

## MODELO DE CUADRO DE MANDO PARA FACTORÍAS SOFTWARE DEL SECTOR FINANCIERO

César Álvarez Pérez

*Caja Rural de Asturias*

Vicente Rodríguez Montequín

Carlos Alba González-Fanjul

Luis Fernández Álvarez

*Área de Proyectos de Ingeniería, Universidad de Oviedo*

### Abstract 1

This paper aims to search for KPI's, Key Performance Indicators which permit to measure productivity in terms of performance and quality of the generation of Financial Software in a Software Factory, which means the workplace where software is developed by using principles and techniques associated to traditional industrial production.

A set of new indicators is presented, which together with the already used and more traditional ones, allow us to assess productivity of individual projects as well as global productivity for Software Factory itself.

The different key indicators are conveniently set in a scorecard that allows decision-making associated with productivity and quality of the project portfolio.

The collecting of values and data measured by the different indicators, together with the information supplied from scorecards, will also allow the assessment and comparison in terms of productivity between Software Factories and the other different structures used in the software generation processes.

**Keywords:** *Software Factory; Scorecard; Key Performance Indicator; Financial Software;*

### Resumen 1

El trabajo tiene por objetivo la búsqueda de indicadores clave del desempeño KPI's, Key Performance Indicators, que permitan medir la productividad, en términos de rendimiento y calidad, de la generación de Software Financiero en una Fábrica de Software, o Software Factory, entendiendo por ella el centro de trabajo donde se desarrolla software aplicando técnicas y principios ligados a la producción industrial tradicional.

Se presenta un conjunto de nuevos indicadores, que junto a los tradicionales ya usados, permiten evaluar tanto la productividad de proyectos individuales como la global para la propia Software Factory.

Los distintos indicadores clave, se agrupan convenientemente en un cuadro de mando que permite la toma de decisiones relacionadas con rendimiento y calidad del portafolio de proyectos.

La toma de datos y valores medidos por los distintos indicadores, junto con la información proporcionada de los cuadros de mando ideados, permitirá además la evaluación y comparación en términos de productividad entre las Software Factory y otras estructuras diferentes usadas en los procesos de generación de software.

**Palabras clave:** *factoría software; cuadro de mando; indicador clave del desempeño; software financiero;*

## 1. Introducción

Las empresas del sector financiero, y en especial las entidades bancarias, se han caracterizado históricamente por la inversión en políticas de organización y gestión que han permitido la mejora de su eficiencia. El sector financiero y en especial el bancario, es uno de los sectores que más se esfuerza en la medición de la productividad de las distintas áreas que forman su negocio, pero mantiene un punto débil en la medición y seguimiento de la productividad de los desarrollos software que soportan su actividad («Medir la productividad del desarrollo de software en Banca», s. f.).

El sector bancario es uno de los que más desarrollo software requiere. Su coste cada vez es más importante y tiene un mayor peso en la cuenta de resultados de las distintas entidades financieras.

En la mayoría de las situaciones el desarrollo software está externalizado, bien con las grandes consultoras internacionales o bien con empresas creadas por las propias entidades financieras y a las que éstas han traspasado el desarrollo, mantenimiento y negocio ligado al desarrollo de software. Debido a la total dependencia para el mantenimiento del negocio que las entidades financieras tienen de las empresas que les desarrollan su software, éstas han preferido optar por ceder el desarrollo a empresas creadas y controladas por las propias entidades financieras, al menos en las partes de desarrollo del denominado *Core Bancario*, que engloba el soporte de clientes, productos de activo y pasivo, tesorería, contabilidad y en definitiva todas aquellas actividades vitales para el mantenimiento del negocio. ISBAN, del grupo Banco Santander y RSI, del grupo Caja Rural, son dos ejemplos de ello.

