

Félix García Rubio, Francisco Ruiz González, Mario Piattini Velthuis

Grupo Alarcos, Escuela Superior de Informática, Universidad de Castilla-La Mancha (UCLM)

<{Felix.Garcia, Francisco.RuizG, Mario.Piattini}@uclm.es>

# Medición de los Procesos Software

## 1. Introducción

La mejora del proceso software se ha convertido en un objetivo fundamental de las empresas dedicadas al desarrollo y mantenimiento de sistemas informáticos. Esta necesidad de mejora del proceso es debida a la correlación que existe entre la calidad del proceso y la calidad del producto, de forma que para obtener mejores productos es necesario disponer de mejores procesos.

A la hora de satisfacer los requisitos de calidad de los procesos software es necesario que produzcan los resultados esperados, que estén correctamente definidos y que sean mejorados en función de los objetivos de negocio, muy cambiantes en empresas altamente competitivas. Estos son los objetivos de la **Gestión del Proceso Software** [5] que, para ser aplicada de forma efectiva, supone asumir cuatro responsabilidades clave: Definir, Medir, Controlar y Mejorar el Proceso.

Estas responsabilidades y sus relaciones se representan en la **figura 1**.

De acuerdo a estas responsabilidades (que son básicas para una gestión exitosa de los procesos software), para llevar a cabo de una forma eficiente la mejora del proceso es necesario tener en cuenta los siguientes aspectos:

■ **Definición del Proceso.** Es la primera responsabilidad clave que hay que asumir para poder realizar una gestión efectiva orientada a la mejora de procesos. Para ello, es necesario modelar los procesos, es decir, representar los elementos de interés que intervienen en los mismos. Dada la diversidad de elementos que hay que tener en cuenta cuando se habla de un proceso software, la definición de los procesos software en una organización no es una tarea trivial.

En la literatura se pueden encontrar diversos lenguajes y formalismos de modelado, conocidos como "Lenguajes de Modelado de Procesos" (LMP), que tienen como objetivo representar, de una forma precisa y no ambigua, los diferentes elementos relacionados. En general, en un proceso software se pueden identificar los siguientes elementos o conceptos generales (aunque con notaciones y términos distintos) en los diferentes LMPs [3]: Actividad, Producto, Recurso, Organización y Rol. Con el fin de proporcionar una referencia común para representar los procesos software, el consorcio OMG (*Object Management Group*) está trabajan-

**Resumen:** la medición de los Procesos Software juega un papel fundamental en su mejora, ya que proporciona la base cuantitativa necesaria para poder detectar los aspectos en los que enfocar los programas de mejora. Sin embargo, dada la gran diversidad de factores y elementos que están relacionados, la medición de los procesos software no es una tarea trivial. Así, para poder medir los procesos de forma efectiva y facilitar la toma de decisiones orientadas a su mejora, es necesario identificar los tipos de entidades que son de interés desde el punto de vista de la medición. También hace falta llevar a cabo programas de medición que permitan, además de medir las entidades relevantes de forma aislada, integrar y relacionar toda la información resultado de la medición. En este artículo se muestra una panorámica general de la medición de los procesos software, destacando la importancia de este tema en la gestión de los procesos orientada a su mejora. También se identifican las entidades relevantes que se pueden medir con relación al proceso y se presenta un ejemplo sobre cómo medir uno de estos tipos de entidades: los modelos de procesos.

**Palabras clave:** medición de procesos, mejora de Procesos Software, métricas software, modelos de Procesos Software, Proceso Software.

## Autores

**Félix García Rubio** es Ingeniero en Informática por la Universidad de Castilla-La Mancha (UCLM). Actualmente se encuentra finalizando los estudios de Doctorado en la citada universidad. Es Profesor Asociado en el Depto. de Informática de la UCLM, en Ciudad Real (España). Autor de varios artículos y capítulos de libro sobre la gestión de los procesos software desde el punto de vista de su modelado, medición y tecnología. Es miembro del grupo de investigación Alarcos. Sus líneas de investigación son los procesos software y la medición software.

