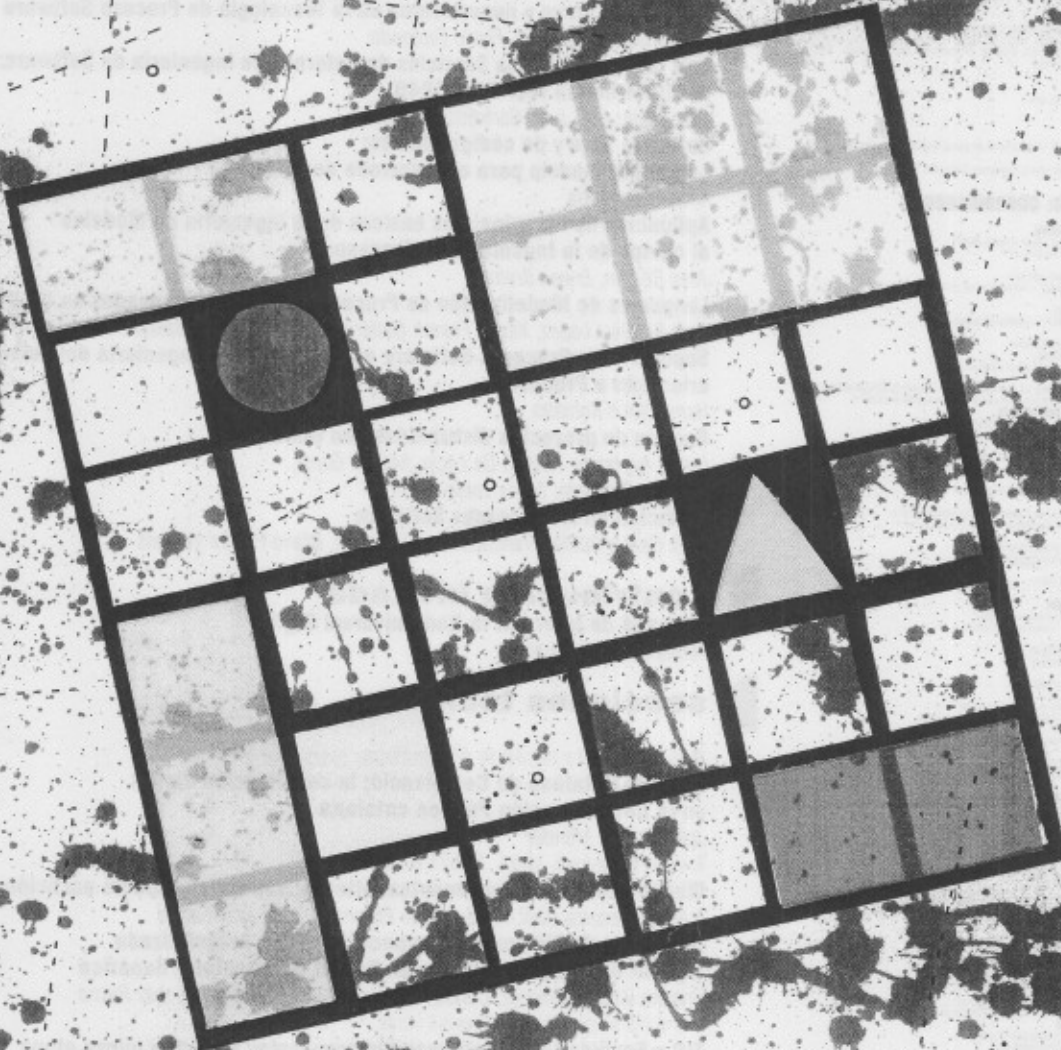


# NOVÁTICA

UPGRADE

Revista de la Asociación  
de Técnicos de Informática

Nº 171, septiembre-octubre 2004, pp. XCV



## Tecnología de Proceso Software



## en resumen TPS o el software como proceso

Rafael Fernández Calvo  
Coordinación Editorial de *Novática*

<rfcalvo@ati.es>

Apreciada lectora / Querido lector:

La mejora del desarrollo de software es un campo donde se producen constantemente novedades de interés que *Novática* ha ido reflejando, desde su creación en 1975, tanto en diversas monografías (recordaremos la última, "Ingeniería del Software, estado de un arte", publicada en el número 164, julio-agosto de 2003) como en numerosos artículos publicados en las dos secciones técnicas relacionadas con el tema que han existido a lo largo de la vida de nuestra revista: "Calidad del Software" e "Ingeniería del Software", que el pasado año se integraron en una sola con el nombre de la segunda.

Persistiendo en este loable empeño, en la monografía de este número 171 el foco se ha proyectado sobre uno de los últimos avances en el campo del desarrollo de software: la "Tecnología de Proceso Software". Para los no iniciados, consiste básicamente en considerar el desarrollo del software como un proceso en sí mismo, que, como todos los procesos de negocio de cualquier organización, requiere estrategias, metodologías y herramientas para su correcta implantación. Con el fin de ofrecer una panorámica de esta nueva tendencia, los editores invitados --nuestro

compañero **Francisco Ruiz González**, de la Universidad de Castilla-La Mancha, y **Gerardo Canfora**, de la *Università degli Studi del Sannio* (Benevento, Italia)-- han seleccionado un conjunto de artículos de autores locales y foráneos que esperamos y deseamos contribuyan a incrementar el acervo profesional de nuestros lectores.