En los últimos años, y fruto de esta última crisis económica que aún se prolonga en el tiempo, las entidades financieras presionan a sus empresas suministradoras de software para abaratar sus servicios, lo que les obliga a la búsqueda de nuevas fórmulas y modelos productivos que las haga más competitivas. La mayoría de las soluciones actuales pasan por la industrialización del desarrollo de software y la búsqueda de mano de obra más barata, pero deberían incluirse también procesos o soluciones que incentivasen el aumento de la productividad de manera que el coste del proyecto o los desarrollos estén ligados a lo producido (y al tiempo teórico estimado y necesario para su producción) y no al tiempo dedicado a su producción.

## 2. Las factorías software en el sector financiero

Las entidades financieras llevan años subcontratando y externalizando la mayoría de los servicios que necesitan para el mantenimiento de su negocio, incluido el desarrollo y mantenimiento de sus sistemas informáticos que sirven de soporte a su actividad.

Al igual que sucede con otros servicios subcontratados, la externalización del desarrollo de software por un proveedor, permite la obtención del producto software deseado mediante las especificaciones y niveles de calidad pactados en el contrato del servicio.

Las principales entidades financieras, han optado por crear sus propias empresas informáticas (o bien comprar aquellas que ya les suministraban su sistema informático) a las que han traspasado toda la actividad relacionada con el desarrollo de software y de mantenimiento de sus sistemas informáticos. De esta manera se logra externalizar el servicio y al mismo tiempo se aseguran el control de las empresas que desarrollan y mantienen su informática mediante la integración de la misma en su grupo empresarial. Las entidades financieras no pueden asumir el riesgo operativo que provocaría la desaparición o cierre de las empresas informáticas que sostienen su negocio, y por eso se decantan por externalizar el servicio sin perder su control. A partir de este momento, las entidades financieras pueden centrarse en las funciones esenciales de su negocio despreocupándose de la informática.

## 2.1 Definición de Factoría Software

Si bien distintos autores definen de manera distinta el concepto de Factoría Software, (Software Factory en inglés), de manera general podemos entender por Factoría Software el centro de trabajo donde se desarrolla software aplicando técnicas y principios ligados a la producción industrial tradicional.

Aunque el término de Factoría Software se utilizó por primera vez en el año 1968 y se considera a Hitachi Software Works como la primera Factoría Software (1669), es en estos últimos años y ante las especiales condiciones socio económicas, tecnológicas y de madurez de la ingeniería del software, cuando el término Factoría Software vuelve a cobrar notoriedad en la industria del software. (Garzás & Piattini, 2007)

Así (Greenfield, Short, Cook, Kent, & Crupi, 2004) de Microsoft usan el concepto de Factoría Software como el enfoque de desarrollo de aplicaciones en el que confluye el desarrollo basado en componentes, el desarrollo dirigido por modelos y las líneas de producto software. Otros como (Correa, Werner, & Zaverucha, 2000) definen Factoría Software (FS) en base a la siguiente ecuación:

$$FS = (\text{Especificaciones de Gestión, Líneas de Producto}) \times (\text{Procesos, Personas, Técnicas}),$$

donde se combina la gestión de la calidad orientada a procesos, con el punto de vista técnico de las líneas de producto basadas en tecnologías de componentes (Garzás & Piattini, 2007). Esta propuesta integra ISO9000, CMM y PSP/TSP (Mario Piattini, Félix Oscar García, & Ismael Caballero, 2006).

El objetivo de una SF es la mejora de la productividad, aumento de la calidad, producción de escala y el mantenimiento del control del desarrollo de software (Siqueira, Barbaran, & Becerra, 2008).

La industria del software está evolucionando hacia un modelo de negocio basado en las SF, donde normalmente colaboran varios centros en la consecución de objetivos y desarrollos. La utilización de equipos especializados y ubicados en lugares donde existe mano de obra cualificada y barata, conllevan el aumento de la productividad y la reducción de costes (Berrocal, García-Alonso, & Murillo, 2010).