**Francisco Ruiz González** es Doctor en Informática por la Universidad de Castilla-La Mancha (UCLM) y Licenciado en CC. Químicas, especialidad Química-Física, por la Universidad Complutense de Madrid en 1983. Es profesor del Depto. de Informática de la UCLM desde 1989, destinado en la Escuela Superior de Informática de Ciudad Real, de la cual ha sido Director entre 1993 y 2000. Anteriormente fue Director de los Servicios Informáticos de la UCLM (1985-1989) y también trabajo de analista-programador y jefe de proyectos en varias compañías privadas. Es miembro del grupo Alarcos de I+D, <<http://alarcos.inf.uclm.es>>. Sus temas de investigación actuales incluyen: tecnología y modelado de procesos software, mantenimiento del software, y metodologías para planificar y gestionar proyectos software. Otros temas de trabajo en el pasado han sido SIG (Sistemas de Información Geográficos), entornos software para la educación y bases de datos deductivas. Ha publicado ocho libros y catorce capítulos sobre los temas citados y tiene 90 publicaciones en revistas y conferencias nacionales e internacionales. Ha sido miembro de nueve comités de programa y de siete comités organizadores de congresos. Pertenece a diversas asociaciones científicas y profesionales: ACM, IEEE-CS, ATL, AEC, AENOR, ISO JTC1/SC7, EASST, AENUI y ACTA.

**Mario Piattini Velthuis** es Ingeniero y Doctor en Informática por la Universidad Politécnica de Madrid. Es también *Certified Information System Auditor Manager (CISA)* por la ISACA (*Information System Audit and Control Association*). Catedrático en el Depto. de Informática de la Universidad de Castilla-La Mancha (UCLM), en Ciudad Real, España. Autor de diversos libros y artículos sobre bases de datos, Ingeniería del Software y sistemas de información, dirige el grupo de investigación Alarcos del Depto. de Informática de la UCLM, en Ciudad Real. Sus líneas de investigación son: diseño avanzado de bases de datos, calidad de bases de datos, métodos de software, mantenimiento de software y seguridad en sistemas de información. Ha coeditado *Advanced Database Technology and Design*, 2000, Artech House, UK; *Auditing Information Systems*, Idea Group Publishing, 2000, USA; *Information and database quality*, 2002, Kluwer Academic Publishers, USA y *Metrics for Software Conceptual Models*, 2004, Imperial College Press, UK. Es coordinador de la sección técnica "Bases de datos" de *Novática*.

do desde hace varios años en el desarrollo del metamodelo SPEM (*Software Process Engineering Metamodel*) [14], que es un lenguaje genérico que extiende UML (*Unified Modeling Language*) para el modelado des-

criptivo de procesos software sin incluir aspectos relacionados con la ejecución (*enactment*) de los mismos. Actualmente SPEM constituye una especificación a partir de la cual se prevé obtener un estándar de



## Las métricas software son necesarias para la mejora del proceso



concretos se obtienen productos. Por lo tanto, para establecer un marco de medición dentro de una organización es necesario abordar las tres dimensiones siguientes:

■ **Medición del Proceso**, basada en el estudio y control de la capacidad de los procesos, así como en la gestión de los cambios en dichos procesos. Las métricas de proceso se extraen midiendo las características de tareas específicas de la Ingeniería de Software y obteniendo como resultados métricas sobre los errores detectados antes de la entrega del software, defectos detectados e informados por los usuarios finales, esfuerzo humano y tiempo consumido, ajuste con la planificación, etc..

A la hora de medir el proceso también resulta conveniente poder establecer indicadores relacionados con el propio modelo del proceso, ya que es posible que la complejidad u otras características de calidad del modelo puedan afectar a su ejecución y por consiguiente a la calidad de los productos finales obtenidos. A pesar de esto, en la literatura los trabajos sobre medición del proceso se han centrado en la medición de los proyectos y productos. Para compensar en parte esta falta, en el apartado siguiente se describe cómo realizar la medición de un proceso software a nivel conceptual, se presenta un conjunto de métricas de modelos de procesos software y se destacan los aspectos del proceso que se pueden evaluar con el uso de estas métricas.