Hay más contenidos de interés en este número, entre los que deseamos resaltar los que nos ofrecen dos artículos: uno del Presidente de ATI, **Josep Molas i Bertrán**, en el que presenta las propuestas de la asociación editora de *Novática* sobre el desarrollo de la Sociedad de la Información y el papel de las organizaciones representativas de los profesionales informáticos en la nueva etapa abierta con la victoria del PSOE en las elecciones del 14 de marzo, y otro de **Fernando Sáez Vacas** sobre lo que él denomina "Red Universal Digital", que inaugura una nueva sección llamada "Contribución invitada", en la que cederemos nuestras páginas a personas con una demostrada capacidad de reflexionar, pero también y sobre todo, de incitar a la reflexión sobre las TIC y sus repercusiones de todo tipo desde perspectivas diversas y novedosas. Los artículos que aparecen en

las secciones técnicas "Administración Pública electrónica", "Ingeniería del Software" y "Seguridad", en "Personal y transferible", y las habituales "Referencias autorizadas", más la solución del sexto problema del Primer Concurso Universitario de Programación de la Comunidad de Madrid (CUPCAM) del año 2003, coronan este número.

Llegados a este punto, queremos destacar que con la solución de dicho problema ponemos fin a la serie de los planteados en el citado concurso, iniciada en el número 166 (noviembre-diciembre 2003) y que hemos podido ofrecer a nuestros lectores gracias al generoso y continuado esfuerzo de un grupo de docentes informáticos encabezado por nuestro colaborador **Cristóbal Pareja Flores**. A todos los que han participado en esta valiosa iniciativa de forma altamente profesional y completamente desinteresada queremos expresarles nuestro más sincero agradecimiento

Cordiales saludos,



# UPENET

## UPGRADE European NETwork

The network of CEPIS member societies' publications

Current partners:

**Mondo Digitale** (AICA, Italy), **Novática** (ATI, Spain),  
**Pliroforiki** (CCS, Cyprus), **Pro Dialog** (PTI-PIPS, Poland)

Francisco Ruiz González<sup>1</sup>,  
Gerardo Canfora<sup>2</sup>

<sup>1</sup> Grupo Alarcos, Escuela Superior de Informática, Universidad de Castilla-La Mancha; <sup>2</sup> Research Centre on Software Technology (RCOST), Università degli Studi del Sannio (Benevento, Italia)

<Francisco.RuizG@uclm.es>  
<canfora@unisannio.it>

## 1. Introducción

Una de las principales líneas de trabajo para la mejora de la calidad de los productos software es el estudio y mejora de los procesos mediante los cuales el software es desarrollado y mantenido. La afirmación anterior está basada en la asunción de que existe una correlación directa entre la calidad del proceso y la calidad del producto obtenido. El área de trabajo, dentro del campo de la Ingeniería del Software, que aborda esta problemática es conocida con el nombre de "Tecnología de Proceso Software" (TPS), o simplemente "Proceso Software". Como disciplina autónoma, la TPS comenzó en los años 80 a nivel de investigación (*International Software Process Workshop, European Workshop on Software Process Technology, Journal of Software Process: Improvement and Practice, ...*), si bien ha sido en los últimos 5 o 6 años cuando ha adquirido una cierta madurez en cuanto a su utilización real en proyectos de Ingeniería del Software.

La primera contribución importante de la TPS ha sido la confirmación de que el desarrollo y mantenimiento del software son procesos complejos, que requieren un esfuerzo colectivo y creativo. Por tanto, la calidad de un producto software depende fuertemente de las personas, la organización y los procedimientos utilizados para crearlo, entregarlo y mantenerlo.

## 2. El contenido de esta monografía

Esta monografía comienza con el artículo "Procesos Software. Características, Tecnología y Entornos", que los editores invitados hemos preparado para servir de introducción al tema en tres aspectos esenciales: características propias de los procesos software; justificación de la TPS como medio de dar soporte integrado tanto a los procesos de producción como a los de gestión; y Entornos de Ingeniería del Software (EIS).

**Nota del Editor de Novática:** por razones de espacio no se incluye en esta monografía el artículo "Process Diversity: How Practitioners Can Manage It" de **Danilo Calvano** y **Corrado Aaron Visaggio**, que fue seleccionado por los editores invitados.

Este artículo ha sido publicado en el número 5/2004 de **Upgrade**, <<http://www.upgrade-cepis.org>>, en inglés, y será publicado en un próximo número de **Novática**, en castellano.

# Presentación La Tecnología de Proceso Software y la mejora de la gestión de los proyectos y de la calidad de los productos

## Editores invitados

**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 trabajó de analista-programador y jefe de proyectos en varias compañías privadas. Es miembro del grupo Alarcos de I+D, <<http://alarcos.inf-cr.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. Pertenecer a diversas asociaciones científicas y profesiona-

les: ACM, IEEE-CS, ATI, AEC, AENOR, ISO JTC1/SC7, EASST, AENUI y ACTA.