## 2.2 Características de las Software Factory del sector financiero

Las Factorías Software del sector financiero que desarrollan y mantienen los sistemas de la banca tienen una serie de características que las definen y en algunos casos diferencias del resto:

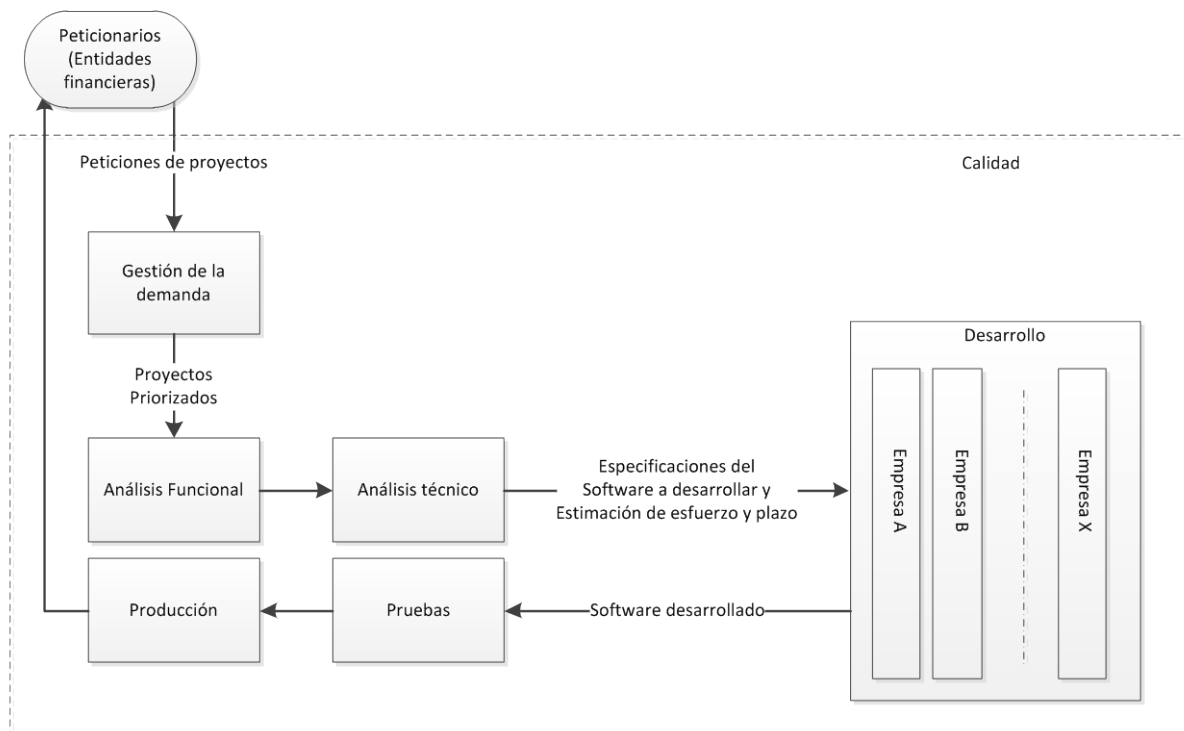
1. Tienen la demanda de trabajo asegurada, siendo incluso superior a la que la propia Factoría Software pueda absorber, por lo que un departamento de gestión de la demanda dependiente del área de organización deberá seleccionar qué desarrollos realizar. Cada uno de los desarrollos solicitados para su creación se suele denominar *petición*.
2. Tiene generalmente asegurado el cobro de sus servicios y desarrollos, ya que su principal cliente será la entidad financiera del grupo empresarial al que pertenece.
3. Para ser lo más productiva posible, la Factoría Software debe facturar por cada petición desarrollada y finalizada, debiendo estar estimado y aceptado previamente por ambas partes el coste de cada una.
4. En base a las características anteriores, el proceso productivo de la Factoría Software debe ser tal que ha de permitir desarrollar el mayor número de peticiones en el menor tiempo posible, manteniendo el grado de calidad establecido.
5. Desarrollado y adaptado el sistema informático a la normativa vigente en cada momento y en cada país, se puede vender el servicio a otras entidades de menor tamaño y cuyos recursos son insuficientes para disponer de un sistema informático propio.

