

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática). **Novática** edita también **Upgrade**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (UPGRADE European Network)

<<http://www.ati.es/novatica/>>
 <<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **Ai2** y **ASTIC**.

CONSEJO EDITORIAL

Antoni Carbonell Nogueras, Francisco López Crespo, Julián Marcelo Cocho, Celestino Martín Alonso, Josep Molas i Bertrán, Roberto Moya Quiles, César Pérez Chirinos, Mario Piattini Velthuis, Fernando Píera Gómez (Presidente del Consejo), Miquel Sarries Griñó, Asunción Yturbe Herranz

Coordinación Editorial

Rafael Fernández Calvo <rfcalvo@ati.es>

Composición y autoedición

Jorge López

Traducciones

Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>

Administración

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

SECCIONES TÉCNICAS: COORDINADORES

Administración Pública electrónica

Gumerindo García Arribas, Francisco López Crespo (MAP)

<gumerindo.garcia@map.es>, <tlc@ati.es>

Argumentos

Jordi Tubella (DAC-UPC) <jordit@ac.upc.es>

Victor Vihales Yufera (Univ. de Zaragoza) <victor@unizar.es>

Audiencia 67%

Maria Tourino, Manuel Palao (ASIA) <manuel@palao.com>

Basas de datos

Coral Calero Muñoz, Mario G. Piattini Velthuis

(Escuela Superior de Informática, UCLM)

<Coral.Calero@uclm.es>, <mpiattini@inf-cr.uclm.es>

Derivados e Internet

Isabel Hernando Collazas (Fac. Derecho de Donostia, UPV) <ihernando@legalket.net>

Isabel Davara Fernández de Marcos (Davara & Davara) <idavara@davara.com>

Escuela Universitaria de la Informática

Josquin Epeleta Marín (OPS-UZAR) <ezepeleta@posta.unizar.es>

Cristóbal Pareja Flores (DSIP-UCM) <cpareja@sis.ucm.es>

Gestión del Conocimiento

Juan Baiget Solé (Cap Gemini Ernst & Young) <juan.baiget@ati.es>

Informática y Filosofía

Josep Corco (UIC) <jcorco@unica.edu>

Esperanza Marcos (ESCET-URJC) <cuca@escet.urjc.es>

Informática básica

Miquel Chover Selles (Universitat Jaume I de Castellón) <chover@lsi.uji.es>

Roberto Vivó (Eurographics, sección española) <rvivo@dsic.upv.es>

Ingeniería del Software

Javier Dolado Costa (Cap Gemini EPV) <dolado@si.ehu.es>

Luis Fernández (PRIS-EI-UEM) <lufern@dpri.ssi.uem.es>

Inteligencia Artificial

Federico Barber, Vicente Botti (DSIC-UPV)

<fvbotti_fbarber@dsic.upv.es>

Interacción Persona-Computador

Julio Abascal González (FI-UPV) <julio@si.ehu.es>

Jesús Luis Vidal (Univ. de Lleida) <jesus@eup.udl.es>

Internet

Alonso Álvarez García (TID) <alonso@ati.es>

Llorenç Panés Casas (Iindra) <pages@ati.es>

Lengua e Informática

M. del Carmen Ugarte (IBM) <cugarte@ati.es>

Lenguajes Informáticos

Andrés Martín López (Univ. Carlos III) <amarin@it.uc3m.es>

J. Angel Velázquez (ESCET-URJC) <a.velazquez@escet.urjc.es>

Librerías e Informática

Alfonso Escobedo (FIH-Univ. de La Laguna) <aescoban@ull.es>

Lingüística computacional

Xavier Gómez Guinovart (Univ. de Vigo) <xgg@uvigo.es>

Manuel Palomar (Univ. de Alicante) <mpalomar@lssi.ua.es>

Mundo estudiantil

Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM)

<a.vazquez@ieee.org>

Profesión Informática

Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>

Miquel Sarries Griñó (Ayto. de Barcelona) <msarries@ati.es>

Redes y servicios telemáticos

Luis Sauraro Coloma (DCOM-UPV) <luisaur@com.upv.es>

Josep Salé Parela (DAC-UPC) <parela@ac.upc.es>

Seguridad

Javier Arellito Bertolin (Univ. de Deusto) <jarellito@eside.deusto.es>

Javier López Muñoz (ETSI Informática-UMA) <jlm@lcc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso, Juan Antonio de la Puente

