMI

UNKASOFT es una empresa dedicada al desarrollo de tecnología y servicios dentro del mundo del marketing móvil, especialmente en aplicaciones y juegos patrocinados mediante

¹ http://www.inteco.es/Presse/Noticias_Calidad_SW_5/EmpresasSoftware

² <http://www.sei.cmu.edu/appraisal-program/profile/pdf/CMMI/2009MarCMMI.pdf>

³ <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06sr001.pdf>

publicidad (advergaming), de ámbito internacional y para grandes anunciantes (Pepsi, Nokia, Renault, Philips, etc.). Cuenta con un equipo técnico y creativo de 15 personas y ha invertido desde el año 2004 más de 2 millones de euros en desarrollar su tecnología y procesos. Unkasoft trabaja en un sector claramente indefinido e inmaduro como es el denominado *mobile marketing*, donde a día de hoy no existen estándares establecidos y algunas compañías alrededor del mundo intentan posicionarse antes de que el mercado madure. Debido a esto, la ejecución de los proyectos es altamente inestable, con cambios de rumbo continuos y sin un patrón claro a seguir.

A finales de 2007 se creó el grupo de mejora de procesos, con el objetivo de alcanzar, en el periodo de un año, el nivel de madurez 2 de CMMI, siendo este objetivo respaldado por la dirección de la empresa y ofreciendo los recursos necesarios para su consecución. Las principales motivaciones de la dirección fueron:

- Sello de calidad reconocido internacionalmente (especialmente en EEUU, importante, debido a una inminente entrada de la empresa en el mercado norte americano)
- Homogeneización de procesos en distintos departamentos.
- Fijación del conocimiento en la empresa.
- Establecer canales organizacionales de mejora y crecimiento.

El programa COMPETIC II⁴ supuso un claro empuje para afrontar las necesidades de implantación de un proceso ambicioso basado en CMMI-DEV v1.2 [14], y adecuarlas a los recursos limitados de una micropyme como Unkasoft.

3. Fases y recursos dedicados

El equipo de mejora de procesos se formó inicialmente por una persona a tiempo completo, la cual tenía amplia experiencia en prácticas avanzadas de ingeniería del software, procesos y metodologías ágiles, así como en implantación y adaptación de herramientas “open source” orientadas a la gestión de proyectos.

⁴ Proyecto liderado por Conetic (Confederación Española De Empresas De Tecnologías De La Información, Comunicaciones y Electrónica) y apoyado por el Plan Avanza del Ministerio de Industria, Turismo y Comercio, que se prevé certificar empresas asociadas en el modelo CMMI.

Dentro del marco del programa COMPETIC II, se siguió el plan de formación establecido e impartido por el “European Software Institute”, que pretendía dotar a los responsables de mejora de los conocimientos teóricos necesarios para la interpretación y adaptación del modelo CMMI a sus empresas. Este plan de formación se evaluó como muy recomendable, especialmente para las personas involucradas en la implantación que no tenían experiencia con modelos de calidad del software como CMMI o ISO 15504.

La fase de definición se desarrolló desde febrero hasta mayo de 2008, siendo necesario añadir un recurso al 50% a mitad de esta fase, para cumplir con los hitos acordados. También se contó con un consultor regional especializado en implantaciones de metodologías ágiles y CMMI, aportado por la empresa Códice Software. Durante esta fase se definió toda la arquitectura de la solución, planes de formación, roles y responsabilidades, canales de comunicación, procedimientos, políticas a distintos niveles, objetivos de negocio, preparación de las herramientas (incluida modificación, adaptación y ampliación de algunas de ellas), etc.

Desde Junio a Agosto de 2008 se continuó con la fase de pilotaje, donde se empezó a realizar la formación de los jefes de proyectos y auditores de calidad, recayendo posteriormente sobre ellos buena parte de la responsabilidad de implantación en sus respectivos proyectos. En esta fase se empleó aproximadamente un recurso a tiempo completo, además de un 25% de la jornada de los jefes de proyecto. Durante esta fase se realizaron cambios continuos en los procesos para adecuarlos a las necesidades de los proyectos pilotados.

Desde Septiembre hasta Noviembre de 2008 se extendió el uso de los procesos al resto de la compañía, siendo necesarios nuevos ajustes. Durante esta fase se contó con una persona al 50% y apoyo de la consultora Kybele Consulting, más especializada en áreas organizacionales como medición y análisis.

A partir de Diciembre de 2008 se entró en fase de SCAMPI A⁵, siendo ésta planificada en una revisión previa con el líder de la evaluación, una fase de *readiness review* (preparación de la auditoría) y una fase *on site*, estas dos últimas con el equipo de evaluación al completo. Tanto la *readiness review* como la fase *on site* tuvieron una

⁵ Standard CMMI Appraisal Method for Process Improvement, auditoría formal de CMMI que posibilita la certificación de un nivel de madurez.

duración de 1,5 semanas en las que participaron un evaluador líder, una persona de Unkasoft y dos evaluadores externos.

	2008											2009
	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Ene
Definición	1	1	1,5	1,5								
Pilotaje					1,5	1,5	1,5					
Uso								0,5	0,5	0,5		
SCAMPI A												

Tabla 1. Recursos.

Finalmente, en Enero de 2009, se completó la evaluación SCAMPI A, alcanzándose el nivel de madurez 2 y entrando en fase de mantenimiento, donde se está dedicando una persona a tiempo parcial para el mantenimiento y evolución de los procesos.

En la **Tabla 1** se puede ver un resumen esquemático de los recursos necesarios y su distribución a lo largo del proyecto.

4. Metodologías y herramientas

Desde el inicio de la definición, se tenía claro que era necesario adaptar un proceso clásico, a uno dentro de la corriente ágil, basando la carga del proceso en herramientas y metodologías ágiles.

Este espíritu ágil no surge de la dirección hacia la estructura, sino de forma de inversa y espontáneamente años atrás: los propios programadores y técnicos propusieron dichas mejoras en sus procesos, siendo estas escuchadas y valoradas por la dirección. La implantación de CMMI ha significado una oportunidad clara para poner en marcha todas estas mejoras propuestas a lo largo del tiempo y darles una forma más institucional.

Para la solución técnica, se han utilizado las siguientes herramientas, introduciéndose algunas de ellas durante el proceso de implantación, y siendo otras adaptadas o extendidas, ya que venían utilizándose años atrás:

1. **Subversión:** como soporte básico para la gestión de la configuración. Ya se utilizaba desde hacía años de forma sistemática, por lo que toda la infraestructura y *know-how* existentes fueron reaprovechados al completo.

2. **Trac:** centralizando toda la gestión de proyectos, estimación, trazabilidad completa, análisis de riesgos, actividades de medición, auditoría, etc. Se llevaba utilizando en el área de plataforma durante el año 2007, y durante la implantación de CMMI se extendió a toda la organización, realizándose multitud de adaptaciones del código original.
3. **Histórico de estimaciones:** sobre la base de Trac, se ha desarrollado un sistema de histórico de estimaciones alimentado en tiempo real. Este sistema recibe datos de todos los proyectos ejecutados, y ofrece estadísticas y valores de estimación ajustados para siguientes proyectos.
4. **Wiki:** integrado dentro del propio Trac, se utiliza para la gestión documental a nivel global: procedimientos, políticas, material de formación, planes de proyecto, documentación técnica, etc.
5. **Maven:** sistema de construcción, gestión de versiones y dependencias para desarrollos Java. Gracias a él es posible mantener un estricto control de las dependencias y de las versiones generadas de cada sistema/librería.
6. **Hudson:** como servidor de integración continua, realizando la construcción del código, ejecución de pruebas en distintos niveles (unitarias, integración, rendimiento, etc.), análisis estático del código y recogida de métricas.
7. **Herramientas ofimáticas:** para generar informes gerenciales y comunicación de resultados hacia el exterior. Hojas de cálculo puntuales de cálculo estadístico y análisis de tendencias.