La estructura organizativa propuesta para la Factoría Software estará basada en el ciclo de vida establecido para el desarrollo de software, y no debería diferenciarse sustancialmente del propuesto a continuación, en el que se incluyen los siguientes departamentos:

- Departamento de Gestión de la Demanda: cuyo objetivo es recoger los requisitos del usuario y establecer los métodos de priorización de peticiones.
- Departamento de Análisis Funcional, que traduce los requisitos de usuario identificados en requisitos funcionales.
- Departamento de Análisis Técnico, que se encarga de realizar el detalle técnico de las especificaciones funcionales que deben ser implementadas.
- Departamento de Desarrollo, que se encarga de realizar el desarrollo, construcción y montaje de los requisitos solicitados.
- Departamento de Pruebas, que ha de validar todo lo que ha sido implementado.
- Departamento de Producción, que ha de realizar la puesta en real de los requisitos solicitados por el cliente.
- Departamento de Calidad, que valorará la calidad de cada desarrollo.

La siguiente figura ilustra el proceso habitual seguido en este esquema de funcionamiento:

**Figura 1 Mapa de procesos del modelo de Factorías Software**



El proceso se inicia con las peticiones realizadas por los “clientes”, que en este caso no son otros que las propias entidades financieras. El Departamento de Gestión de la Demanda se encarga de estudiar dichas peticiones y priorizar los proyectos según corresponda. La especificación de los proyectos se realiza en dos etapas normalmente, una a más alto nivel por el Departamento de Análisis Funcional y otra a más bajo nivel por el Departamento de Análisis Técnico. En cualquiera de los casos, la salida del proceso tras estos puntos tiene que ser una especificación detallada del software que se va a desarrollar y una estimación del esfuerzo en horas y del plazo de ejecución. Esta información es la base utilizada para asignar los distintos proyectos al conjunto de empresas que conforman el área de desarrollo y que habitualmente está externalizada. Cada proyecto suele ser controlado a nivel externo por un miembro del Departamento de Análisis Técnico que actúa como Jefe de Proyecto. Una vez que el proyecto es desarrollado pasa a los Departamentos de Pruebas y Producción, momento en el que se entrega al cliente. Todo el proceso es evaluado por el Departamento de Calidad.

### **2.3 Medida de la productividad en las Factorías Software del sector financiero**

El sistema descrito implica varios niveles diferentes de control desde los que se puede evaluar la productividad. Por un lado cada empresa realiza la evaluación de la productividad de los desarrollos que lleva a cabo. El enfoque que se sigue para este tipo de medición de la productividad sigue los esquemas tradicionales de la Ingeniería del Software, basado en métricas como los Puntos de Función, número de defectos, etc. Todas ellas han sido ampliamente estudiadas en la literatura, constituyendo el trabajo de Fenton (Fenton & Pfleeger, 1998) la base de la mayor parte de estas prácticas. La propuesta de cuadro de mando que aquí se presenta no incorpora este nivel, puesto que dentro de la estructura presentada es mucho más interesante la evolución del nivel de productividad que se realiza a nivel superior por el conjunto de Jefes de Proyecto de la organización y por la dirección general. A este nivel se precisa conocer el estado general de la cartera de proyectos, y los

ratios de productividad del conjunto del sistema. Algunas de las preguntas a las que el cuadro de mando tiene que dar respuesta son:

- ¿Cuántas horas/jornadas estoy utilizando?
- ¿A qué coste?
- ¿Cuál son las desviaciones respecto a lo estimado?
- ¿Cuál de mis proveedores es más productivo?

Para incentivar y aumentar la productividad de la Software Factory, la facturación se hará en base a las peticiones desarrolladas. Cada petición tendrá asociado un coste estimado en horas, y se facturará cada petición en base a este coste, independientemente de que una vez desarrollada la petición se hubiesen dedicado más o menos horas a la misma.

