



UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

Marco Metodológico para la Mejora de la Eficiencia de Uso de los Procesos Software

Autora:

Fuensanta Medina Domínguez

Directores:

**María Isabel Sánchez Segura
Antonio de Amescua Seco**

DEPARTAMENTO DE INFORMÁTICA

Leganés, Febrero 2010

TESIS DOCTORAL

MARCO METODOLÓGICO PARA LA MEJORA DE LA EFICIENCIA DE USO DE LOS PROCESOS SOFTWARE

AUTOR: FUENSANTA MEDINA DOMINGUEZ

**DIRECTORES: MARIA ISABEL SANCHEZ SEGURA
ANTONIO DE AMESCUA SECO**

Firma del Tribunal Calificador:

Firma

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, de de

A mi familia

RESUMEN

La mejora de procesos proporciona beneficios a la organización como reducción de costes, incremento de la productividad, mejora de la calidad, satisfacción del cliente y mayor nivel competitivo. Por ello es cada vez mayor el número de empresas que abordan proyectos de mejora de procesos.

Sin embargo, los proyectos de mejora de procesos requieren de un gran esfuerzo humano, son largos de ejecución, y por consiguiente, muy costosos. Estos proyectos son críticos para la organización que los aborda, puesto que implican un cambio en su proceso de producción, con el fin de lograr una mayor productividad y calidad de los productos que elaboran.

En los proyectos de mejora hay dos actividades clave: la **definición del proceso** y la implantación del proceso definido a toda la organización, denominado **despliegue del proceso**. Actualmente, se puede decir que la definición de procesos está en un nivel poco maduro. Por consiguiente, la tarea de definir procesos es difícil y costosa puesto que cada vez que se aborda la definición de un nuevo proceso se parte de cero. Por otro lado, el despliegue de los procesos definidos, constituye la tarea más desafiante a la que una organización se enfrenta. Esta actividad tiene como finalidad la implantación de un nuevo proceso de manera global en la organización.

La solución que plantea esta tesis doctoral es una solución práctica a la formalización de la definición y despliegue del proceso. Dicha formalización permitirá, por un lado, almacenar el conocimiento y la experiencia de los ingenieros software que han usado el proceso en proyectos anteriores y; por otro lado, permitirá la difusión, búsqueda, uso y reutilización de los activos de procesos en nuevos proyectos.

La solución propuesta es un marco metodológico, orientado a la reutilización y difusión del conocimiento, que incluye:

- “Patrones de Producto”, como artefacto de encapsulación de conocimiento.
- “Estrategia Corporativa”, para desplegar los procesos en toda la organización.
- “Plataforma Colaborativa”, para mejorar la diseminación entre los miembros de los equipos de trabajo de nuevas piezas de conocimiento obtenidas como consecuencia del uso del mismo en proyectos anteriores, así como reducir el rechazo en la adopción de estos.

ABSTRACT

Software Process Improvement (SPI) provides benefits, such as cost reduction, increased productivity, quality improvement and customer satisfaction, to organizations. As a result, more and more organizations are addressing software process improvement projects.

However, a software process improvement project requires a great deal of time and effort and are consequently costly. These projects are very important for organizations because they entail a change in production processes to achieve product quality and greater productivity.

There are two main activities in software process improvement projects: process definition and process deployment. Currently, process definition is under-developed, so each process must be defined from scratch, which is very difficult and costly. On the other hand, process deployment is the most challenging task because a totally new process has to be implemented throughout the organization.

This thesis describes a practical approach to formalizing processes definition and deployment. This will allow the knowledge and experience of software engineers acquired from previous projects and from software engineering best practices to be gathered and the dissemination, search, use and reuse of software processes and projects.

The proposed solution is a methodological framework oriented to knowledge reuse and dissemination, which has three main components:

- “Product Patterns” as a knowledge artifact.
- “Corporate Strategy” to deploy the processes throughout the organization.
- “Collaborative Platform” to improve communication among team members, transfer of new knowledge and to reduce the number of rejections of the framework.

AGRADECIMIENTOS

A mis directores de tesis, Maribel y Antonio, por vuestro inestimable apoyo y dedicación. Gracias por animarme a investigar con vosotros en el área de la Ingeniería del Software.