**Gerardo Canfora** es Catedrático de Informática en la Facultad de Ingeniería y Director del Research Centre on Software Technology (RCOST) de la Università degli Studi del Sannio, en Benevento (Italia). Ha participado en las comités de programa y de organización de diversas conferencias internacionales. Ha sido co-presidente del comité de programa de "International Workshop on Program Comprehension" (IWPC'1997); "International Conference on Software Maintenance" (ICSM'2001); y "European Conference on Software Maintenance and Reengineering" (CSMR'2004). También ha sido presidente general del CSMR'2003. Sus intereses de investigación incluyen mantenimiento y evolución del software, comprensión de programas e ingeniería inversa, mejora del proceso software, gestión del conocimiento, e ingeniería del software orientada a servicios. Sobre estos temas ha publicado más de 100 artículos en revistas y conferencias internacionales. Es editor asociado del "IEEE Transactions on Software Engineering" y participa en el consejo editorial del "Journal of Software Maintenance and Evolution: Research and Practice".

En este último punto se remarcan las distintas dimensiones de integración de las herramientas software que forman un EIS y la propuesta de que los EIS sean orientados a procesos (*Process-centered Software Engineering Environment*, PSEE).

"*Cuestiones Clave y Nuevos Retos en la Tecnología de Proceso Software*", escrito por **Jean Claude Derniame** y **Flavio Oquendo** (ambos son protagonistas principales de la serie de conferencias EWSPT), es un análisis de la evolución y resultados en los 20 años de vida de este campo y de los principales retos que la TPS tiene abiertos en estos momentos: dar soporte a los procesos típicos de los métodos ágiles, el software de código abierto (*open source*) o el desarrollo del software a escala mundial (globalización). La primera parte es una introducción a la TPS que complementa muy bien al artículo introductorio presentando un marco de trabajo general para procesos y la relación entre la TPS y la madurez de los procesos (de cara a su mejora).

Como prueba del nivel de madurez industrial que la TPS está alcanzando, **Dan H. Lee** y **Juan Garbajosa Sopena**, en "*Una taxonomía de los Servicios de Entornos de Ingeniería de software: La futura norma ISO/IEC 15940*", nos presentan el futuro estándar ISO del que son co-editores. Los autores clasifican, enumeran y definen todos los posibles servicios que un EIS puede proveer para dar soporte automático a los diversos procesos del ciclo de vida del software.

Existe una estrecha relación entre los modelos de negocio utilizados por la industria del software y los procesos llevados a cabo para desarrollar y mantener el software. Por esta razón, las reflexiones, análisis y aclaraciones que hace **Alfonso Fuggetta** en "*Software libre y de código abierto, ¿Un nuevo modelo para el desarrollo de software?*" son muy interesantes, útiles y clarificadoras.

Los dos siguientes artículos tratan sobre el modelado de los Procesos software. En "*Aplicación de los principios básicos de la ingeniería de modelos al campo de la Ingeniería*"

de procesos", **Jean Bezivin** y **Erwan Breton** hacen una introducción a la ingeniería de sistemas dirigida por modelos, presentando la propuesta MDA (*Model-Driven Architecture*) del OMG (*Object Management Group*), cuyo objetivo es poder separar, al diseñar la arquitectura de los sistemas software, los aspectos independientes de la plataforma de los dependientes. Como prueba de la potencia del nuevo paradigma de desarrollo *model-driven*, los autores muestran su aplicación a procesos utilizando un software empaquetado que no está basado en MDA: MS-Project. Con ello demuestran que esta aproximación permite el diseño y construcción de soluciones más generales.

En "*Lenguajes de Modelización de Procesos Software basados en UML*", **Perc Botella i López**, **Xavier Franch Gutiérrez** y **Josep M. Ribó Balust** introducen al lector en los lenguajes de Modelado de Procesos (LMP). En particular, analizan las posibilidades de UML (*Unified Modeling Language*) como formalismo para modelar los aspectos estructurales y de comportamiento de los procesos, y presentan los LMP que sacan partido de lo anterior para modelar procesos software (SPM y PROMENADE).

Cambiando el punto de interés, otros dos artículos versan sobre los aspectos más tec-

nológicos de los EIS. **Hans-Ulrich Kobiak**, en "*Soporte a los Procesos Software en un Entorno de Ingeniería del Software orientado a Procesos*", realiza un estudio sistemático de los requisitos de soporte a procesos que es deseable que satisfaga un PSEE y propone una lista de ingredientes (grupos de servicios) para tal fin. Para dar una visión más concreta del asunto, el autor presenta los mecanismos disponibles en el LMP ALADYN para la automatización de procesos (disparadores, patrones de tarea, restricciones, etc.) y para controlar el impacto (permisos).