Área de proceso CMMI / Herramienta utilizada	Svn	Trac	Wiki	Hudson	Maven	Word/ Excel
PP+PMC (Project Planning + Project Monitoring Control)	X	X	X			X
CM (Configuration Management)	X	X	X	X	X	
REQM (Requirements Management)	X	X	X	X		X
MA (Measurement and Análisis)	X	X	X	X		X
PPQA (Process and Product Quality Assurance)		X	X	X		

Tabla 2. Uso de herramientas de software libre de apoyo para la implantación de las distintas áreas de proceso del modelo CMMI-DEV v1.2 en Unkasoft.

	Svn	Trac	Wiki	Hudson	Maven	Word/ Excel
GP 2.1 Establish and Organizational Policy			X			
GP 2.2 Plan the Process	X	X	X			
GP 2.3 Provide Resources		X	X			X
GP 2.4 Assign Responsibility		X	X			
GP 2.5 Train People			X			
GP 2.6 Manage Configurations	X	X	X	X	X	
GP 2.7 Identify and Involve Relevant Stakeholders		X	X			
GP 2.8 Monitor and Control the Process		X	X			X
GP 2.9 Objectively Evaluate Adherente		X	X			
GP 2.10 Review Status with Higher Level Management		X	X			X

Tabla 3. Uso de herramientas de software libre de apoyo para la implantación de las prácticas genéricas del modelo CMMI-DEV v1.2 en Unkasoft.

En las **Tablas 2 y 3** se puede apreciar cómo se han utilizado las distintas herramientas para cubrir cada una de las áreas de proceso y prácticas genéricas del modelo CMMI

5. Recomendaciones y lecciones aprendidas

Una vez superado y auditado el proceso de implementación de CMMI, se pueden extraer las siguientes recomendaciones:

- El proceso de definición e implantación de los procesos debe ser iterativo y, a ser posible, incremental. En nuestro caso se ejecutaron tres iteraciones, refinándose en cada una los procedimientos, responsabilidades, actividades, herramientas, etc.
- Para una micropyme como Unkasoft, la única forma de alcanzar el objetivo impuesto por el nivel 2 de CMMI se basa en centrar el proceso en el uso de métodos y herramientas ágiles, en nuestro caso Trac. La automatización del mayor número de procesos fue fundamental.
- El uso de la metodología ágil SCRUM ha aportado flexibilidad al proceso, y por lo tanto una clara ventaja sobre la implantación de metodologías más tradicionales.

- Contar con equipos multidisciplinares tanto en los proyectos como en el grupo de mejora es la única garantía de éxito. Es necesaria implicación, flexibilidad y motivación por parte de todos, muy especialmente en el equipo de mejora y la jefatura de proyectos.
- Para afrontar el SCAMPI A, ser imaginativo con los artefactos (evidencias) y tenerlas en cuenta desde el principio. Reutilizar todas las posibles, y fomentar el uso de evidencias generadas por automatizaciones.
- El área de medición ha supuesto un ejercicio de análisis del negocio y ha marcado las bases para establecer objetivos a distintos niveles así como mecanismos de reporte. Ha permitido crear una estructura organizativa escalable.
- Habría sido posible optimizar los tiempos de implantación si se hubiera contado con un equipo de mejora con experiencia en implantaciones CMMI bajo el paradigma ágil.

6. Conclusiones

En este caso de estudio, se ha demostrado que la implantación de un proceso basado en CMMI utilizando metodologías ágiles es perfectamente viable, y en caso de micropymes, resulta una opción más que recomendable, aunque requiera de unos requisitos especiales.

El proceso de implantación ha significado un esfuerzo importante a distintos niveles, con el objetivo de conseguir un proceso flexible, ágil y escalable. Y ha resultado evidente que el trabajo no concluye al finalizar el SCAMPI A, sino que es necesario continuar con un esfuerzo de reinterpretación del modelo, mejora de las automatizaciones, simplificación del proceso, etc.

El principal factor de éxito han sido un equipo motivado, multidisciplinar y con amplia experiencia, así como un conjunto de herramientas adaptadas a las necesidades del proceso y las singularidades de la empresa.

Agradecimientos

A todo el equipo de Unkasoft, por la enorme dedicación y esfuerzo que realizan todos los días y especialmente durante la implantación de los procesos.

A todo el equipo de Kybele Consulting, por su esfuerzo en incrementar cada día la calidad del software en España.

Referencias

- [1] Piattini M, Garzás J. (eds). *Fábricas de software: Experiencias, tecnologías y organización*. Ra-ma, 2007.
- [2] Hurtado J, Pino, F. y Vidal, J.. *Software Process Improvement Integral Model: Agile SPI. Technical Report SIMEP-SW-O&A-RT-6-V1.0. 2005*. Universidad del Cauca - Colciencias, 2006.
- [3] SEI, *Process Maturity Profile. CMMI v1.1, SCAMPI v1.1, Class A Appraisal Results. 2006 Mid-Year Update*. Software Engineering Institute. 2006.
- [4] ISO, *ISO/IEC 15504-2:2003/Cor.1:2004(E). Information technology - Process assessment - Part 2: Performing an assessment*. International Organization for Standardization, 2004.
- [5] INTECO, *Estudio sobre la certificación de la calidad como medio para impulsar la industria de desarrollo del software en España*, INTECO, 2008 (http://www.inteco.es/Calidad_del_Software/estudios_e_indicadores/publicaciones/calidad_sw_estudios_e_informes/Calidad_software_32, consultado por última vez en Abril de 2009).
- [6] Staples M., Niazi M., Jeffery R., Abrahams A., Byatt P. y Murphy R. "An exploratory study of why organizations do not adopt CMMI", *Journal of Systems and Software*, vol.80, nº 6, pp.883-895, 2007.
- [7] Hareton L. y Terence Y., "A Process Framework for Small Projects", *Software Process Improvement and Practice*, vol. 6, nº 2, pp. 67-82, 2001.
- [8] Saiedian H, Carr N., "Characterizing a software process maturity model for small organizations", *ACM SIGICE Bulletin*, vol. 23, nº1, p.2-11, 1997.
- [9] Fayad M.E., Laitinen M. y Ward R.P., "Software Engineering in the Small", *Communications of the ACM*, vol. 43, nº3, pp. 115-118, 2000.
- [10] Zahran, S., *Software Process Improvement: Practical Guidelines for Business Success*. Addison-Wesley, 1998.

- [11] Dyba, T., "An Empirical Investigation of the Key Factors for Success in Software Process Improvement" *IEEE Transactions on Software Engineering*, vol 31, n°5, pp.410-424, 2005.
- [12] Calvo-Manzano J.A, Garzías J., Piattini M., Pino F.J., Salillas J. y Sánchez J.L. "Perfiles del ciclo de vida del software para pequeñas empresas: Los informes técnicos de ISO/IEC 29110", REICIS, vol. 4, n ° 3, pp. 96-108.
- [13] Sutherland J., Jakobsen, C.R. y Johnson, K., "Scrum and CMMI Level 5: The Magic Potion for Code Warriors", Agile Conference 2007, p.1-6.
- [14] Chrissis M.B., Konrad M. y Shrum S. *CMMI: Guidelines for Process Integration and Product Improvement*, Addison-Wesley Professional, 2006.

Metodología para la evaluación de la calidad en los modelos UML

Moisés Rodríguez, Javier Verdugo, Ramón Coloma
Alarcos Quality Center, s.l.
{moises.rodriguez, javier.verdugo, ramon.coloma}@alarcosqualitycenter.com

Marcela Genero, Mario Piattini
Departamento de Tecnologías y Sistemas de Información
Universidad de Castilla-La Mancha
{marcela.genero, mario.piattini}@uclm.es

Resumen

Este artículo presenta una metodología para la evaluación de la calidad de modelos UML, formada por un conjunto estructurado de procesos, orientada a la relación con el cliente y a la externalización de la evaluación de la calidad, y además pensada para ser una metodología fácilmente adaptable.