A Luis García que despertó en mí la inquietud por la docencia y me dio esta oportunidad.

A mis compañeros de laboratorio, especialmente a Arturo, Ana, Javi y Alberto, compañeros de fatiga e investigación.

A Jose (PJ), Silvia, Pei Sin y Chapi, por los ánimos y la energía que me habéis aportado.

A mis padres, mi marido y mi hermana, gracias por vuestra paciencia, comprensión y cariño.

Muchas gracias a todos.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	1
1.1 Contexto	5
1.2 Definición del problema y motivación.....	6
1.3 Objetivos de investigación	11
1.4 Aproximación a la solución	12
1.5 Aportaciones de la investigación.....	14
1.6 Validez de la solución.....	15
1.7 Estructura de la tesis doctoral	18
2. REVISIÓN CRÍTICA DEL ESTADO DE LA CUESTIÓN.....	19
2.1 Gestión del conocimiento	24
2.2 Reutilización: Patrones	43
2.3 Herramientas que contemplan las necesidades de la solución	59
2.4 Resumen del estado de la cuestión	69
3 SOLUCIÓN PROPUESTA	81
3.1 Elemento Clave y su Mecanismo de Creación y Retroalimentación	86
3.2 Lenguaje de patrones de producto	94
3.3 Estrategia corporativa de despliegue de la solución	103
3.4 Componentes y adaptación de la estrategia corporativa a la solución tecnológica....	118
4. VALIDACIÓN	144
4.1 Introducción.....	149

4.2 Planificación de la Validación Experimental.....	150
4.3 Ejecución de la Validación Experimental.....	164
4.4 Análisis de las Fases de validación.....	171
5 CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN.....	217
5.1 Conclusiones	221
5.2 Futuras líneas de investigación	222
6 BIBLIOGRAFÍA.....	225

1. INTRODUCCIÓN

1. INTRODUCCIÓN	1
1.1 Contexto	5
1.2 Definición del problema y motivación.....	6
1.3 Objetivos de investigación	11
1.4 Aproximación a la solución	12
1.5 Aportaciones de la investigación.....	14
1.6 Validez de la solución.....	15
1.7 Estructura de la tesis doctoral	18

1.1 CONTEXTO

El proceso software es el medio por el cual las personas, procedimientos, métodos y herramientas se integran para producir los resultados deseados. Asimismo, el proceso software también puede ser definido como el conjunto de actividades, métodos, prácticas y transformaciones que las personas utilizan para desarrollar y mantener el software y los productos asociados (planes de proyecto, documentación de diseño, código, casos de prueba, manuales de usuario) (CMU/SEI, 2006).

La estrategia orientada a encaminar a las organizaciones a trabajar con un enfoque centrado en procesos se denomina Mejora de Procesos. Esta estrategia de trabajo ha demostrado su eficacia, al proporcionar importantes beneficios a las organizaciones; principalmente, en mejorar la calidad de sus productos y servicios, aumento de la productividad, y mayor tasa de éxito de los proyectos (Capell, 2004) (Sommerville, 2004) (SEI, 2006) (Niazi et al, 2006) (Allison et al, 2007).

Esta estrategia está siendo adoptada por organizaciones en el mundo entero (SEI, 2006). Podemos decir, que hoy en día se ha convertido en un estándar internacional y que las organizaciones que no adopten este esquema de trabajo quedarán descolgadas en un mercado cada vez más globalizado y competitivo.

El hecho de que la industria del software juegue un papel cada vez más importante en la economía global ha conducido a que el concepto de Gestión del Proceso Software¹ se haya convertido en una disciplina; generando nuevos retos a la comunidad investigadora para abordar problemas relativos al proceso software.

Este trabajo de investigación aborda temas de investigación que el Consorcio Internacional de Investigación de Procesos² (IPRC, 2006) ha establecido como prioritarios; en concreto, la reutilización de procesos y la adopción efectiva de los mismos por todos los profesionales de una organización.

¹ El concepto de Gestión del Proceso Software fue introducida por el Software Engineering Institute (SEI) de la Universidad de Carnegie Mellon, en 1985.