El artículo "*Gestión de proyectos distribuidos con GENESIS*", escrito por **Lerina Aversano**, **Andrea de Lucia**, **Matteo Gaeta**, **Pierluigi Ritrovato**, y **Maria Luisa Villani**, presenta una aproximación para dar soporte automatizado a la gestión de proyectos software distribuidos, que permite definir y reificar (es decir, ejecutar de forma automática por un PSEE de forma similar a como un intérprete puede ejecutar el código de un programa) modelos de procesos software de forma descentralizada y autónoma en diferentes sitios.

En "*Medición de los Procesos Software*", **Félix García Rubio**, **Francisco Ruiz González** y **Mario Piattini Velthuis** argumentan la importancia de medir los proce-

dos software para poder realizar su evaluación y mejora. Los autores identifican las entidades de un proceso software factibles de ser medidas y, como ejemplo, presentan una colección de métricas que sirven para estimar la mantenibilidad de un modelo de proceso, es decir, la facilidad con que dicho modelo puede ser modificado y adaptado.

Es habitual que un proceso experimente diversas adaptaciones debidas a los distintos contextos operativos en los que se ejecuta el proceso. Estas adaptaciones suponen versiones diferentes del mismo proceso genérico, conocidas como procesos especializados.

Esperamos que esta colección de artículos, cuya elaboración agradecemos a los autores, ofrezca una primera aproximación y una introducción a la Tecnología de Proceso Software. Estamos convencidos de que, mediante la automatización e integración de los diversos procesos de ingeniería y de gestión, este campo está llamado a ser una ayuda importante para los ingenieros de software en los próximos años.

## Referencias útiles sobre "Tecnologías de Agentes"

Para quienes estén interesados en profundizar en la materia, estas referencias complementan las incluidas en los diferentes artículos que componen la monografía.

### Asociaciones

■ ISPA - International Software Process Association. <<http://www.ece.utexas.edu/~perry/prof/ispa/>>

### Libros

■ **Fuggetta y A. Wolf** (eds). *Software Process*, Vol 4, Trends in Software. J. Wiley & Sons (USA), 1996. <<http://serl.cs.colorado.edu/~alw/doc/Trends.html>>

■ **J.C. Derniame, B.A. Kaba, & D. Wastell** (eds.). *Software Process: Principles, Methodology and Technology*. Springer-Verlag, serie LNCS n° 1500, 1999.

■ **B. Westfechtel**. *Models and Tools for Managing Development Processes*. Springer-Verlag, serie LNCS n° 1646, 1999.

### Revistas

■ Journal of Software Process Improvement and Practice. Revista trimestral editada por Wiley. <<http://www.wiley.com/WileyCDA/WileyTitle/productCd-SPIP.html>>

### Congresos y Conferencias

■ EWSPT - European Workshop on Software Process Technology. Serie de talleres científicos con periodicidad bianual. La novena edición fue en Helsinki (Finlandia) en septiembre de 2003. Libro de actas: <<http://www2.springeronline.com/sgw/cda/frontpage/0,11855,5-40109-22-7063147-0,00.html>>

■ ICSP - International Conference on Software Process Improvement. La segunda edición de esta conferencia se celebró en Washington DC (EE.UU.) en junio de 2004. <<http://www.icspi.com/>>

■ PROFES - International Conference on Product Focused Software Process Improvement. Conferencia anual cuya quinta edición se celebró en Kansai Science City (Japón) en abril de 2004. <<http://www.vtt.fi/ele/profes2004/>>

### Sitios web

■ Capability Maturity Models. University of Carnegie Mellon, Software Engineering Institute (EE.UU.). <<http://www.sei.cmu.edu/cmm/cmms/cmms.html>>

■ Graphical Development Process Assistant. Universidad de Bremen (Alemania). <<http://www.informatik.uni-bremen.de/uniform/>

vm97/>. Sitio web especializado que incluye más de 6.000 páginas de información sobre este tema, incluyendo conceptos y definiciones, modelos de procesos, entornos, estándares, metodologías, elementos de procesos, clases y aproximaciones de modelado de PS, proyectos, herramientas y referencias bibliográficas clasificadas de publicaciones referidas con los puntos anteriores.

### Artículos

■ **V. Ambriola, R. Conradi, A. Fuggetta** (1997): Assessing Process-centered Software Engineering Environments. *ACM Transactions on Software Engineering and Methodology*, 6:3, 1997, pp. 283-328.

■ **G. Cugola, C. Ghezzi** (1998): Software Processes: a Retrospective and a Path to the Future. *Software Process: Improvement and Practice*, 4(3), 1998, pp. 101-123.

■ **B. Curtis, M.I. Kellner, J. Over** (1992): Process Modeling. *Communications of the ACM*, 35(2), September 1992, pp. 75-90.

■ **L. Osterweil** (1987): Software Processes are Software Too. Proc. of 9th International Conference on Software Engineering. Washington DC (USA). IEEE Comp. Soc. Press.