■ **Medición del Proyecto**, basada en la gestión de proyectos, que ya es un campo maduro desde hace años.

■ **Medición del Producto**. La medición del producto software está centrada en evaluar la calidad de los entregables. Sobre esta dimensión existen multitud de estudios, propuestas y métricas desde hace bastantes años, algunas tan conocidas y asentadas como las

líneas de código fuente, los puntos función o la complejidad ciclomática de McCabe.

### 3. Medición del Proceso a nivel Conceptual

Puesto que el estudio de la evaluación de procesos se ha centrado en la recogida de datos del proyecto para obtener medidas de rendimiento, productividad, eficiencia, etc., la situación resultante es que no se han definido métricas explícitas para modelos de procesos. La definición y validación de métricas para modelos de procesos nos puede permitir estudiar la influencia de la complejidad estructural de dichos modelos de procesos sobre su *mantenibilidad* (facilidad de mantenimiento o de cambio), teniendo en cuenta que un modelo de procesos con un alto grado de complejidad será más difícil de cambiar, lo que reducirá sensiblemente su facilidad de mantenimiento. Ello influirá en la mejora del proceso y, en consecuencia, también podrá repercutir en los proyectos (que serían más costosos en recursos y tiempo) y en la calidad final de los productos obtenidos.

La principal base teórica para el desarrollo de modelos cuantitativos que relacionan los atributos de complejidad estructural y de calidad externa de artefactos software ha sido establecida en [2]. La **figura 3** la ilustra de forma gráfica. Tal y como se sugiere en dicha figura, es posible estimar la mantenibilidad a partir de un conjunto de métricas que miden propiedades estructurales de dichos modelos (tamaño, acoplamiento, etc.). El mantenimiento de los modelos de procesos software consiste en realizar modificaciones sobre los mismos con el fin de mejorarlos, corrigiendo errores que puedan tener, adaptándolos a nuevas necesidades, o mejorando algunas de sus propiedades (como la calidad). Por ejemplo, podría

ser necesario realizar una corrección sobre un modelo de procesos en el que existen actividades que no reciben productos de entrada o que no producen nada a la salida, o quizás tener que mejorar un modelo eliminando dependencias innecesarias entre actividades. Modelos de procesos software difíciles de mantener pueden afectar negativamente a la ejecución de los proyectos y a la calidad final de los productos obtenidos.

En conclusión, para poder evaluar la facilidad de mantenimiento de los modelos de procesos software es necesario: i) definir un conjunto de métricas que nos permitan evaluar la complejidad estructural; ii) demostrar la utilidad de dichas métricas realizando estudios empíricos para comprobar si pueden ser usadas como indicadores de la mantenibilidad de los modelos. En los siguientes subapartados se presenta un conjunto de métricas para modelos de procesos y un ejemplo de cálculo de ellas.

#### 3.1. Métricas para Modelos de Procesos Software

Las métricas presentadas a continuación han sido definidas considerando la terminología de SPEM [14], pero pueden ser aplicables directamente con otros lenguajes de modelado ya que las diferencias radican prácticamente sólo en la terminología. El modelo conceptual de SPEM está basado en la idea de que un proceso de desarrollo de software consiste en la colaboración entre entidades abstractas y activas, denominadas "roles de proceso", que realizan operaciones denominadas "actividades" sobre entidades tangibles denominadas "productos de trabajo". A la hora de establecer las métricas de los modelos de procesos se consideran dos niveles de alcance:

■ **Nivel de Modelo**. Son métricas que se aplican para medir la complejidad estructu-