A modo de ejemplo, supongamos que una Petición tiene asignado un coste estimado de 800 horas y el coste de cada hora es de 30 euros. En este caso la Software Factory facturaría 24.000 euros por el desarrollo y puesta en real de la petición, independientemente de que tuviese que destinar 600 ó 1000 horas a su desarrollo. Si se logra finalizar la petición en 600 horas la Software Factory habría ganado el coste extra de 200 horas, y si la finalizase en 1.000 horas habría tenido una pérdida de 200 horas. Estas ganancias o pérdidas se repercutirían a la parte variable de los salarios de los empleados o empresas subcontratadas por la Software Factory que forman los distintos departamentos y grupos de trabajo, logrando de esta manera que sea de interés común el mayor número de peticiones posible.

### **3. Aplicación de los cuadros de mando para la medición de la productividad**

De una manera u otra, con simples hojas de cálculo o con programas software específicos, todas las organizaciones miden y valoran diferentes indicadores que les permiten evaluar su productividad.

Para unos la productividad es la relación entre lo producido y los medios empleados para dicha producción, para otros es la relación entre las salidas (outputs) y las entradas (inputs), otros la definen como el indicador de eficiencia que relaciona la cantidad de producto utilizado y la cantidad de producción obtenida. De todas formas y la vista de las distintas definiciones, se deduce que para conocer la productividad se hace necesario poder medirla.

En el área del desarrollo de software, las empresas también necesitan medir la productividad y el rendimiento de los procesos de desarrollo de software con el fin de controlar, gestionar y lograr la mejora continua de sus procesos (Beate, Robert M., & Jochen, 2005).

El retorno de la inversión, conocido por sus siglas en inglés ROI, es un indicador básico en el área de la economía, pero no se suele utilizar de forma eficaz en el campo de los proyectos software, donde son más usuales las métricas de evaluación de productividad ligadas a la generación de código como el “número de líneas de código” o los “puntos de función” (Garzás & Cabrero, 2007). (Erdogmus, Favaro, & Strigel, 2004) recuerdan la carencia de buenos modelos de ROI para el software, haciendo difícil la valoración de las inversiones. Pese a todo no puede negarse que una mejora de la productividad debe verse reflejada en el ROI y que la productividad es la principal métrica para medir la eficiencia en el proceso de desarrollo del software (Garzás & Cabrero, 2007).

Pese a que los indicadores financieros han sido históricamente los más utilizados para la medición de la productividad, evolución y rendimiento de la empresa, la importancia de

combinar indicadores financieros y no financieros ha ido creciendo con los años, especialmente desde la presentación del Cuadro de Mando Integral, o *Balanced Scorecard* (BSC) (Kaplan & Norton, 1992), que identifica cuatro perspectivas (Clientes, Interna, Innovación y Aprendizaje, Financiera) que permiten a los gestores tener una visión general del negocio. Previamente a la publicación del BSC ya se venían utilizando otros cuadros de mando, como el *Tableau de Bord*, introducido en Francia en 1930s y que permite monitorizar el rendimiento de la organización (Bessire & Baker, 2005), o el desarrollado en los 1950s por General Electric, que fue diseñado como un sistema de gestión del rendimiento y que recomendaba la medición de ocho perspectivas diferentes, una financiera y siete no financieras, para la medición del rendimiento.

Posterior a la publicación del BSC, sus autores, siguieron publicando numerosos artículos y libros que ampliaron la visión y conceptos ligados al BSC. Paralelamente a sus publicaciones, otros muchos autores siguieron investigando y publicando acerca del BSC y sus aplicaciones para el control del rendimiento, la productividad y la aplicación de los BSC a los distintos ámbitos empresariales. Otros como (Van Germbergen, 2000) o (Marr & Creelman, 2012) hicieron sus propuestas de IT BSC, ligando el BSC a las Tecnologías de la Comunicación.

El Advanced Performance Institute ([www.ap-institute.com](http://www.ap-institute.com)) pone de manifiesto que las organizaciones que siguen las recomendaciones del BSC logran mejores resultados en cuanto a su rendimiento que aquellas que no lo siguen (Marr, 2010).