# Procesos Software: características, tecnología y entornos

Francisco Ruiz González<sup>1</sup>,  
Gerardo Canfora<sup>2</sup>

<sup>1</sup>Grupo Alarcos, Escuela Superior de Informática, Universidad de Castilla-La Mancha; <sup>2</sup>Research Centre on Software Technology (RCOST), Università degli Studi del Sannio (Benevento, Italia)

<Francisco.RuizG@uclm.es>,  
<canfora@unisannio.it>

## 1. Características de los Procesos Software

La definición de Proceso Software (PS) complementa el concepto de ciclo de vida en el sentido de que éste último define el esqueleto y la filosofía para llevar a cabo un PS, pero no es suficiente para guiar y controlar un proyecto de desarrollo y/o mantenimiento. Un PS es "un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software" [3].

La naturaleza especial de los PS está determinada por las siguientes características:

- Son complejos.
- No son procesos de producción típicos, ya que están dirigidos por excepciones, se ven muy determinados por circunstancias impredecibles y cada uno tiene peculiaridades que lo distinguen de los demás.
- Tampoco son procesos de ingeniería 'pura', ya que se desconocen las abstracciones adecuadas (no existe una ciencia experimental en la que apoyarse), dependen demasiado de demasiada gente, el diseño y la producción no están claramente diferenciados, y los presupuestos, calendarios y calidad no pueden ser planificados de forma suficientemente fiable.
- No son (completamente) procesos creativos, ya que algunas partes pueden ser descritas en detalle y algunos procedimientos son impuestos previamente.
- Están basados en descubrimientos que dependen de la comunicación, coordinación y cooperación dentro de marcos de trabajo predefinidos: los entregables generan nuevos requerimientos; los costes del cambio del software no suelen reconocerse; y el éxito depende de la implicación del usuario y de la coordinación de muchos roles (ventas, desarrollo técnico, cliente, etc.).

La necesidad de participación humana de forma creativa y la ausencia de acciones repetitivas hacen que ni el desarrollo ni el mantenimiento del software sean procesos de fabricación, pero existen algunas similitudes entre ambos tipos de procesos que son útiles para comprender los procesos software con una perspectiva más amplia. Al igual que los procesos de fabricación, los procesos software constan de dos sub-procesos interrelacionados: el proceso de producción y el proceso de gestión [5].

**Resumen:** en este artículo introductorio presentamos el concepto de Proceso Software (PS) y las propiedades que caracterizan y distinguen a estos procesos de otros tipos de procesos (por ejemplo, los típicos de producción industrial). A continuación justificamos el interés de disponer de una tecnología, conocida como Tecnología de Proceso Software (TPS), que permita automatizar e integrar los procesos de producción y de gestión en los proyectos software. Por último, presentamos las colecciones de herramientas integradas, conocidas como Entornos de Ingeniería del Software (EIS), cuyo objetivo es dar soporte a los citados procesos. Para concluir, se hace un resumen del problema de la integración de las herramientas componentes de un Entorno y de la manera de hacer que un Entorno sea orientado a procesos.

**Palabras clave:** Entorno de Integración de Herramientas, Ingeniería del Software, Orientación a Procesos, Proceso Software, Tecnología de Proceso Software.

## Autores

**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. Asimismo ha sido 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 la UCLM (www.alarcos.informatica.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 Geográfica), Ingeniería del Software para la edificación y bases de datos deductivas. Ha publicado numerosos y libros científicos sobre los temas citados y tiene 90 publicaciones en revistas y conferencias internacionales. Ha sido miembro de nueve comités de programa y de siete comités organizadores de congresos pertenecientes a diversas asociaciones científicas y profesionales: ACM, IEEE CS, AIL, AEC, AENOR, ISO/TC1/SC7, EASST, AENUI y ACTA.

**Gerardo Canfora** es Catedrático de Informática en la Facultad de Ingeniería y Director del Research Centre on Software Technology (RCOST) de la Università degli Studi del Sannio, en Benevento (Italia). Ha participado en los comités de programa y de organización de diversas conferencias internacionales. Ha sido co-presidente del comité de programa de "International Workshop on Program Comprehension" (IWPC 1997); "International Conference on Software Maintenance" (ICSM 2001); y "European Conference on Software Maintenance and Reengineering" (CSMR 2004). También ha sido presidente general del CSMR 2003. Sus intereses de investigación incluyen mantenimiento y evolución del software, comprensión de programas e ingeniería inversa, mejora del proceso software, gestión del conocimiento e ingeniería del software orientada a servicios. Sobre estos temas ha publicado más de 100 artículos en revistas y conferencias internacionales. Es editor asociado del "IEEE Transactions on Software Engineering" y participa en el consejo editorial del "Journal of Software Maintenance and Evolution: Research and Practice".

El proceso de producción se relaciona con la producción y el mantenimiento del producto propiamente dichos, mientras que el proceso de gestión proporciona los recursos necesarios para el proceso de producción y lo controla. Esto último es posible si el proceso de producción devuelve información al proceso de gestión sobre su comportamiento.

Estas relaciones están representadas en la figura 1, donde también se muestran las relaciones entre el proceso y el entorno externo: la petición sobre el producto ha de ser llevada a cabo desde el mundo exterior, es decir el entorno exterior es quien justifica la existencia del proceso de producción. Además la gestión tiene que cumplir con los

estándares actuales que existen en el entorno; es decir, el entorno exterior influye también indirectamente en el proceso de producción. Finalmente, los procesos de producción y de gestión explotan tecnologías que también vienen del entorno.