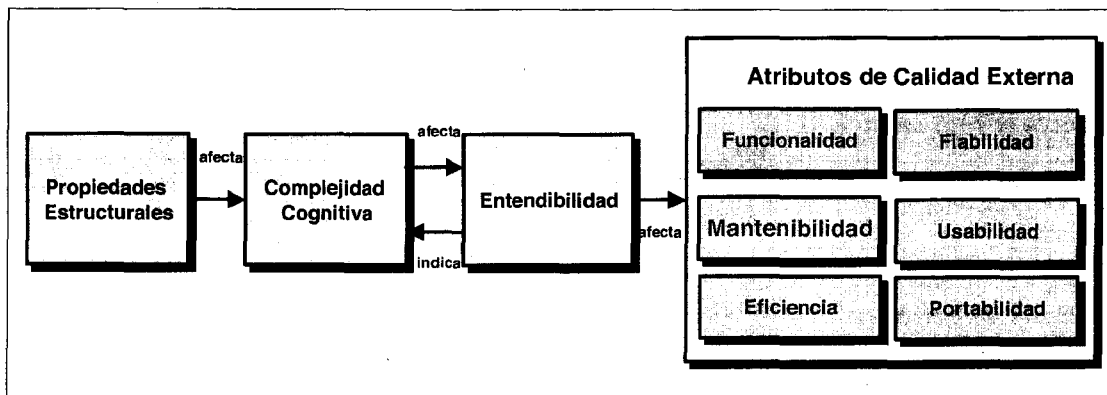


Figura 3. Modelo de Complejidad Estructural de Briand et al. [2].



La comunidad dedicada a los servicios Web también ha desarrollado normas para el modelado de los procesos



NA	NPT	NRP	NDPTIn	NDPTOut	NDPT	NDPA	NCA	RDPTIn	RDPTOut	RPTA	RRPA
5	8	4	13	6	19	4	1,25	0,68	0,32	1,6	0,8

Tabla 2. Valores de las Métricas para el ejemplo de la figura 4.

proceso software con dos tecnologías recientes: “Flujos de Trabajo” y “Servicios Web”.

Los “Sistemas de Gestión de Flujos de Trabajo” [17] dan soporte al modelado, ejecución y gestión de procesos de negocio. Por tanto, y tal como algunos autores han postulado [13], pueden ser una herramienta muy útil para que los ingenieros software gestionen y lleven a cabo sus procesos de desarrollo y mantenimiento. En esta línea son significativos los nuevos estándares para representar procesos mediante flujos de trabajo. El principal ejemplo de ellos es el “*Workflow Process Definition Interface - XML Process Definition Language (XPDL)*” [18].

Desde el ámbito de la tecnología de servicios Web también se ha estudiado el problema del modelado de los procesos. Al fin y al cabo, el diseño de una aplicación basada en invocar una colección de servicios web se parece mucho al diseño del modelo del proceso de negocio que dicha aplicación soporta.

Es por ello, que la comunidad dedicada a los servicios Web también ha desarrollado normas para el modelado de los procesos: BPML (*Business Process Modeling Language*), BPSS (*Business Process Specification Schema*), BPEL4WS (*Business Process Execution Language for Web Services*) y WSCI (*Web Service Choreography Interface*) son algunas de las más significativas. A este respecto, al lector le pueden resultar muy interesantes los números, ambos de octubre de 2003 de “Communications de ACM” (dedicado a Informática orientada a servicios) y de “IEEE Computer” (dedicado al software como servicio).

La confluencia e integración de estas tecnologías abrirá nuevas posibilidades para la manera en que el ingeniero software podrá realizar su trabajo y, en especial, en los aspectos que tienen que ver con la gestión y mejora de los procesos. Así, cabe pensar en que los modelos de los procesos software usados para el desarrollo y mantenimiento del software podrán ser diseñados y gestionados usando un Sistema de Gestión de Flujos de Trabajo, el cual invocará, para llevar a cabo ciertas actividades automáticas o semiautomáticas, a determinados servi-

cios Web que harán las veces de herramientas CASE (por ejemplo, un servicio de compilación o pruebas unitarias) o de soporte a la gestión y la organización. Todo este desarrollo, previsible en los próximos años, pasará por que los ingenieros software tendrán que prestar especial atención, además de al qué hacen (producto), al cómo lo hacen (proceso). Y como buenos ingenieros, deberán medir ambos, el producto y el proceso.