(DI-UPM) <{alalonso, juapente}@di.upm.es>

Software Libre

Jesús M. González Barahona, Pedro de las Heras Quirós

(GSYC-URJC) <{jog, pheras}@gsyc.esct.urjc.es>

Tecnología de Objetos

Jesús García Molina (DIS-UM) <jmolina@correo.um.es>

Gustavo Rossi (LFIH-UNEP, Argentina) <gustavo@sol.info.unlp.edu.ar>

Tecnología para la Educación

Juan Manuel Dodero Beardo (UC3M) <dodero@inf.uc3m.es>

Francisco Riviere (PamCAT) <friviere@wanadoo.es>

Tecnología y Empresa

Pablo Hernández Medrano (BlueMAT) <pablohm@bluemat.biz>

TIC para la Sanidad

Valentín Masero Vargas (DI-UNEX) <vmasero@unex.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga)

<{aguayo, guevara}@lcc.uma.es>

Las opiniones expresadas por los autores son responsabilidad exclusiva de los mismos.

Novática permite la reproducción de todos los artículos, salvo los marcados con © o *copyright*, debiéndose en todo caso citar su procedencia y enviar a **Novática** un ejemplar de la publicación.

Coordinación Editorial, Redacción Central y Redacción ATI Madrid

Padilla 66, 3º dcha., 28006 Madrid

Tel. 914029391; fax 913093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia

Av. del Reino de Valencia 23, 46005 Valencia

Tel./fax 963330392 <secreval@ati.es>

Administración y Redacción ATI Cataluña

Via Laietana 41, 1º 08003 Barcelona

Tel. 934125235; fax 934127713 <secretgen@ati.es>

Redacción ATI Andalucía

Isaac Newton, s/n, Ed. Sadiel,

Isla Cartuja 41092 Sevilla, Tel./fax 954460779 <secretand@ati.es>

Redacción ATI Aragón

Lagasca 9, 3-B, 50006 Zaragoza,

Tel./fax 976235111 <secretara@ati.es>

Redacción ATI Asturias-Cantabria

<gg-astucant@ati.es>

Redacción ATI Castilla-La Mancha

<gg-clmancha@ati.es>

Redacción ATI Galicia

Recinto Ferial s/n, 36540 Silleda (Pontevedra)

Tel. 986581413; fax 986580162 <secretgal@ati.es>

Subscripción y Ventas

<<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid

Publicidad

Padilla 66, 3º dcha., 28006 Madrid

Tel. 914029391; fax 913093685 <novatica.publicidad@ati.es>

Imprenta

9 Imprime S.A. Juan de Austria 66, 08005 Barcelona.

Dedicado legal: B.15.154-1975 -- ISSN. 0211-2124. CODEN NOVACB

Portada: Antonio Crespo Folx / © ATI 2004

Diseño: Fernando Agresta / © ATI 2004

en resumen

TPS o el software como proceso > 02
 Rafael Fernández Calvo

monografía

Tecnología de Proceso Software

(En colaboración con **Upgrade**)

Editores invitados: *Francisco Ruiz González, Gerardo Canfora*

Presentación. La Tecnología de Proceso Software y la mejora de la gestión de los proyectos y de la calidad de los productos > 03

Francisco Ruiz González, Gerardo Canfora

Procesos Software: características, tecnología y entornos > 05

Francisco Ruiz González, Gerardo Canfora

Cuestiones clave y nuevos retos en la Tecnología de Proceso Software > 09

Jean-Claude Derniame, Flavio Oquendo

Una taxonomía de los Servicios de Entornos de Ingeniería de Software: la futura norma ISO/IEC 15940 > 14

Dan Hyung Lee, Juan Garbajosa Sopena

Software libre y de código abierto: ¿un nuevo modelo para el desarrollo de software? > 18

Alfonso Fuggetta

Aplicación de los principios básicos de la Ingeniería de Modelos al campo de la Ingeniería de Procesos > 22