Palabras clave: evaluación de calidad, calidad en modelos UML.

Methodology for the evaluation of quality in UML models

Abstract

This article presents a methodology for evaluate the quality of UML models. This methodology consists of a structured set of processes, it is oriented to the relationship with the customer and to the outsourcing of quality evaluation, and it is easily adaptable.

Key words: quality evaluation, quality in UML models.

Rodríguez,M., Verdugo, j., Coloma,R., Genero, M. y Piattini,M. "Metodología para la evaluación de la calidad en los modelos UML", REICIS, vol. 6, no.1, 2010, pp.17-36. Recibido: 6-11-2009; revisado: 4-1-2010; aceptado:12-3-2010

1. Introducción

Si bien la calidad del software es un campo maduro en el que existen modelos de calidad (ISO 9126, ISO 25000, IEEE 1061-1998, etc.), modelos de madurez (CMMI, ISO 15504, etc.), estándares relacionados con la verificación y validación de software (IEEE 1012, IEEE 1028, etc.), herramientas para el cálculo de métricas, etc., que llevan utilizándose en

la industria hace más de 20 años, no ocurre lo mismo en el campo de la calidad de las especificaciones de software y más concretamente, en la calidad de los modelos UML.

Tal y como se refleja en la revisión sistemática [1], que contempla las propuestas existentes en la literatura desde 1998 hasta finales de 2007 sobre la calidad en el modelado con UML, este tema es de suma actualidad y relevancia, pero carece todavía de suficiente madurez.

Por todo ello, se ha elaborado la presente metodología de evaluación de la calidad en los modelos UML, enmarcada dentro del proyecto EVVE “Entorno para la Verificación y Validación de Especificaciones software” (en adelante metodología EVVE).

1.1. Revisión de estándares

Como punto de partida para la elaboración de la metodología EVVE, se ha realizado una revisión de los principales estándares, normas y metodologías existentes en la actualidad que, en mayor o menor medida, se encuentran relacionados con la evaluación del software. A continuación se identifican y se describen brevemente dichos estándares (ver tabla 1).

Estándar/Metodología	Descripción
IEEE 1012-1998. Standard for Software Verification and Validation [2]	Estándar que define los procesos de verificación y validación (V&V) del software
IEEE 1028-2008. Standard for Software Reviews [3]	Estándar que define la manera de realizar revisiones sistemáticas para la adquisición, suministro, desarrollo y operación de software
IEEE Std 1061-1998. Standard for a Software Quality Metrics Methodology [4]	Estándar que define una metodología para identificar, implementar, analizar y validar métricas de calidad del producto y el proceso
CMMI-DEV (Capability Maturity Model Integration for Development) [5]	Modelo para la mejora para los procesos de desarrollo de productos y servicios
CMMI-ACQ (Capability Maturity Model Integration for Acquisition) [6]	Modelo para la mejora de procesos en la gestión de la cadena de suministro, adquisición y contratación externa
CMMI-SVC (Capability Maturity Model Integration for Services) [7]	Modelo para la mejora de procesos, centrado en las actividades que requieren gestionar, establecer y entregar servicios
ISO/IEC 9126. Software engineering-Product quality [8]	Norma sobre la calidad del producto software

Estándar/Metodología	Descripción
ISO/IEC 12207. Systems and software engineering-Software life cycle processes [9]	Norma que define los procesos del ciclo de vida del software
ISO/IEC 14598. Information technology-Software product evaluation [10]	Norma para la evaluación del software
ISO/IEC 15504. Software engineering-Process assessment [11]	Norma para la evaluación de procesos software
ISO/IEC 25000. Software engineering-Software product Quality Requirements and Evaluation (SQuaRE) [12-19]	Familia de normas relacionadas con los modelos, mediciones y evaluaciones de la calidad, que supone la evolución de las normas ISO/IEC 9126 e ISO/IEC 14598
ISO/IEC 29119. Software Testing [20-22]	Familia de normas relacionadas con las pruebas del software, que cubren el ciclo de vida completo
TMAP Next [23]	Metodología de pruebas del software
PSM: Practical Software and Systems Measurement [24]	Guía para el proceso de medición del software
EMISQ [25]	Metodología para la evaluación de la calidad interna del software
LifeCycleQM [26]	Framework para la selección y personalización de las actividades de aseguramiento de la calidad

Tabla 1. Estándares, normas y metodologías relacionadas con la evaluación de la calidad software.

A partir de dicha revisión, se ha elaborado la metodología EVVE, analizando las características clave de cada uno de ellos y seleccionando, adaptando e integrando los aspectos más relevantes. Además, se ha integrado todo con los resultados de las investigaciones realizadas durante el proyecto relativos a las técnicas y métricas de evaluación de la calidad para modelos software en UML. A continuación se exponen brevemente las principales ideas que se han tomado de cada una de las referencias utilizadas para la elaboración de la metodología:

- IEEE: Se han revisado los principales estándares IEEE [2, 27, 28] relacionados con la evaluación y medición de la calidad del software. Estos estándares han aportado a la metodología EVVE la visión de cómo realizar las actividades de la

verificación y validación del software, así como una base objetiva en el apartado de definición de métricas. Sin embargo la metodología EVVE aporta una visión más especializada, centrada en la evaluación de la calidad de los modelos UML.

- CMMI: Tal y como se puede observar en el último informe publicado por el SEI en el año 2009 [29], el número total de evaluaciones realizadas bajo este modelo ha aumentado significativamente, lo que nos da una idea de la importancia e influencia de este modelo. Por ello, se han revisado las “Áreas de Proceso” (Process Areas) que el modelo describe relacionadas con la evaluación y la calidad del software, así como los entregables (productos de trabajo) que el modelo identifica, con el objetivo de seleccionar y adaptar aquellos que mejor se relacionan con los requisitos de la metodología EVVE.
- ISO: Dada la importancia de la ISO como organización de estandarización y la aceptación de sus normas como guías a seguir, se han estudiado las principales normas relacionadas con la calidad del producto, el ciclo de vida de desarrollo y los procesos de evaluación y medición de artefactos que se generan en un proyecto software. Dentro de estas normas tiene especial importancia, para la metodología EVVE, la familia de normas ISO 25000 que engloba los modelos, mediciones y procesos para realizar la evaluación de la calidad del producto software. El modelo de calidad genérico en el que se apoya la metodología EVVE, está basado en las características y subcaracterísticas de calidad detalladas en la ISO 25010 [30].
- TMAP: Es una de las metodologías de pruebas del software más conocidas en el ámbito profesional. La metodología EVVE ha tenido en cuenta como TMAP relaciona el conjunto de procesos con las técnicas y herramientas que permiten su implementación, adaptando la visión que TMAP aporta para las pruebas del software a la evaluación de la calidad de los modelos en la que se centra la metodología EVVE.
- PSM: Es un proceso de medición que permite dirigir los objetivos técnicos y de negocio de una organización, y recoge las mejores prácticas utilizadas por los profesionales de la medición dentro de las comunidades del software, la adquisición de sistemas y la ingeniería. PSM define un conjunto de actividades

que permiten evaluar el propio proceso de medición y en la metodología EVVE se han tenido en cuenta a la hora de definir un proceso de soporte al proceso de evaluación, el proceso de gestión de la evaluación, cuyo objetivo es detallar las actividades a realizar durante el proyecto que son necesarias para controlar toda la documentación generada y la información sobre el propio proceso de evaluación, de manera que se pueda medir la calidad del propio proceso y mejorarlo en sucesivas aplicaciones de la metodología.