## 2. Tecnología de Proceso Software

La esencia de la Tecnología de Proceso Software (TPS) es que permite la integración de tecnologías de producción y de gestión en un nuevo entorno de trabajo, conocido como "Entorno de Ingeniería del Software orientado al Proceso" (PSEE, *Process-centered Software Engineering Environment*), que da soporte a los procesos de gestión y de producción de forma integrada. La figura 2

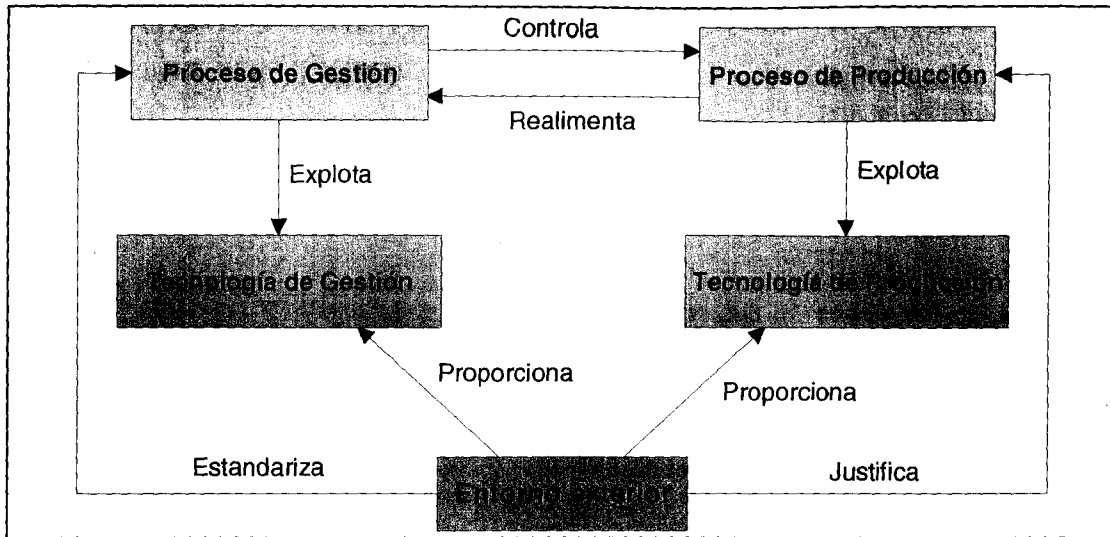


Figura 1. Proceso de producción vs proceso de gestión.

muestra el impacto de esta nueva tecnología, mostrando como el PSEE implementa, controla y mejora los flujos de información con los cuales el proceso de gestión controla al proceso de producción. El objetivo principal de la TPS es dominar la complejidad inherente al PS mediante una comprensión profunda del proceso en sí mismo y mediante un soporte automatizado por medio de un PSEE.

Un aspecto fundamental para lograr el citado objetivo es el soporte computerizado del proceso; es decir, la disponibilidad de un modelo de procesos y los medios adecuados para definirlo, modificarlo, analizarlo y realizarlo [1]. Consecuentemente con la definición anterior,

la TPS saca provecho de diversas áreas y conceptos:

1. **Tecnologías** de desarrollo y mantenimiento de software, que aportan las herramientas e infraestructuras necesarias para hacer posible --y económicamente factible-- crear y mantener productos software complejos que satisfagan las necesidades actuales y futuras.
2. **Métodos y técnicas** para el desarrollo y mantenimiento de software, que suponen el soporte metodológico esencial para aprovechar de manera eficiente las tecnologías y realizar con éxito las actividades de desarrollo y mantenimiento del software.
3. **Comportamiento organizacional**, es decir, la ciencia de las organizaciones y las perso-

nas es útil en TPS porque, en general, los proyectos software se llevan a cabo por equipos de personas que tienen que ser coordinados y dirigidos dentro de una estructura organizacional eficiente.

4. **Marketing y economía**, ya que los proyectos de desarrollo y mantenimiento de software no son esfuerzos autónomos sino que, como pasa con cualquier otro producto, el software debe estar dirigido a satisfacer las necesidades de clientes/usuarios reales.

En conclusión, al desarrollar o mantener software es necesario prestar atención a la compleja interrelación que se produce entre los factores organizacionales, culturales, tecnológicos y económicos.