Jean Bézivin, Erwan Breton

Lenguajes de Modelización de Procesos de Software basados en UML > 27

Pere Botella i López, Xavier Franch Gutiérrez, Josep M. Ribó Balust

Soporte a los Procesos Software en un Entorno de Ingeniería del Software orientado a Procesos > 32

Hans-Ulrich Kobiak

Gestión de proyectos distribuidos con GENESIS > 38

Lerina Aversano, Andrea De Lucia, Matteo Gaeta,

Pierluigi Ritrovato, Maria Luisa Villani

Medición de los Procesos Software > 43

Félix García Rubio, Francisco Ruiz González, Mario Piattini Velthuis

contribución invitada

Más allá de Internet: la Red Universal Digital

> 48

Fernando Sáez Vacas

secciones técnicas

Administración Pública electrónica

Agència Catalana de Certificació: la certificación digital en la Administración Pública catalana

> 52

Josep Llopis Méndez

Ingeniería del Software

Diseño basado en componentes: alternativas en la etapa de partición > 55

Arantza Irastorza Goñi, Arturo Jaime Elizondo, Oscar Díaz García

Desarrollo de Sistemas de Inspección Visual Automatizada a partir de la descripción de un Patrón Arquitectural Genérico

> 63

Cristina Vicente Chicote, Carlos Fernández Andrés, Pedro Sánchez Palma

Profesión informática

TIC y Sociedad de la Información: propuestas para una nueva etapa > 66

Josep Molas i Bertrán

Referencias autorizadas > 68

sociedad de la información

Personal y transferible

Protección de datos personales y Seguridad del Estado

> 73

Ofelia Tejerina Rodríguez

programar es crear

Diseño de suelos (CUPCAM 2003, problema G, solución)

> 74

Juan Céspedes Prieto, Antonio Fernández Anta, Ángel Herranz Nieva

asuntos interiores

Coordinación editorial / Programación de Novática

> 76

Normas de publicación para autores / Socios Institucionales

> 77

Monografía del próximo número: "Criptografía"

Procesos Software: características, tecnología y entornos

Francisco Ruiz González¹,
Gerardo Canfora²

¹ Grupo Alarcos, Escuela Superior de Informática, Universidad de Castilla-La Mancha; ² 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. 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.infcr.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, 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”.

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**

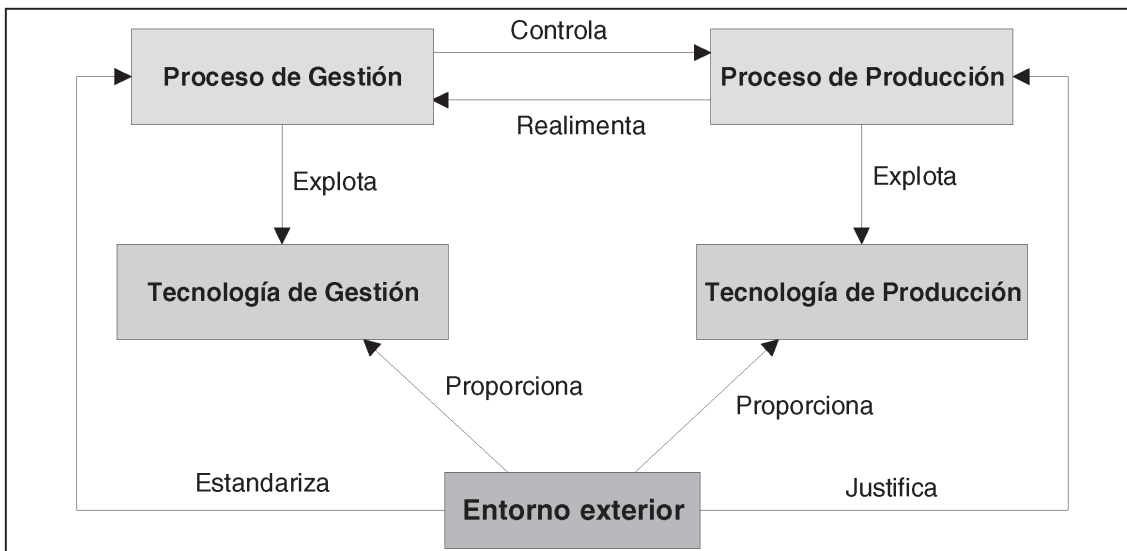


Figura 1. Proceso de producción vs proceso de gestión.