² En 2004 el SEI formó el International Research Process Consortium (IRPC) para identificar las líneas de investigación estratégicas en los procesos de sistemas y de software

La investigación en Gestión del Proceso Software requiere de una visión multidisciplinar, no monolítica. Es por ello, que este trabajo se basa en los fundamentos del diseño de patrones, y en los fundamentos de la gestión del conocimiento para abordar el objetivo principal de esta investigación: “Mejorar la eficiencia de uso de los Procesos Software”.

El patrón, desde el punto de vista de reutilización del software, es un componente o producto que puede utilizarse en una factoría de producción software para el desarrollo de un sistema software. Pero desde el punto de vista de la Gestión del Conocimiento, es un activo que da valor a la organización; y por consiguiente, debe ser capaz de adquirirse, representarse, ser usado, y poder ser difundido por toda la organización.

La Reutilización del Software ha evidenciado una mejora de la productividad mediante la reducción del coste de desarrollo del producto software (Selby, 2007) (Mohagheghi et al, 2007). La Gestión del Conocimiento genera, igualmente, una mayor productividad al identificar el conocimiento válido y socializarlo en la organización. Además dota a los profesionales y por consiguiente a la organización de una mayor capacidad de innovación.

Desde esta visión multidisciplinar, hemos redefinido y diseñado un patrón que pudiera recoger el concepto integral de procesos software y, que al mismo tiempo, fuese un componente que fuese fácil de usar y comprender por los profesionales no expertos en el proceso y, por supuesto, que proporcionase valor para la organización de desarrollo de software.

1.2 DEFINICIÓN DEL PROBLEMA Y MOTIVACIÓN

Los proyectos de mejora de procesos requieren de un gran esfuerzo humano, son largos de ejecución, y por consiguiente, muy costosos. Estos proyectos son críticos para la organización que los aborda, puesto que implican un cambio de su proceso de producción, con el fin de lograr una mayor productividad y calidad de los productos que elaboran.

En estos proyectos hay dos actividades claves: la **definición del proceso** y la implantación del proceso definido a toda la organización, denominado **despliegue del proceso**.

Actualmente, la definición de un proceso carece de una formalización que proporcione mecanismos para su caracterización, búsqueda, facilidad de uso, recogida de información de experiencia por las personas que han usado el proceso en proyectos anteriores, medidas de uso, etc.

Podemos decir que la definición de procesos está en un nivel poco maduro. Por consiguiente, la tarea de definir procesos es difícil y costosa puesto que cada vez que se aborda la definición de un nuevo proceso se parte de cero. Es necesario siempre tener expertos en el área del proceso y expertos de gestión de procesos para abordar esta actividad. Actualmente, existen pocas herramientas para ayudar a realizar esta actividad y además son muy caras.

Por otro lado, el despliegue de los procesos definidos, constituye la tarea más desafiante a la que una organización se enfrenta. Esta actividad tiene como fin el implantar el nuevo proceso de una manera total en la organización. Por consiguiente, las personas implicadas en el proceso deben realizar sus actividades de manera diferente a cómo lo estaban realizando hasta el momento.

La minimización a la resistencia al cambio es un elemento clave que se debe tener en cuenta a la hora de buscar una solución para lograr el éxito en un despliegue de procesos.

En resumen, esta investigación aborda los siguientes problemas:

- definir un proceso que se pueda reutilizar por otros ingenieros de software; esto es, cómo diseñar un proceso a partir de componentes de procesos o de otros procesos.
- evolucionar un proceso fácilmente; es decir, cómo recoger información o experiencias que aporten valor del uso del proceso en un proyecto por los usuarios del proceso.
- adopción efectiva y despliegue del proceso; lo que significa, cómo facilitar el uso del proceso, cómo lograr que los usuarios del proceso lo sigan fielmente, y cómo lograr los resultados y la calidad esperada de la realización del proceso.

Motivación

La motivación por la que las organizaciones abordan programas de mejora de proceso es debido a los beneficios que proporciona, como la reducción de costes, incremento de la productividad, mejora de la calidad, satisfacción del cliente y mayor competitividad a las organizaciones ver Tabla 1-1(Gibson et al, 2006). Por ello, es cada vez mayor el número de empresas que abordan proyectos de mejora de procesos (Phillips, 2008).