- **EMISQ:** Es una metodología para la evaluación sistemática de la calidad interna del software, basada en la norma ISO 14598. Para desarrollar la metodología EVVE se ha tenido en cuenta EMISQ como base para definir una metodología basada en un estándar ISO y a partir de ahí crear un Framework apoyado por una herramienta que permita llevar a cabo la evaluación. A diferencia de EMISQ, en la metodología EVVE la norma ISO de base será la familia ISO 25000.
- **LifeCycleQM:** Es un proyecto del Instituto Fraunhofer (Alemania) que ha desarrollado una guía para la selección y personalización de las actividades de aseguramiento de la calidad que mejor se adaptan a las necesidades de la organización. La principal desventaja detectada es que está orientado principalmente a empresas desarrolladoras de software. Por ello, la metodología EVVE se ha pensado tanto para organizaciones y fábricas de software que desarrollan sus propios modelos, como para empresas que adquieren productos software y desean asegurar la calidad de sus modelos.

2. Características y descripción de la metodología

Las principales características de la metodología EVVE, se pueden resumir en los siguientes principios básicos:

- Está formada por un conjunto estructurado de procesos.
- Está orientada a la relación con el cliente y a la externalización de la evaluación de calidad.
- Está pensada para ser una metodología fácilmente adaptable.
- Está soportada por un conjunto de técnicas y herramientas.

La metodología EVVE proporciona un marco de trabajo donde se identifica claramente el qué, cuándo, y el quién, de cada una de las fases y actividades de los procesos, así como la secuencia de pasos que se debe seguir a la hora de llevar a cabo la evaluación.

Además, está orientada a la relación con el cliente de manera que, en distintos puntos del proceso de evaluación, el cliente se encuentra involucrado en la toma de decisiones. Y por otro lado, la metodología contempla el modo de trabajo externalizado, de manera que para realizar la evaluación no sea necesario encontrarse en las instalaciones del cliente, sino que se pueda planificar, diseñar y realizar la evaluación externamente, poniéndose en contacto con el cliente en los momentos puntuales en los que sea necesario.

De igual manera, la metodología de evaluación está pensada para poder ser adaptada a las distintas necesidades del cliente, existiendo unos catálogos de técnicas de evaluación que determinan el nivel y profundidad con el que se desea realizar la evaluación, las métricas de calidad que se obtendrán y las herramientas de evaluación que se utilizarán.

Por último, la metodología EVVE se encuentra soportada por un conjunto de técnicas de evaluación y herramientas de soporte, que agilizan la evolución de las fases de la metodología y permiten diseñar y realizar la evaluación de manera semiautomática. Actualmente, se está utilizando un conjunto de herramientas independientes para realizar dicha evaluación, entre las que podemos destacar (PMD, Checkstyle, NDepend, FXCop, JavaNCSS, etc). Si bien, se está trabajando para integrar todas las herramientas en un entorno común que permita la evaluación de forma automática.

2.1. Objetivo de la metodología

El principal objetivo de la metodología EVVE es definir un marco de trabajo que permita determinar los procesos necesarios para llevar a cabo la evaluación de los modelos UML

del software, así como facilitar la comunicación entre el cliente (patrocinador de la evaluación) y el equipo de evaluación.

2.2. Ámbito de aplicación

La metodología de evaluación EVVE se puede aplicar a cualquier modelo de análisis y diseño, siempre que esté especificado bajo la notación UML. Si bien en un principio está pensada para ser utilizada con los siguientes tipos de diagramas:

- Diagramas de casos de uso.
- Diagramas de clases.
- Diagramas de transición de estados.

Por otro lado, la metodología, está pensada para ser utilizada independientemente del modelo de ciclo de vida de desarrollo seguido por la empresa cliente. De manera que no existe un proceso, fase, actividad o tarea del ciclo de vida a partir del cual se pueda empezar a aplicar la metodología de evaluación. Por ejemplo se podría aplicar la metodología para evaluar la calidad de los modelos UML finales, que se generan como entregables de un proyecto, o aplicar la metodología para evaluar la calidad después de cada iteración en el desarrollo de un software que siga un ciclo de vida iterativa e incremental como es el Proceso Unificado.

3. Catálogo de roles

A continuación se detallan los roles y agrupaciones de estos que se identifican a lo largo del proceso de evaluación, así como las principales tareas y responsabilidades de cada uno de ellos.

En la siguiente figura (ver figura 1), se pueden observar los roles involucrados en el proceso de evaluación según la metodología EVVE, así como la relación que existe entre ellos. Tal y como se aprecia en la figura, la comunicación entre las empresas (cliente y evaluadora) es realizada a través del patrocinador y del evaluador jefe. Éstos a su vez serán los que se comuniquen directamente con sus equipos internos.

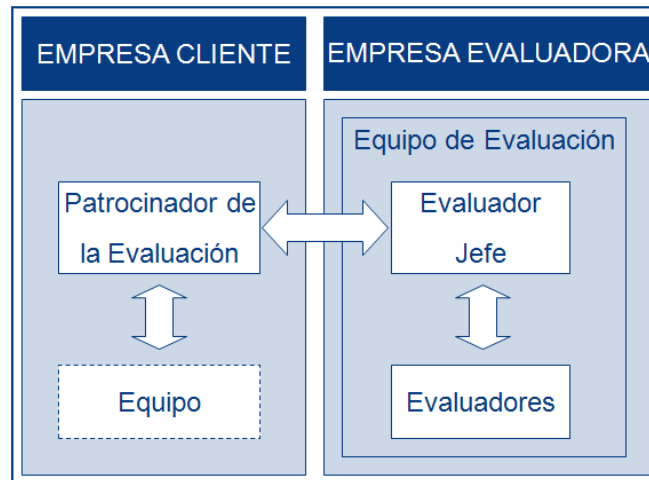


Figura 1. Roles y relaciones en el proceso de evaluación.

- **Empresa cliente.** Representa a la organización que ha expresado su intención y/o necesidad de contratar los servicios de evaluación de la calidad de los modelos UML (bien desarrollados por ella o por otra empresa).
- **Patrocinador de la evaluación.** El patrocinador de la evaluación es el representante de la empresa cliente que se pone en contacto con la empresa evaluadora para expresar la necesidad de evaluación de la calidad de sus modelos UML (normalmente suele ser el jefe de proyecto o el responsable del departamento de calidad). En algunos casos, el patrocinador de la evaluación podrá disponer de un equipo de trabajo interno formado por personal propio de la empresa (ver figura 1, rectángulo punteado).
- **Equipo de la empresa cliente.** Sirve de apoyo al patrocinador de la evaluación. Su principal responsabilidad consiste en analizar los resultados reportados por la empresa evaluadora para detectar defectos o inconsistencias. La existencia de este equipo no es necesaria siempre, depende de la envergadura del proyecto a evaluar.
- **Empresa evaluadora.** Representa la organización que ha sido contratada para llevar a cabo el proceso de evaluación. La empresa evaluadora necesitará disponer de un evaluador jefe y uno o varios evaluadores que formaran el equipo de trabajo que llevará a cabo el proceso de evaluación.

- **Evaluador jefe.** Es el representante de la empresa evaluadora y líder del equipo de evaluación, cuyo objetivo es asegurar el correcto desarrollo del proceso de evaluación. Este rol debe ser ocupado por un profesional con conocimientos en el análisis y diseño con UML y con experiencia en la práctica de las principales normas y estándares sobre evaluación del producto y procesos software.
- **Evaluador.** Está encargado de realizar las actividades y tareas propias del proceso de evaluación. Este rol deberá ser ocupado por una persona con conocimientos de modelado UML y de la metodología de evaluación. Las principales responsabilidades son, realizar las actividades de evaluación asignadas por el evaluador jefe y recoger información del propio proceso de evaluación.
- **Equipo de evaluación.** Representa el equipo de la empresa evaluadora, formado por el evaluador jefe y un conjunto de uno o varios evaluadores de calidad (según los requisitos del proyecto). Su principal objetivo es cumplir con los requisitos de evaluación acordados con el patrocinador dentro de los tiempos planificados.