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.

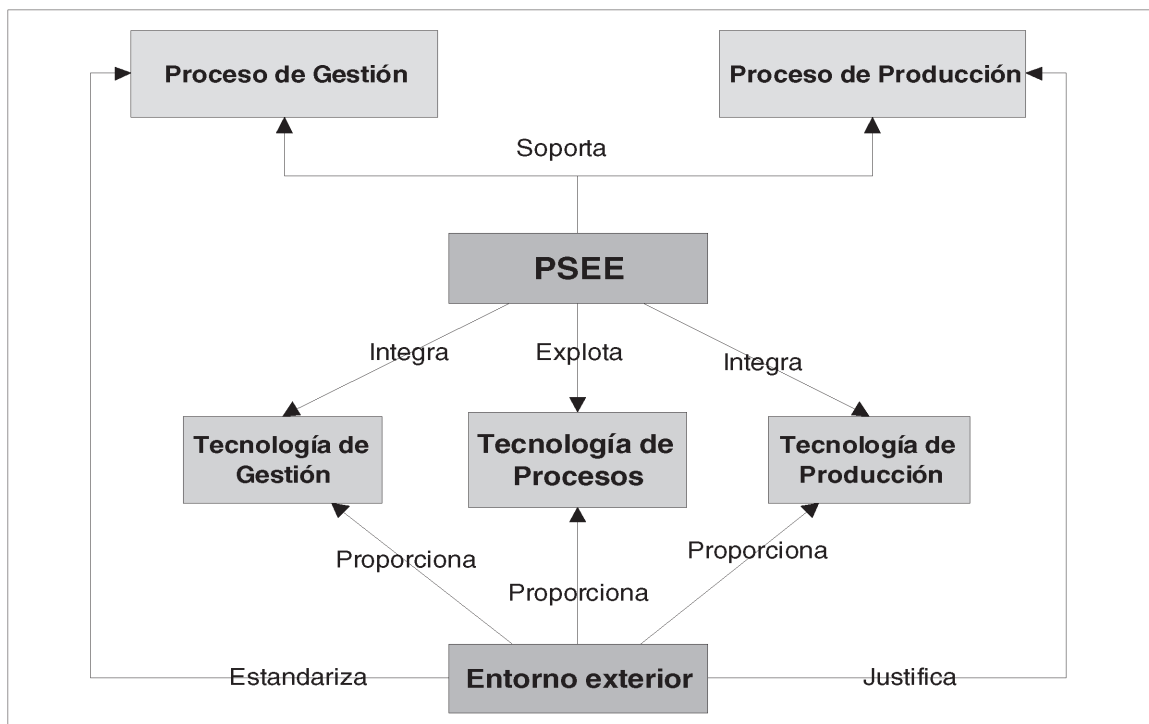


Figura 2. Impacto de la Tecnología de Proceso Software.



El concepto de Entorno de Ingeniería del Software (EIS) es bastante reciente



3. Entornos de Ingeniería del Software

Aunque el uso de herramientas para ayudar a los desarrolladores en la producción de software ha existido, de una u otra manera, desde los días iniciales de la Informática, el concepto de *Software Engineering Environment* (SEE), es decir, Entorno de Ingeniería del Software (EIS), es bastante reciente. Un EIS se define como "una colección de herramientas que proporcionan un soporte automático, parcial o total, a las actividades de Ingeniería del Software". Habitualmente, estas actividades se llevan a cabo en el marco de un proyecto software y se refieren a aspectos tales como la especificación, desarrollo, reingeniería o mantenimiento de sistemas software.

Los EIS también se han conocido con otros nombres: IPSE (*Integrated Project Support Environment*), ISEE (*Integrated Software Engineering Environment*), Coalición de herramientas CASE, Herramientas CASE federadas, o ISF (*Integrated Software Factory*).