Categoría	Rendimiento medio
Coste	34%
Planificación	50%
Productividad	61%
Calidad	48%
Satisfacción del Cliente	14%
Retorno de la Inversión	4.0:1

Tabla 1-1 Categorización de Mejoras de Rendimiento

Mayor nivel de competencia de las empresas que desarrollan software ha llevado a buscar soluciones más económicas para producir software. Las factorías de software son centros especializados en desarrollo de una línea de productos que al ubicarse en países donde la mano de obra es más barata consigue precios más económicos. En los últimos años se ha incrementado el número de empresas que realizan outsourcing. El outsourcing mueve al año billones de dólares, en 2001 movió 70 billones de dólares en comparación con los 160 billones de dólares que movió en 2005 (Gartner 2005). Se estima que actualmente el 15% de las empresas europeas, lideradas por Gran Bretaña y Alemania, realizan outsourcing (Sourcingmag 2007).

Debido a que las factorías de software son centros de alto rendimiento, trabajar con un enfoque basado en Procesos es un estándar de obligado cumplimiento.

Países emergentes, como China e India, han decidido liderar la industria del software. Son en estos países donde se encuentran, actualmente, el mayor número de empresas con mayor nivel de capacidad de sus procesos. Otros países como Brasil, Japón, Francia, Taiwán y España están haciendo un fuerte esfuerzo para tener niveles altos de competitividad en esta industria (Phillips, 2008).

Los beneficios que proporcionan el trabajar con un enfoque basado en proceso, junto con requerimientos contractuales de certificación de tener un determinado nivel de capacidad de proceso, que las organizaciones contratantes demandan a las organizaciones que desarrollan software, más la globalización del desarrollo de software, implica encontrar soluciones más asequibles económicamente y de plazos de tiempo.

Por otro lado, el modelo de Mejora de Procesos CMMI (CMU/SEI, 2006) establece que las organizaciones deben iniciar sus proyectos de mejora centrándose en los procesos de Gestión de Proyectos. Los Procesos de Gestión de Proyectos tienen como objetivo realizar los proyectos de ingeniería con éxito (proyectos que terminan en tiempo, presupuesto y cumpliendo con todos los requisitos del cliente). Actualmente, sólo el 32% (Standish Group, 2009) de los proyectos software han finalizado con éxito. Por ello, esta tesis doctoral propone un enfoque orientado al proceso y a la reutilización del conocimiento como propuesta para mejorar la eficiencia de uso de los procesos software de los proyectos, manteniendo y/o mejorando parámetros de calidad de los productos obtenidos.

En la Tabla 1-2 se muestran los problemas identificados en dos de las actividades de los proyectos de mejora donde se centra este trabajo de investigación, la definición y el despliegue de los procesos software; el área donde se ubica la solución, el enfoque de la solución propuesta y las evidencias de los beneficios de abordar dichos problemas con el enfoque de la solución propuesta.

Actividad de Mejora: Definición de Procesos Software			
Problema	Área donde se ubica la Solución	Enfoque	Evidencia
Adquisición, Representación y Uso del conocimiento	Gestión del Conocimiento	Conocimiento de la organización es un activo y da valor al Capital Intelectual de la organización.	Conocimiento genera mayor productividad y mayor capacidad de innovación.
Usabilidad del Proceso	Mejora de Procesos Software	Estrategia de trabajo orientada a procesos: combina personas, métodos y tecnología.	Beneficios de mejora de productividad y de calidad de los productos y reducción de riesgo de fracaso de proyectos.
Actividad de Mejora: Despliegue del Proceso Software			
Problema	Área donde se ubica la Solución	Enfoque	Evidencia
Accesibilidad del conocimiento	Reutilización del Software	Enfoque de factoría de producción: Definir componentes que puedan integrarse y utilizarse en otros desarrollos.	Aumentar la productividad: reducir tiempo de desarrollo del producto.
Transferencia del conocimiento	Entornos colaborativos	Difusión del conocimiento en los diferentes niveles organizativos: mecanismos de colaboración, coordinación y compartición de conocimiento.	Incremento de la calidad de los productos y reducción de costes y tiempos.