4. Catálogo de elementos

A continuación se detallan todos los elementos (documentos, informes, catálogos, artefactos, etc.) que se identifican a lo largo de las actividades de evaluación. Estos elementos podrán ser de entrada y/o salida para las actividades del proceso de evaluación, siendo además algunos de ellos entregables finales del proyecto de evaluación.

- **Información de contacto.** Documento donde la empresa evaluadora recoge los datos de contacto de la empresa cliente. Como mínimo debe presentar la siguiente información:
 1. Nombre de la empresa.
 2. Dirección.
 3. Teléfono/Fax.
 4. Página Web.
 5. Persona de contacto (posible patrocinador de la evaluación)
 6. Email de contacto.
- **Contrato de evaluación.** Este documento representa el primer entregable (documento de salida, se entrega a la empresa cliente) del proyecto de evaluación

y será generado como resultado de la fase 1 (planificación). En él se recogen las condiciones bajo las que se realizará el proyecto de evaluación, es de carácter público para la empresa evaluadora y para la empresa cliente. Está dividido en dos partes, que bien podrán estar incluidas en el mismo documento, o en dos documentos separados:

1. Propuesta técnica de evaluación: contiene aspectos técnicos del contrato de evaluación tales como el objetivo de la evaluación, el plan de trabajo, el equipo de trabajo, el calendario y el esfuerzo necesario.
 2. Propuesta económica de evaluación: contiene aspectos económicos del contrato de evaluación tales como el precio, el modo de facturación, las cláusulas, las condiciones generales, etc.
- **Modelo de calidad genérico.** Es un documento propiedad de la empresa evaluadora y contiene el conjunto de características, subcaracterísticas y atributos de calidad que se pueden evaluar para cada artefacto software. Como se ha indicado anteriormente, estas características y subcaracterísticas de calidad se encuentran detalladas en la ISO 25010 [30].
 - **Modelo de calidad específico.** Es un documento que solo contiene el conjunto de características, subcaracterísticas y atributos de calidad que la empresa cliente ha seleccionado para el proyecto de evaluación. Este documento es de carácter público para la empresa evaluadora y la empresa cliente, está integrado dentro de la especificación de la evaluación que representa el tercer entregable del proceso de evaluación.
 - **Catálogo de herramientas de evaluación.** Es un documento propiedad de la empresa evaluadora y que permite identificar las herramientas necesarias y su modo de utilización, en función de las técnicas de evaluación que se vayan a utilizar para evaluar los modelos UML.
 - **Plan de evaluación.** Este documento público para ambas partes, y representa el segundo entregable del proyecto de evaluación. Una primera versión de este documento se desarrolla previamente al inicio del proyecto de evaluación, incluida en la propuesta técnica del contrato de evaluación. Esta primera versión es refinada por ambas partes tras la aceptación del contrato, generándose así la

versión definitiva del plan de evaluación, que guiará todos los procesos y actividades realizados durante el proyecto. Este documento es generado como resultado de la fase 1 (planificación) del proceso de evaluación, sin embargo estará sujeto a cambios durante todo el proyecto de evaluación y en él se deberán registrar los retrasos y re-planificaciones.

- **Conjunto de artefactos a evaluar.** Entendiendo por artefactos los modelos UML que la empresa cliente quiere validar, así como toda la documentación que sea necesaria para poder comprender correctamente la naturaleza y contenido de dichos modelos. Entre otras, es responsabilidad del patrocinador que los artefactos a evaluar estén disponibles en las fechas planificadas. Y responsabilidad del evaluador jefe asegurar que los artefactos entregados para ser evaluados cumplen con los requisitos impuestos por las técnicas y herramientas que se van a utilizar en el proceso de evaluación.
- **Especificación de la evaluación.** Este artefacto representa el tercer entregable del proyecto de evaluación y como tal es público para ambas empresas. La especificación de la evaluación será generada como resultado de la fase 2 (especificación) del proceso de evaluación y estará formado por uno o varios documentos donde se recoja, una vez analizado el conjunto de artefactos a analizar, el modelo de calidad específico que se va a analizar, qué técnicas se van a utilizar, qué métricas se esperan obtener, qué herramientas se van a manejar, etc. En definitiva un conjunto de características a evaluar por cada uno de los modelos UML, las técnicas y herramientas que se utilizarán y el listado de métricas e indicadores que se esperan obtener. Esta especificación posteriormente será incluida como apartado dentro del informe final de evaluación, conforme lo establece la norma ISO/IEC 14598 [10].
- **Informe de evaluación.** Este artefacto representa el cuarto entregable del proyecto de evaluación y como tal es público para ambas empresas. El informe de evaluación será generado como resultado de la fase 4 (conclusión) y estará formado por un documento donde se recogen, todos los resultados obtenidos mediante las técnicas de evaluación, las conclusiones obtenidas del análisis de dichos resultados y un conjunto de recomendaciones y acciones futuras para la

mejora de los modelos UML, además de la especificación de la evaluación. El informe va acompañado de una presentación de los resultados que va dirigida tanto al patrocinador de la evaluación como a los directivos de la empresa cliente y al equipo interno si lo hubiese. Una característica importante del informe es la retroalimentación que se produce posteriormente a la presentación de los resultados y en cuyo caso la empresa cliente reporta a la empresa evaluadora mediante el documento petición de modificación.

- **Petición de modificación.** Este artefacto representa el quinto entregable del proceso de evaluación y como tal es público para ambas empresas. Este documento podrá ser generado como resultado de la fase 4 (conclusión) y representa el documento mediante el cual la empresa cliente, manifiesta sus opiniones y posibles cambios respecto a los resultados presentados en el informe de evaluación. La empresa evaluadora recibirá el documento que evaluará, revisará y actualizará el informe de evaluación de acuerdo a los cambios sugeridos. El informe modificado será de nuevo entregado a la empresa cliente quien podrá solicitar una nueva reunión o reportar una nueva petición de modificación, actuándose de la misma manera que la primera vez. El resultado de esta petición no es más que un refinamiento del informe de evaluación así como la mejora del proceso de evaluación.
- **Información interna de evaluación.** Además de los propios entregables y resultados de la evaluación, durante la evaluación se genera un conjunto de documentación relacionada con el propio proceso. Esta información principalmente son informes del proceso de gestión y monitorización, informes de métricas de la propia evaluación (progreso, retrasos, costes/beneficios, etc.), informes de cambios y ajustes realizados durante la evaluación, informes de defectos y adaptaciones realizadas en la infraestructura de evaluación, etc. Posteriormente esta información será utilizada por los procesos **Gestión de la Evaluación** y **Gestión de la Infraestructura** de la presente metodología, para realizar una mejora continua del proceso de evaluación.

5. Procesos de la metodología de evaluación

La metodología de evaluación EVVE se encuentra compuesta por tres procesos principales (ver figura 2).



Figura 2. Procesos y fases de la metodología de evaluación EVVE.

5.1. Proceso de evaluación

Este proceso representa el esqueleto de la metodología de evaluación ya que contiene las fases de planificación, especificación, ejecución y conclusión del proyecto de evaluación. La ejecución de este proceso está directamente relacionada con los dos primeros objetivos de la metodología EVVE.

El proceso de evaluación⁶ se descompone en cuatro fases:

5.1.1. Fase 1: Planificación

El objetivo de esta fase es la elaboración del contrato y la obtención de un plan de evaluación de los modelos UML.

Esta fase está formada por cuatro actividades:

- Contratación.
- Arranque del proyecto.

⁶ Tanto las fases del proceso de evaluación como cada una de las actividades que componen cada fase pueden adaptarse, en cuanto a la complejidad del contenido y al orden de las mismas, pudiéndose realizar evaluaciones más ligeras o completas según las necesidades y características de la empresa cliente.

- Planificación detallada.
- Consolidación del plan.

5.1.2. Fase 2: Especificación

El objetivo de esta fase es definir el alcance de la evaluación, determinando el conjunto de características que se van a evaluar para cada uno de los modelos UML (artefactos), las técnicas y herramientas que se van a utilizar para realizar la evaluación, los niveles de evaluación (bajo, medio o alto) que se van a aplicar a cada artefacto y el listado de indicadores y métricas que se esperan obtener.