El término EIS se puede aplicar a sistemas de un alcance muy diferente: desde un conjunto de unas pocas herramientas ejecutándose sobre el mismo sistema, hasta un entorno totalmente integrado capaz de gestionar y controlar todos los datos, procesos y actividades del ciclo de vida de un producto software.

Gracias a la automatización de actividades (de forma parcial o total), un EIS puede aportar importantes beneficios a una organización: reducción de costes (alta productividad), mejora en la gestión y mayor calidad en el producto final. Por ejemplo, la automatización de actividades repetitivas -- como la ejecución de casos de prueba -- no sólo mejora la productividad, sino que también ayuda a asegurar la 'completitud' y consistencia de las actividades de prueba.

Habitualmente, un EIS maneja información relacionada con:

- El software en desarrollo o mantenimiento (especificaciones, datos de diseño, código fuente, datos de pruebas, planes de proyecto, ...);
- Los recursos del proyecto (costes, recursos informáticos, personal, responsabilidades y obligaciones, ...); y
- Los aspectos organizacionales (políticas de la organización, estándares y

metodologías empleados, ...).

Un EIS da soporte a actividades humanas mediante una serie de **servicios** que describen las capacidades del entorno. Los servicios proporcionan una correspondencia entre un conjunto de procesos escogidos, relativos al ciclo de vida del software, y su automatización mediante el uso de herramientas. En la mayoría de los casos la funcionalidad de una herramienta está relacionada con uno o más servicios.

El interés en los EIS comenzó a principios de los años 90, cuando surgieron las primeras propuestas de modelos de referencia y se propusieron las primeras clasificaciones de los servicios que deberían incluirse [8]. Pero ha sido a comienzos del siglo XXI cuando se han desarrollado entornos que realmente intentan satisfacer los ambiciosos objetivos que implica la definición de EIS dada anteriormente [6].

3.1. Integración

El concepto que más diferencia un EIS de un simple conjunto de herramientas ejecutándose en una computadora bajo un mismo sistema operativo es el grado de integración que provee. El concepto de integración aplicado a un EIS puede significar varias cosas relacionadas pero diferentes:

- El grado en que diferentes herramientas pueden comunicar eficazmente entre sí dentro del marco de trabajo del EIS.
- Una medida de las relaciones entre los componentes de un EIS.
- La facilidad, interoperabilidad, portabilidad, escalabilidad, productividad, etc., producida por la interacción "sin parches" entre un conjunto de componentes de un EIS.

Compartir un mismo sistema de gestión de objetos (gestor del repositorio) en vez de un sistema de ficheros separado para cada herramienta es un aspecto importante de la integración, pero no es el único. Un EIS debe disponer de un conjunto de interfaces que permitan la cooperación entre herramientas de fabricantes diversos. Por esta razón, la integración implica los tres aspectos siguientes:

- **Un conjunto de servicios.** Muchos de los servicios que se describen en apartados posteriores son aplicables a la integración. Por ejemplo, utilizar un sistema de gestión de objetos común con esquemas comunes permite que las herramientas compartan objetos; utilizar características de presentación globales en el interfaz de usuario permite

disponer de un "aspecto de visualización" similar en todas las herramientas; o los servicios de gestión de procesos y de comunicación son necesarios para que las herramientas puedan comunicarse unas con otras.

- **Una nueva dimensión para cada servicio.** Tener servicios comunes permite pero no obliga a la integración (los constructores de herramientas no están obligados a utilizarlos). Esta nueva dimensión indica el grado en que un servicio puede contribuir a aumentar la integración.

- **Una política.** También se requiere implantar políticas para que los constructores de las herramientas, marcos de trabajo y plataformas utilicen los servicios de integración eficientemente. Un ejemplo de esto son las "guías de estilo" para constructores de herramientas.