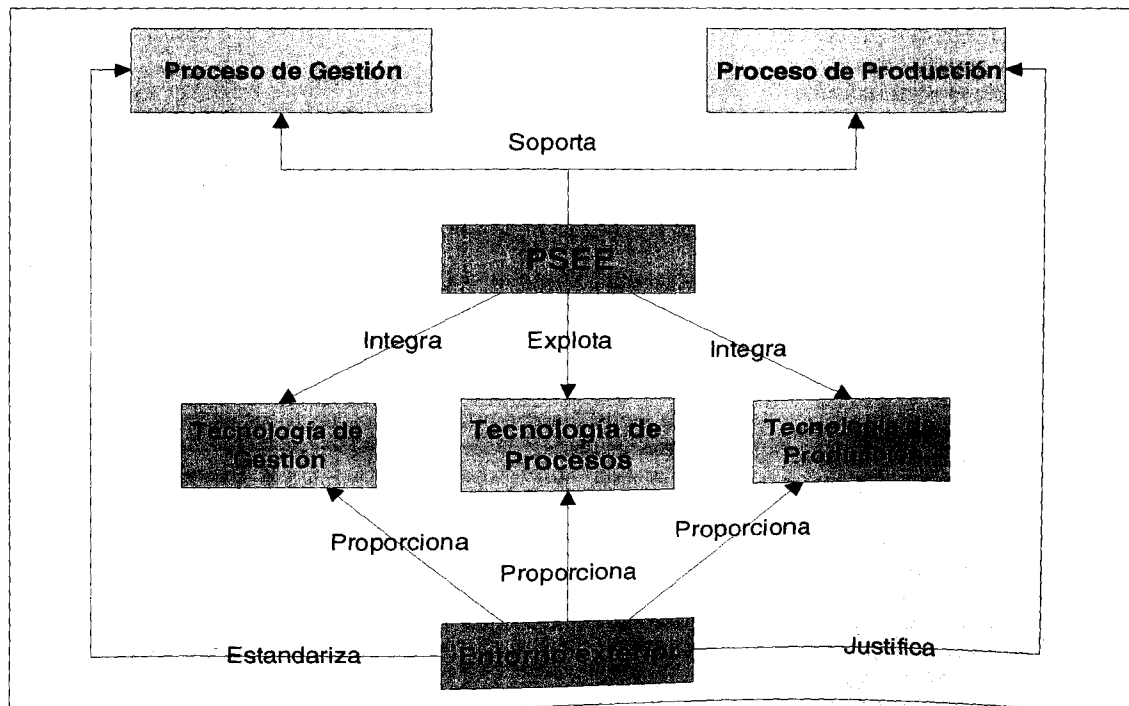


Figura 2. Impacto de la Tecnología de Proceso Software.



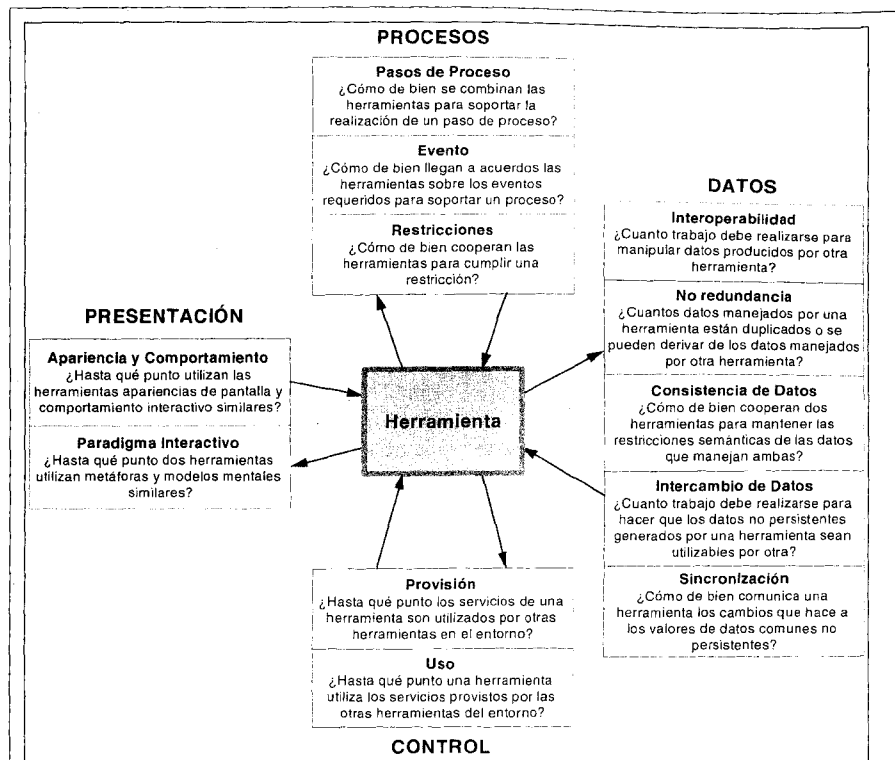


Figura 3. Propiedades de la Integración de Herramientas en un EIS.

la calidad y la productividad basadas en la disciplina, la gestión, y la ayuda de EIS y otras tecnologías informáticas.

Muchas organizaciones tienen problemas para definir y realizar los pasos que transforman las necesidades del usuario en un producto software, de manera que sean repetibles, medibles con respecto a su impacto en los objetivos de calidad y adaptables o mejorables. Por tanto, la ayuda de un EIS para implantar un proceso definido durante la realización de un proyecto software puede proporcionar beneficios sustanciales a corto plazo.