Esta fase está formada por cuatro actividades:

- Obtención y análisis de artefactos a evaluar.
- Selección del modelo de calidad y las técnicas.
- Planificación interna de la evaluación.
- Verificación de la especificación.

5.1.3. Fase 3: Ejecución

El objetivo de la fase de ejecución es la aplicación de las técnicas de evaluación y el lanzamiento de las herramientas (utilizando como entrada los artefactos a evaluar) para obtener los resultados iniciales (métricas de más bajo nivel) sobre la calidad de los modelos UML.

Una vez obtenidos los resultados iniciales, es necesario asegurar su veracidad (pueden existir defectos en los resultados, como los conocidos falsos positivos) y realizar su almacenamiento de una manera unificada que permita su posterior explotación.

Esta fase está formada por tres actividades:

- Aplicación de técnicas de evaluación.
- Análisis de la ejecución.
- Unificación de resultados.

5.1.4. Fase 4: Conclusión

El objetivo de la fase de conclusión es la elaboración del informe de evaluación y la presentación de los resultados tanto al patrocinador como al resto de personas involucradas dentro de la empresa cliente.

Esta fase está formada por tres actividades:

- Elaboración del informe de evaluación.
- Presentación de resultados.
- Corrección del informe y finalización de la evaluación.

5.2. Proceso de gestión de la evaluación

Es el proceso de soporte al proceso de evaluación, que permite controlar, evaluar y mejorar el propio proceso de evaluación. La ejecución de este proceso está directamente relacionada con el segundo y tercer objetivo de la metodología EVVE.

Este proceso está formado únicamente por una fase, denominada de igual manera que el proceso. Esta fase a su vez se descompone en tres actividades⁷ bien diferenciadas:

- Monitorización.
- Documentación.
- Ajuste y Evolución.

5.3. Proceso de gestión de la infraestructura

Es el proceso de soporte al proceso de evaluación, que permite gestionar todo lo relacionado con la infraestructura necesaria para llevar a cabo el proceso de evaluación (técnicas y herramientas de evaluación). La ejecución de este proceso está directamente relacionada con el tercer y cuarto objetivo de la metodología EVVE.

Este proceso está formado únicamente por una fase, denominada de igual manera que el proceso. Esta fase a su vez se descompone en tres actividades² bien diferenciadas:

- Especificación de la infraestructura.
- Mantenimiento de la infraestructura.
- Adaptación y transferencia de la infraestructura.

6. Implementación de la metodología con SPEM y EPFC

SPEM 2.0 (Software & System Process Engineering Metamodel) es un metamodelo de OMG considerado como el estándar de facto en la industria para la representación de modelos de procesos de ingeniería del software e ingeniería de sistemas.

⁷ Cada una de las actividades que componen los procesos de gestión de la evaluación y gestión de la infraestructura, pueden adaptarse (en cuanto a la complejidad del contenido y al orden de las mismas), en función de las características de la empresa cliente y del proyecto de evaluación.

Por otro lado, dentro de la plataforma abierta Eclipse, se ha desarrollado un editor de SPEM 2 llamado EPFC (Eclipse Process Framework Composer), que permite definir, gestionar y reutilizar un repositorio de fragmentos de métodos y procesos. De modo que con EPFC se pueden crear implementaciones en formato SPEM 2 de cualquier método, proceso o metodología de ingeniería del software.

La idea central de SPEM 2 para representar procesos está basada en tres elementos básicos: rol, producto de trabajo y tarea[31].

- Las tareas representan el esfuerzo a hacer.
- Los roles representan quién lo hace.
- Los productos de trabajo representan las entradas que se utilizan en las tareas y las salidas que se producen.

6.1. Ventajas de SPEM y EPFC

Las principales razones por las que se han seleccionado SPEM y EPFC para la implementación práctica de la metodología EVVE, frente a otras alternativas son las que se describen a continuación [31, 32]:

- El formato en el que se maneja la información en las metodologías suelen ser documentos de texto natural.
- La creación, revisión, reutilización, adaptación y generación de documentación para las metodologías suele ser un proceso puramente manual.
- EPFC permite crear metodologías y procesos con un rico contenido (texto con formato, imágenes, elementos multimedia, etc.).
- EPFC proporciona una estructura de gestión común para todo el contenido metodológico de una organización.
- EPFC proporciona un entendimiento claro y visual de cómo se relacionan las diferentes tareas de una metodología.
- EPFC permite seleccionar, combinar, adaptar y ensamblar de forma rápida procesos a partir del contenido de método creado.
- EPFC permite publicar la documentación de los procesos y metodologías modelados en formato para la web.

- EPFC reduce el coste de reutilización de contenidos de método en una organización.

6.2. Ejemplo de diagrama de actividad generado por EPFC

A continuación (ver figura 3), se presenta el flujo de trabajo generado por EPFC para las tareas que se realizan en la actividad **Obtención y Análisis de Artefactos a Evaluar**, actividad que pertenece a la fase 2 (Especificación) del proceso de evaluación especificado anteriormente.

Tal y como se puede ver en la figura 3, esta actividad está compuesta por 3 tareas las cuales se deben realizar en orden secuencial. Además para cada una de las tareas se muestra en el margen superior izquierdo la abreviatura del rol implicado en su realización.

EPFC permite la generación automática de este tipo de diagramas a partir de la información de la metodología introducida en el sistema. Además, desde este tipo de diagramas, EPFC facilita la navegación directa a cada uno de los componentes de la metodología (roles, tareas, productos de trabajo) identificando la trazabilidad bidireccional entre todos estos elementos.

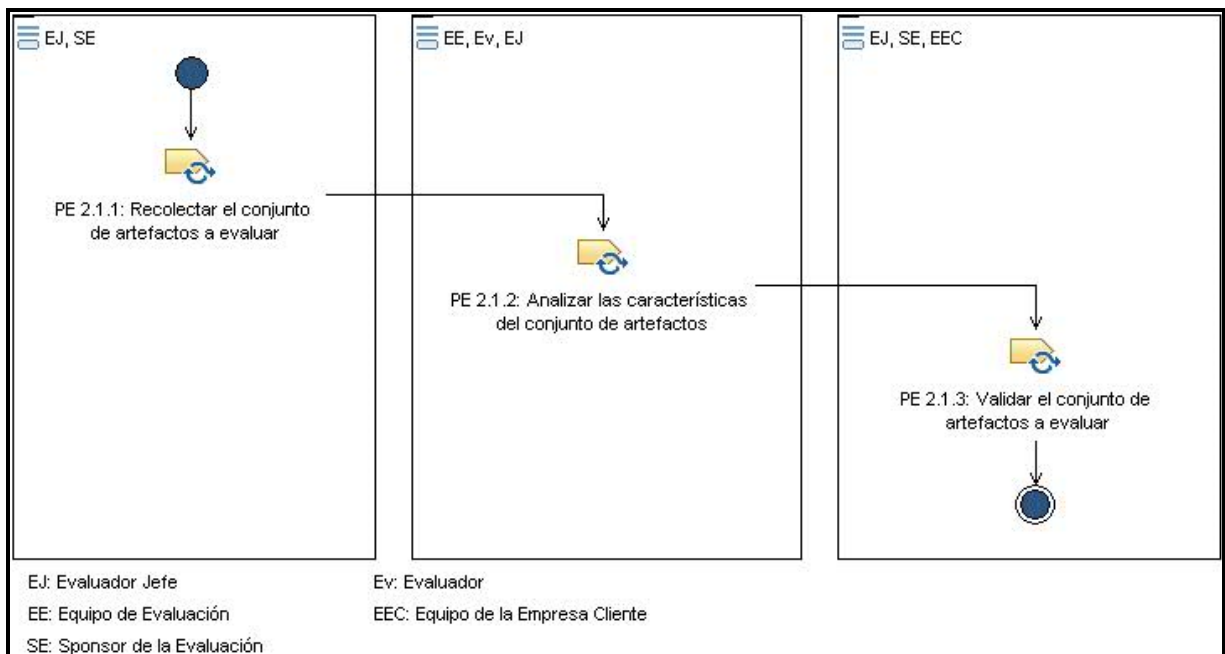


Figura 3. Diagrama de actividad generado por EPFC para la actividad “Obtención y Análisis de Artefactos a Evaluar” de la fase 2 (Especificación), del proceso de evaluación.