Según Thomas y Nejme [7], la necesidad de integración en un EIS abarca varias dimensiones diferentes (**figura 3**):

- **Datos.** La integración de los datos es la capacidad de compartir la información dentro del EIS. El grado de integración de datos puede ser alto (las herramientas usan una base de datos común con un esquema común), mediano (formatos de datos comunes) o bajo (utilizar mecanismos de traducción). Otra característica que la integración de datos puede incluir es la composición.
- **Control.** La integración del control es la capacidad de combinar las funcionalidades ofrecidas en un entorno de forma flexible. Las combinaciones pueden corresponder a preferencias de un proyecto y estar dirigidas por los procesos software subyacentes.
- **Presentación.** La integración de la presentación es la capacidad de interactuar con las funcionalidades del entorno mediante pantallas de apariencia similar y modos de interacción similares.
- **Procesos.** La integración de procesos es la capacidad de acceder a las funcionalidades del entorno utilizando un PS predefinido que está siendo realizado con soporte automático.

3.2. Orientación a procesos

Ya se ha comentado la importancia que los EIS orientados a procesos (o PSEE) tienen en la TPS. De hecho, el principal papel de un EIS es dar soporte para llevar a cabo los PS de forma eficaz. Este punto de vista está ganando peso porque los procesos de desarrollo y mantenimiento de software se han convertido, cada vez más, en actividades complejas y laboriosas de carácter intelectual, con un alto potencial para la mejoras en

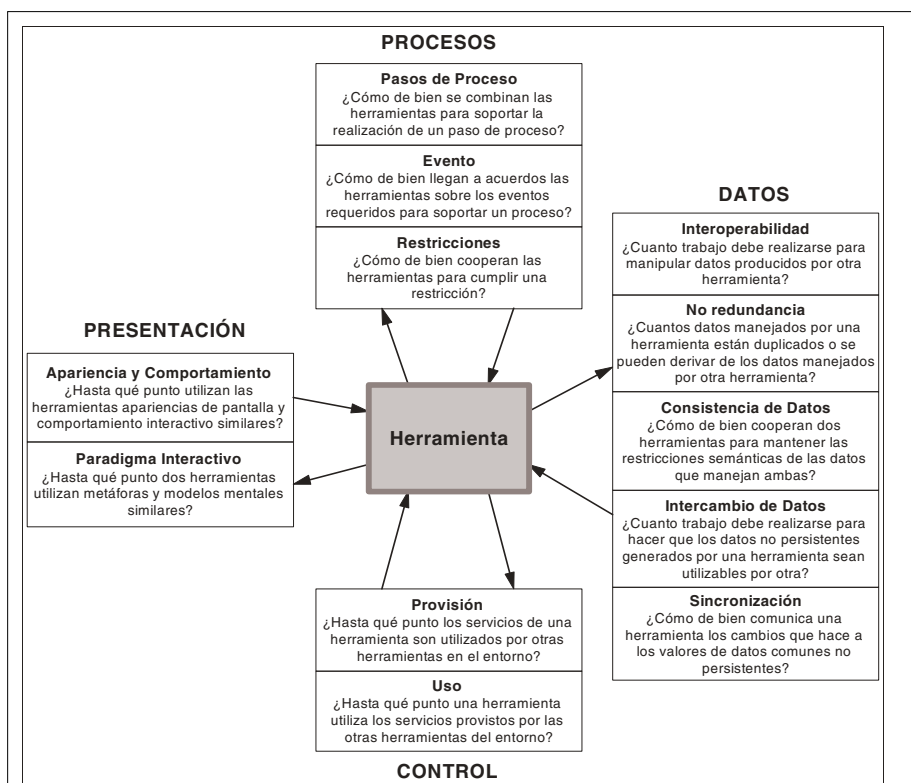


Figura 3. Propiedades de la Integración de Herramientas en un EIS.

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 proporcionando facilidades orientadas al usuario final para definir y utilizar procesos que pueden reemplazar la invocación indisciplinada, difícil de controlar, y tediosa de herramientas individuales. Garg y 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 y mantenimiento de un producto software sea realizado con la colaboración de varias empresas u 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, y, más concretamente, que exista interoperabilidad entre los procesos que soportan. Entre las diversas propuestas formuladas para abordar 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 tiene 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, que favorecen la centralización al existir modelos de procesos comunes; y las basadas en el estado, que disponen de un espacio de trabajo donde se almacena el estado común [2].

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 y 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.