Dentro de un EIS, los servicios de gestión de procesos contribuyen a este soporte eficaz de los PS proveyendo facilidades orientadas al usuario final para definir y utilizar procesos que pueden reemplazar la invocación indiscriminada, difícil de controlar, y tediosa de herramientas individuales. Gargy Jazayeri [4] han considerado que el soporte a procesos en un EIS está basado en las siguientes funcionalidades:

- **Definición de procesos.** Los ingenieros software utilizan el EIS para definir un proceso de cara a su utilización en uno o más proyectos.
- **Análisis de procesos.** Dentro del EIS, un modelo de proceso puede ser analizado para verificar su consistencia, completitud y corrección.
- **Presentación de procesos.** El EIS incluye soporte para la visualización gráfica de los PS (flujos de actividades) y de los productos (diagramas estructurados).

- **Simulación de procesos.** El EIS soporta el uso de simulaciones para poder evaluar la idoneidad de un proceso antes de consumir recursos en su realización real.
- **Automatización de procesos.** Una vez un proceso ha sido definido, las actividades que no requieren intervención humana pueden ser identificadas y automatizadas por el EIS.
- **Supervisión de procesos.** El EIS supervisa la realización de un proceso y registra la historia de las actividades llevadas a cabo. Esta historia del proceso puede utilizarse después para futuros desarrollos de nuevos procesos o mejora del existente.
- **Soporte de cambios en procesos.** El EIS permite que una organización cambie sus definiciones de procesos sin tener que interrumpir el trabajo.
- **Apertura.** El EIS provee herramientas para intercambiar datos y metadatos con herramientas no integradas o con otros EIS.
- **Soporte multiusuario.** Habitualmente, los proyectos de Ingeniería del Software son realizados por grupos de personas con diferentes roles, por tanto, el EIS debe dar servicio a todas las personas que trabajan juntas en un proceso.
- **Dirección de procesos.** Los ingenieros software utilizan el EIS para llevar a cabo diferentes etapas de un proceso. El EIS debe ofrecer ayuda para elegir las etapas siguientes en base al modelo del proceso y del estado actual.
- **Interfaz de usuario específico para cada tarea.** Basándose en el modelo del proceso, el EIS puede adaptar el interfaz de usuario a las necesidades de cada tarea y así evitar un exceso

de información presentada al usuario. Cada vez es más frecuente que el desarrollo de mantenimiento de un producto software realizado con la colaboración de varias entidades con organizaciones. Por ello, en los últimos años ha cobrado auge el estudio de los problemas que surgen cuando se desea que varios PSEEs diferentes y separados colaboren más concretamente, que exista interoperabilidad entre los procesos que soportan. Entre las diversas propuestas formuladas para dar este problema, destacan las federaciones de PSEEs. En esta línea, algunos autores han propuesto usar la metáfora de la alianza internacional, donde cada organización gestiona sus propios procesos (igual que cada país sus leyes) y los procesos inter-organizacionales actúan de forma semejante al de los tratados entre países.

En la bibliografía se han propuesto dos tipos de arquitecturas conceptuales para federaciones de PSEEs: basadas en el control, favorecen la centralización al existir muchos procesos comunes; y las basadas en el estado, que disponen de un espacio de trabajo donde se almacena el estado común.

#### 4. Conclusiones

En este artículo hemos presentado los principales aspectos de la Tecnología de Proceso Software: el objeto de atención (los procesos software y sus características); el interés de justificación de dar soporte automático a estos procesos; y los requisitos, funcionalidad y características de integración y de orientación a procesos que deben tener las colecciones de herramientas para satisfacer estas necesidades.

#### Referencias

- [1] J.C. Derniame, B.A. Kaba, D. Wastell, (eds.) *Software Process: Principles, Methodology and Technology*. LNCS 1500, Springer-Verlag, 1999.
- [2] J. Estublier, P.Y. Cunin, N. Belkhatir. *Architectures for Process Support System Interoperability*. Proceedings of the *Fifth International Conference on the Software Process (ICSP'98)*, 15-17 Junio, Chicago (Estados Unidos), pp. 137-147, 1998.
- [3] A. Fuggetta. *Software Process: A Roadmap*. 22nd International Conference on Software Engineering (ICSE'2000), *Future of Software Engineering Track*, June 4-11, Limerick (Irlanda), ACM, 2000.
- [4] P.K. Garg, M. Jazayeri. *Process-centered Software Engineering Environments: A Grand Tour*. En A. Fuggetta, A. Wolf (eds.); *Software Process*. John Wiley & Sons, 1996.
- [5] R. McLeod Jr. *Management Information Systems*. McMillan Publishing, New York, 1990.
- [6] H. Ossher, W. Harrison, P. Tarr. *Software Engineering Tools and Environments: a Roadmap*. International Conference on Software Engineering (ICSE) - *Future of SE Track*. Limerick (Irlanda), pp. 261-277, 2000.
- [7] I. Thomas, B.A. Nejme. *Definitions of Tool Integration for Environments*. *IEEE Software*, 9(2), pp. 29-35, 1992.
- [8] M.V. Zelkowitz. *Software Engineering Environment Capabilities*. *Journal of Systems and Software*. Elsevier Science, 35(1), pp. 3-14, 1996.