7. Conclusiones y trabajos futuros

La innovación que la metodología de evaluación EVVE proporciona, unida a las características de su implementación con SPEM y EPFC hacen de ella una metodología útil en el ámbito de la evaluación de calidad en los modelos UML.

En estudios futuros se pretende ampliar los márgenes de evaluación de calidad alcanzados con la metodología EVVE, para permitir evaluar los artefactos software generados desde las primeras fases del ciclo de vida hasta la entrega e implantación del producto (análisis, diseño, codificación y pruebas). Este nuevo entorno está pensado que se base en la nueva familia de normas ISO 25000, conocida como SQuaRE y que reemplazará a las actuales ISO 9126 e ISO 14598.

De esta manera, el nuevo entorno permitiría una evaluación continua del producto software, evaluando desde el análisis y diseño (utilizando EVVE), hasta el código, los casos de prueba y la documentación final de soporte, y permitiendo así observar desviaciones a lo largo de todo el proceso de desarrollo sin tener que esperar a las pruebas finales.

Agradecimientos

Esta investigación forma parte del proyecto EVVE (HITO-2008-49) financiado por la “Consejería de Ciencia y Tecnología de la Junta de Comunidades de Castilla-La Mancha”.

Referencias

- [1] Genero, M., Fernandez A. M., Nelson H.J., Poels G. y Piattini, M. "A Systematic Literature Review on UML Conceptual Modeling Quality." *Working Papers of Faculty of Economics and Business Administration*, Ghent University, 2008.
- [2] IEEE Computer Society, *IEEE Std. 1012-1998 Standard for Software Verification and Validation*, ISO, 1998.
- [3] IEEE Computer Society, *IEEE Std. 1028-2008 Standard for Software Reviews and Audits*, IEEE, 2008.
- [4] IEEE Computer Society, *IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology*, IEEE, 1998.
- [5] SEI, *CMMI® for Development, Version 1.2 (CMMI-DEV, V1.2), Improving processes for better products*, SEI, 2006.
- [6] SEI, *CMMI® for Acquisition, Version 1.2 (CMMI-ACQ, V1.2), Improving processes for acquiring better products and services*, SEI, 2007.
- [7] SEI, *CMMI® for Services, Version 1.2 (CMMI-SVC, V1.2), Improving processes for better services*, SEI, 2009.

- [8] ISO, *ISO/IEC DTR 9126 Software Product Evaluation - Quality Characteristics and Guidelines for their Use*, ISO, 2001.
- [9] ISO, *ISO/IEC 12207-2007. International Standard. Systems and software engineering — Software life cycle processes*, ISO, 2007.
- [10] ISO, *ISO/IEC 14598 1999-2001. Information Technology - Software Product Evaluation - Parts 1-6*, ISO, 1999.
- [11] ISO, *ISO/IEC 15504-5, Information technology - Process assessment - Part 5: An exemplar Process Assessment Model*, ISO, 2006.
- [12] ISO, *ISO/IEC 25000 Software and system engineering – Software product Quality Requirements and Evaluation (SQuaRE) –Guide to SQuaRE*, ISO, 2005.
- [13] ISO, *ISO/IEC 25001 Software and system engineering – Software product Quality Requirements and Evaluation (SQuaRE) –Evaluation planning and management*, ISO, 2005.
- [14] ISO, *ISO CD 25010.4, Software engineering-Software product Quality Requirements and Evaluation (SQuaRE) Quality model*, ISO, 2009.
- [15] ISO, *ISO/IEC 25020 Software and system engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Measurement reference model and guide*, ISO, 2007.
- [16] ISO, *ISO/IEC 25030 Software and system engineering – Software product Quality Requirements and Evaluation (SQuaRE) –Quality requirements*, ISO, 2007.
- [17] ISO, *ISO CD 25040.3: Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) – Evaluation reference model and guide*, ISO, 2009.
- [18] ISO, *ISO/IEC 25051 Software engineering-Software product Quality Requirements and Evaluation (SQuaRE)-Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing*, ISO, 2006.
- [19] ISO, *ISO/IEC 25062 Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Common Industry Format (CIF) for usability test reports*, ISO, 2006.
- [20] ISO, *ISO/IEC 29119 Software Testing. Part 1: Concepts and Vocabulary (Working Draft)*, ISO, 2009.
- [21] ISO, *ISO/IEC 29119 Software Testing. Part 2: Test Process (Working Draft)*, ISO, 2009.
- [22] ISO, *ISO/IEC 29119 Software Testing. Part 3: Test Documentation (Working Draft)*, ISO, 2009.
- [23] Koomen, T., van der Aalst, L., Broekman, B. y Vroon, M., *TMAP Next for result-driven testing*, UTN Publishers, 2006.
- [24] McGarry, J., Card D., Jones C., Layman, B., Clark, E., Dean C. y Hall, F., *Practical Software Measurement. Objective Information for Decision Makers*, Addison-Wesley, 2002.
- [25] Plösch, R., *The EMISQ method and It's Tool Support-Expert Based Evaluation on Internal Software Quality*. Journal of Innovations in Systems and Software Engineering, Springer London, vol. 4, nº 1, pp. 3-15, 2008.
- [26] Denger, C. y Elberzhager, F., *Basic Concepts to Define a Customized Quality Assurance Strategy*, Report of WP 3.1 in the LifeCycleQM project, Report Number IESE118.06/E, 2007.
- [27] IEEE Computer Society, *IEEE Std. 1028-1997 Standard for Software Reviews*, IEEE, 1997.

- [28] IEEE Computer Society, *IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology*, ISO, 1998.
- [29] SEI, *Process Maturity Profile. CMMI® For Development SCAMPI Class A Appraisal Results 2008 End-Year Update*, SEI, 2009.
- [30] ISO, *ISO/IEC 25010 Software and system engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Quality model and guide*, ISO, 2005.
- [31] OMG, *Software Process Engineering Metamodel Specification version 2.0*, OMG, 2007.
- [32] Eclipse Foundation, *Eclipse – An open development Platform*, Eclipse Foundation, 2007.

Sostenibilidad del software: más allá de Green IT

Guillermo Montoya

CEO, DEISER S.L.

www.deiser.com

guillermo.montoya@deiser.com

Introducción

En la actualidad parece consolidarse la tendencia a considerar la sostenibilidad como un factor fundamental en las TIC. No obstante, y parece lógico, todos los esfuerzos se centran en abordar el problema desde el punto de vista de las infraestructuras, centros de datos, servidores y, en general, los aspectos relacionados con los dispositivos que consumen energía. El concepto de Green IT es ya una realidad en miles de organizaciones que buscan un mayor compromiso con la gestión y cuidado de los recursos de nuestro planeta.

No obstante, las TIC conforman un panorama mucho más rico y complejo que debe ser abordado desde esta perspectiva. El concepto de sostenibilidad tiene mucho más alcance que el factor ecológico y debe aplicarse también a cualquier otro ámbito del panorama tecnológico.

Si partimos de la definición formal de Sostenibilidad, podemos adaptarlo a nuestras circunstancias particulares partiendo de sus tres pilares fundamentales: el Social, el Económico y el Ecológico. El esquema gráfico mostrado en la Figura 1 surge con las primeras declaraciones internacionales (Declaración de Río, 1.992), que defienden la sostenibilidad como la única manera viable de asegurar y preservar los recursos de nuestro planeta para las siguientes generaciones. En este caso, aplicaremos estas mismas ideas al campo de las tecnologías de la Información, y más concretamente a las áreas que generalmente aparecen menos relacionadas con esta idea.