#### 4. Cuadro de mando para una Factoría Software del Sector Financiero

A continuación se presenta la propuesta de cuadro de mando objeto de este trabajo:

Tabla 1 Cuadro de mando propuesto

<b>CUADRO DE MANDO PARA UNA FACTORÍA DE SOFTWARE</b>	
<b>Perspectiva financiera</b> , que capta el valor ganado gracias al proceso de las IT. Dentro de la SF	
	Ratio de inversión en IT sobre el total de la inversión ROI de la inversión en IT Mejora de ingresos esperados (por aumento de la productividad) ROI del accionista
<b>Perspectiva del cliente</b> que representará la evaluación que hace el usuario de los desarrollos	
	Calidad técnica y funcional observada Facilidad de uso y tiempo de aprendizaje Cumplimiento de las expectativas Comunicación con el proveedor Resolución de incidencias Grado de satisfacción con el proveedor Recomendación de los Servicio a terceros

<b>Perspectiva de los procesos internos</b>	
	<p><u>Calidad</u></p> <p>Valoración de la documentación</p> <p>Componentes reutilizados / Componentes utilizados</p> <p>Fallos/Incidencias detectados en pruebas por Petición</p> <p>Fallos/Incidencias detectados en producción por Petición</p> <p>Valoración Departamento Calidad</p>
	<p><u>Desviaciones</u></p> <p>Tiempo medio por Petición desde la solicitud del cliente</p> <p>Tiempo medio estimado por Petición</p> <p>Tiempo medio imputado por Petición</p> <p>Desviación en coste por Petición: <math>[(\Sigma \text{Coste estimado} - \Sigma \text{Coste imputado}) / \text{N}^\circ \text{ Peticiones}]</math></p> <p>Desviación en plazo por Petición: <math>[(\Sigma \text{Duración estimada} - \Sigma \text{Duración imputada}) / \text{N}^\circ \text{ Peticiones}]</math></p> <p>Horas Ganadas por Petición: <math>[(\Sigma \text{Horas estimadas} - \Sigma \text{Horas imputadas}) / \text{N}^\circ \text{ Peticiones}]</math></p>
	<p><u>Productividad y Rendimiento:</u></p> <p>Peticiones finalizadas ciclo / Peticiones solicitadas ciclo</p> <p>Productividad estimada por empleado (Total horas estimadas ciclo / N° Empleados)</p> <p>Productividad imputada por empleado (Total horas imputadas ciclo / N° Empleados)</p> <p>Ganancia de productividad (Productividad imputada por empleado / Productividad estimada por empleado)</p>
<b>Perspectiva del Desarrollo y Aprendizaje, que valora el grado de bondad de la organización frente a necesidades futuras</b>	
	<p>Ratio de formación por empleado: <math>[(\Sigma \text{Horas formación} - \Sigma \text{Horas desarrollo}) / \text{N}^\circ \text{ Empleados}]</math></p> <p>Ratio de innovación: (Horas de formación en innovación / Total horas formación)</p> <p>Ratio de investigación: (Horas de investigación / Horas de desarrollo)</p> <p>Edad media de las tecnologías aplicadas</p> <p>Valoración del empleado a la empresa</p>



## 5. Aplicación del Cuadro de Mando a un caso de estudio concreto

Aunque el cuadro de mando ha sido diseñado para ser de aplicación a cualquier Factoría Software del sector financiero, en la actualidad se está aplicando a modo de validación en una factoría concreta. Se trata de una Factoría Software asociada a un importante grupo bancario nacional, aunque por razones de confidencialidad no se puede publicar su nombre. La organización sigue el esquema descrito en la Figura 1, subcontratando a varios proveedores el desarrollo del software. En la actualidad dispone de un nivel de madurez 3 de acuerdo al estándar CMMI-DEV. Se trata de una organización que ha sufrido en los últimos dos años una importante transformación, pasando de una modalidad de trabajo de proyecto tradicional cerrado a un modo de gestión por procesos. El cuadro de mando propuesto ha servido para realizar la medición del rendimiento del nuevo modelo de trabajo. Dado lo reciente de su aplicación, sólo se disponen de datos puntuales que no permiten evaluar la evolución temporal de los indicadores. En la actualidad se está realizando la validación del cuadro de mando a modo prueba piloto. Durante los próximos meses se espera alcanzar una versión estable sobre la que se puedan realizar registros periódicamente de manera que se pueda estudiar la evolución del rendimiento alcanzado por la organización.