## Referencias

- [1] V. Basili, D. Weiss. A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, 10, pp. 728-738, 1984.
- [2] L. Briand, S. Arisholm, F. Counsell, F. Houdek, P. Thévenod-Fosse. Empirical Studies of Object-Oriented Artifacts, Methods, and Processes: State of the Art and Future Directions. *Empirical Software Engineering*, 4(4), pp. 387-404, 1999.
- [3] J.C. Derniame, B.A. Kaba, D. Wastell. *Software Process: Principles, Methodology and Technology* (Vol. LNCS 1500). Springer-Verlag, 1999.
- [4] N. Fenton. Metrics for Software Process Improvement. In M. Haug, E. W. Olsen & L. Bergman (Eds.), *Software Process Improvement: Metrics, Measurement and Process Modelling* (pp. 34-55). Springer, 2001.
- [5] W.A. Florac, A.D. Carleton. *Measuring the Software Process. Statistical Process Control for Software Process Improvement*. Addison Wesley, 1999.
- [6] F. Garcia, F. Ruiz, J.A. Cruz, M. Piattini. Integrated Measurement for the Evaluation and Improvement of Software Processes. *9th European Workshop on Software Process Technology (EWSPT'9)*, Helsinki (Finland), pp. 94-111, 2003.
- [7] F. Garcia, F. Ruiz, M. Piattini. Definition and Empirical Validation of Metrics for Software Process Models. *5th International Conference Product Focused Software Process Improvement (PROFES' 2004)*, Kansai Science City (Japan), pp. 146-158, 2004.
- [8] F. Garcia, F. Ruiz, M.F. Bertoa, C. Calero, M. Genero, L.A. Olsina, M.A. Martín, C. Quer, N. Tondori, S. Abrahao, A. Vallecillo, M. Piattini. Una Ontología de la Medición del Software. Technical Report, Depto. de Informática, Universidad de Castilla-La Mancha. Disponible en <<http://www.info-ab.uclm.es/rep.php? & codrep=DIAB-04-02-2>>, 2004.
- [9] IEEE. *IEEE Std 1061-1992, "IEEE Standard for a Software Quality Metrics Methodology"*, 1992.
- [10] ISO/IEC. *ISO IEC 15504 TR2:1998, Software Process Assessment - Part 4: Guide to conducting assessment*. International Organization for Standardization, 1998.
- [11] ISO/IEC. *ISO 15939: Software Engineering - Software Measurement Process*, 2002.
- [12] J. McGarry, D. Card, C. Jones, B. Layman, E. Clark, J. Dean, F. Hall. *Practical Software Measurement. Objective Information for Decision Makers*. Addison-Wesley, 2002.
- [13] C. Ocampo, P. Botella. *Some Reflections on Applying Workflow Technology to Software Processes*. Universitat Politècnica de Catalunya, Dep. de Llenguajes y Sistemes Informàtics, Technical Report TR-LSI-98-5-R, Barcelona, 1998.
- [14] OMG. *Software Process Engineering Metamodel Specification; adopted specification*. Object Management Group, 2002.
- [15] H.D. Rombach. Design measurement: some lessons learned. *IEEE Software*, 7(3), pp. 17-25, 1990.
- [16] SEI. *Capability Maturity Model Integration (CMMI<sup>SM</sup>), version 1.1*. Software Engineering Institute, 2002.
- [17] WfMC. *TC00-1003 1.1: The Workflow Reference Model*. Workflow Management Coalition, January-1995.
- [18] WfMC. *TC-1025 final draft 1.0: Workflow Process Definition Interface - XML Process Definition Language (XPDL)*. Workflow Management Coalition, October-2002.