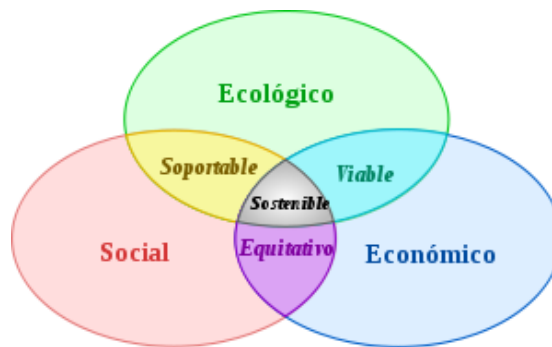


Figura 1. Pilares básicos de la sostenibilidad

Esta representación sienta las bases de los niveles de sostenibilidad en el que se puede encontrar una organización desde el punto de vista de sus proyectos y áreas relacionadas con el desarrollo de software, calidad, etc. Puesto que cualquier proyecto, en especial los de carácter tecnológico, está constituido de centenares de acciones y decisiones, debemos incorporar criterios de sostenibilidad en cada una de éstas decisiones. El conjunto de las mismas nos ayudará a establecer el grado de sostenibilidad, que emplea los nombres de las intersecciones que se producen entre los tres pilares para denominar cada uno de estos niveles (aparecen numerados a continuación, aunque no necesariamente los niveles intermedios deban tener esa numeración u orden):

1. **NO CONCIENCIADO O INSOSTENIBLE.** La organización no es consciente del problema, se toman decisiones puramente de negocio pero no se tienen en cuenta ninguno de los pilares de la sostenibilidad. Si se toman decisiones sostenibles es por pura coincidencia o azar.
2. **EQUITATIVO.** La organización toma conciencia de la necesidad de tomar medidas en los aspectos que parecen más cercanos: Social (empleados, clientes, proveedores, etcétera) y económico.
3. **VIABLE.** La organización es consciente del consumo de recursos perecederos y centra sus esfuerzos en ese pilar. Además, asocia de manera inmediata los aspectos económicos orientados a poder reducir los costes de las organizaciones, y lo combina con decisiones 'ecológicas'.
4. **SOPORTABLE.** Las organizaciones toman conciencia de la importancia del pilar social y ecológico, dando menor importancia al económico (aunque en época de crisis

la tentación será moverse a niveles 2 y 3 precisamente por considerar más importante el aspecto económico). Todas las decisiones tienen en cuenta el pilar ecológico y el pilar social. Un ejemplo podría ser la apuesta por las energías renovables que ahora cuestan mucho más dinero pero son claramente una apuesta ecológica y social.

5. **SOSTENIBLE.** Las organizaciones han tomado conciencia de que sólo la suma de decisiones conjuntas de los tres pilares tendrá los efectos adecuados para preservar dicha organizaciones a las generaciones (empleados) venideras. Todas las decisiones se toman de manera consciente y en toda la organización existe la concienciación y proactividad para que cada decisión que deba tomarse, valore la alternativa más o menos sostenible, desechando aquellas que no contemplen todos los aspectos en su conjunto.

Valorar la sostenibilidad de un proyecto o departamento TIC

En otras disciplinas, como el urbanismo, existen mecanismos que nos permiten tabular y medir, dentro de un rango acotado, la sostenibilidad de un proyecto. Algo tan sencillo como puntuar un aspecto entre 0 y 3 (nada sostenible, poco sostenible, bastante sostenible, muy sostenible) nos permitirá elaborar un conjunto de listas de comprobación organizadas por áreas o elementos a valorar del proyecto o departamento.

Esta valoración global permitirá obtener una suma de factores que nos darán una idea cuantitativa de la sostenibilidad de nuestros procesos o actividades TIC. Además, estableciendo un umbral determinado, que puede ser acordado en base a diferentes criterios estratégicos, obtendremos la manera de determinar si nuestras áreas objeto de estudio son o no sostenibles. Este umbral permitirá establecer una regla de corte para considerar el nivel de sostenibilidad que estamos aplicando a nuestras decisiones.

Ejemplos para establecer la sostenibilidad de un marco de actuación

Pilar ecológico

- Fomentar los formatos electrónicos: no papel, no dispositivos contaminantes.
- Diseñar procesos y sistemas de calidad limpios, con registros electrónicos.

- Usar herramientas software que no requieran grandes recursos de máquina, sencillas, distribuibles electrónicamente.
- Que se instalen con rapidez (reducción de consumo).
- Premiar aquellas iniciativas que acumulen una mayor cantidad de logros en este sentido (pensando ‘ecológicamente’).
- Implantar soluciones sencillas, rápidamente y con escasez de consumo de recursos.
- Evitar desplazamientos de personal y reuniones presenciales sustituibles por otras alternativas (reducción de emisiones de CO2 por ahorro de desplazamientos innecesarios).

Pilar Social

- Facilitar sistemas globalizados y modelos combinados de desarrollo.
- Aprovechar la web 2.0 para implicar más y mejor a todas las partes.
- Acercar ‘de verdad’ a los implicados mediante interfaces apropiados para cada caso.
- Simplificar los lenguajes y las herramientas.
- Fomentar la colaboración e integración de cada individuo del equipo: cliente, proveedor, etcétera.
- Crear sistemas de calidad perdurables en el tiempo, asequibles en su ejecución y asumibles por todos.

Pilar Económico

- Ajustar las planificaciones y aplicar mentalidad “Agile” en nuestros proyectos de calidad y desarrollo de software. Además del desarrollo de software, la calidad puede ser reutilizable y ofrecer patrones claramente identificables.
- Buscar soluciones económicas pero suficientes y equilibradas.
- Garantizar la adquisición y mantenimiento de las herramientas software dentro de unos costes razonables: no a los martillos de oro y brillantes (la herramienta no puede ser mucho más costosa que el servicio de valor)
- Asegurar una inversión reducida por usuario y año de manera prolongada en el tiempo.

- Crear sistemas de ayuda y soporte virtualizados que permitan la optimización de los recursos asignados y la aplicación de tarifas y costes más razonables y ajustables.

La sostenibilidad ha venido para quedarse

Parece claro que tendremos que adaptarnos con enorme rapidez a estos cambios. La sociedad demanda compromisos y acciones que no tengan que ver necesariamente con cuestiones estrictamente económicas. Más allá de la Responsabilidad Social Corporativa, la Sostenibilidad será un criterio esencial en muchas organizaciones, especialmente aquellas que están obligadas a ofrecer a sus consumidores una cadena de suministro alineada con los compromisos estratégicos de preservación de los recursos ofrecidos por la naturaleza.

Muy pronto, este criterio será uno más, tan importante como los demás y puede que con el tiempo más importante aún, que contribuirá a tomar las decisiones estratégicas, tácticas y operativas dentro de los departamentos TIC. Más aún, será el hecho diferencial que permitirá a las empresas de servicios poder continuar trabajando para miles de empresas que han hecho, están haciendo y harán de la sostenibilidad uno de sus elementos estratégicos. Y tenemos que estar preparados para ello.

Perfil profesional



Guillermo Montoya Fanegas, licenciado en Informática por la Universidad Politécnica de Madrid, es en la actualidad Director General de DEISER, empresa de servicios TIC especializada en mejora de procesos TI y calidad de software, desde su creación en 1.998. Ha dedicado toda su vida profesional, desde el año 1.991, a las actividades de consultoría y ha participado en múltiples iniciativas y comités relacionados con las metodologías y la mejora de procesos.

En la actualidad, combina su responsabilidad como CEO de DEISER con la participación en diferentes grupos e iniciativas: miembro fundador del Grupo de Calidad de la Plataforma INES y ponente habitual en jornadas de esta disciplina que se organizan en España.