## 6. Conclusiones

Este artículo presenta un cuadro de mando para una Factoría Software basado en las propuestas descritas en el Balanced Scorecard y el IT Balanced Scorecard.

Ideado para su análisis por los niveles directivos de una Factoría Software, el cuadro de mando expuesto presenta un conjunto de indicadores, que agrupados según las cuatro perspectivas del BSC, permiten evaluar el estado general de la propia Factoría Software.

La evaluación de los indicadores de rentabilidad, de los clientes de la Factoría Software, de productividad y calidad y formación permiten a la dirección tomar las decisiones oportunas para el mantenimiento futuro del negocio.

## 7. Referencias

- Beate, L., Robert M., B., & Jochen, K. (2005). Holistic Software Process Performance Measurement From the Stakeholders' Perspective.
- Berrocal, J., García-Alonso, J., & Murillo, J. (2010). Lean Management of Software Processes and Factories Using Business Process Modeling Techniques.
- Bessire, D., & Baker, R. (2005). The French Tableau de Bord and the American Balanced Scorecard.
- Correa, A., Werner, C., & Zaverucha, G. (2000). Object oriented design expertise reuse: an approach based on heuristics, design patterns and anti-patterns. *Software Reuse: Advances in Software Reusability*, 33–191.

- Erdogmus, H., Favaro, J., & Strigel, W. (2004). Return on Investment.
- Fenton, N. E., & Pfleeger, S. L. (1998). *Software metrics: a rigorous and practical approach*. PWS Publishing Co.
- Garzás, J., & Cabrero, D. (2007). *El valor y el retorno de la inversión en TSI. En El Gobierno de las TSI*. Ra-ma.
- Garzás, J., & Piattini, M. (2007). *Factorías de Software: Experiencias, tecnologías y organización*. Ra-ma.
- Greenfield, J., Short, K., Cook, S., Kent, S., & Crupi, J. (2004). *Software Factories: Assembling Applications with Patterns. Frameworks, Models & Tools*.
- Kaplan, R. S., & Norton, D. (1996a). Using the Balanced Scorecard as a strategic management system.
- Kaplan, R. S., & Norton, D. (1996b). *The Balanced Scorecard: Translating Strategy Into Action*.
- Kaplan, R. S., & Norton, D. P. (1992). The Balanced Scorecard - Measures that Drive Performance. *Harvard Business Review*.
- Mario Piattini, Félix Oscar García, & Ismael Caballero. (2006). *Calidad de Sistemas Informáticos*. Ra-ma.
- Marr, B. (2010). What is a modern Balanced Scorecard?, Management Case Study, The Advanced Performance Institute.
- Marr, B., & Creelman, J. (2012). Implementing an IT Service Balanced Scorecard.
- Medir la productividad del desarrollo de software en Banca. (s. f.). Recuperado marzo 3, 2012, a partir de [http://www.financialtech-mag.com/000\\_estructura/index.php?ntt=12479&vn=1&sec=4&idb=219](http://www.financialtech-mag.com/000_estructura/index.php?ntt=12479&vn=1&sec=4&idb=219)

Siqueira, F., Barbaran, G., & Becerra, J. (2008). A software factory for education in software engineering.

Van Germbergen, W. (2000). The Balanced Scorecard and IT governance. *Information Systems Control Journal*.

**Correspondencia** (Para más información contacte con):

Área de Proyectos de Ingeniería, Universidad de Oviedo.  
Phone: + 34 985 10 42 72  
E-mail: [montequi@api.uniovi.es](mailto:montequi@api.uniovi.es)  
URL: <http://www.api.uniovi.es>