Tabla 1-2 Problemas identificados, Área, Enfoque y Evidencia de la Solución

1.3 OBJETIVOS DE INVESTIGACIÓN

El propósito de esta tesis doctoral es realizar un marco metodológico que permita la difusión, uso y reutilización del conocimiento de las mejores prácticas de la ingeniería del software y de la experiencia de los ingenieros software, con el objetivo de mejorar la eficiencia de uso de los procesos software (reduciendo el tiempo empleado para desarrollar las actividades), manteniendo y mejorando la calidad de los productos software generados.

Los objetivos de la tesis doctoral son:

- **Objetivo 1:** Definir un artefacto que permita representar la experiencia y el conocimiento de los ingenieros software expertos sobre cómo hacer un producto software, basándose en la reutilización de las mejores prácticas de la ingeniería del software y del conocimiento tácito de los ingenieros expertos para obtener un producto software específico, con el fin de incidir en la resolución de la adquisición, representación, uso y, reutilización del conocimiento.
- **Objetivo 2:** Definir un marco metodológico que proporcione una estrategia corporativa que describa los procesos para realizar la difusión, uso y reutilización de los activos de proceso en una organización, con el fin de incidir en la resolución de la accesibilidad y transferencia del conocimiento.
- **Objetivo 3:** Mejorar la eficiencia de uso de los procesos software reduciendo el tiempo de desarrollo de los productos software (Hornbæk, 2006), con el fin de incidir en la resolución de la mejora de la usabilidad de procesos software.
- **Objetivo 4:** Mejorar parámetros que inciden en la calidad de los productos software a desarrollar, con el fin de incidir en la resolución de la mejora de la usabilidad de procesos software.

1.4 APROXIMACIÓN A LA SOLUCIÓN

La solución que plantea esta tesis doctoral es una solución práctica a la formalización de la definición del proceso. Dicha formalización permitirá: almacenar el conocimiento y la experiencia de los ingenieros software que han usado el proceso en proyectos anteriores; y, la difusión, búsqueda, uso y reutilización de los activos de procesos y proyectos. La solución propuesta se basa en la reutilización del conocimiento y en mecanismos de colaboración que facilitarán el despliegue de los procesos definidos a través de una estrategia corporativa definida.

La solución planteada se basa en la reutilización del conocimiento a través del uso de patrones software. Los patrones software son un artefacto que representan soluciones a problemas que surgen en el desarrollo de aplicaciones, permitiendo su reutilización como respuesta a conflictos similares surgidos en contextos diferentes. La clave está en que el patrón software es una generalización o abstracción reusable que puede utilizarse como punto de partida para soluciones futuras. El hecho de tener en consideración este aspecto es importante ya que lo que se pretende con esta tesis doctoral es facilitar la difusión, uso y reutilización de activos de procesos, por ello, la utilización de patrones como el artefacto que almacene el conocimiento de las mejores prácticas de los ingenieros expertos y del conocimiento tácito de las organizaciones para su posterior reutilización, se adecua a las necesidades de esta tesis doctoral.

Existen muchos tipos de patrones en el área de la Ingeniería del Software (ver capítulo 2), pero ninguno proporciona una solución que se adecue a las necesidades de la formalización de la definición del proceso software. El único patrón que se asemeja es el patrón de proceso, pero la solución que proporcionan los patrones de proceso es más restringida y menos portable, ya que proporcionan los pasos a realizar para llevar a cabo dicho proceso y no está enfocado en proporcionar una solución que proporcione los pasos para la obtención de un producto software en el desarrollo de un proyecto o proceso que es el objetivo a perseguir en esta tesis doctoral. Por ello, *este trabajo propone un tipo de patrón, denominado Patrón de Producto, que permite almacenar, empaquetar, proporcionar y reutilizar el conocimiento de las mejores prácticas de la ingeniería del software así como el conocimiento de los ingenieros expertos en los proyectos de las organizaciones para obtener un producto software específico.* Considerando un producto como el elemento mínimo en la Ingeniería del Software que se obtiene de la ejecución de una actividad de desarrollo software.

Uno de los problemas encontrados en la interacción de la gestión de conocimiento con la mejora de procesos, fue la difusión del conocimiento en los diferentes niveles organizativos, lo que se traduce en una mínima transferencia de conocimiento en las organizaciones, a pesar de que se han realizado estudios sobre los beneficios que obtienen las organizaciones cuando se aplica en los proyectos software, el conocimiento generado en proyectos previos: incremento de la calidad de los productos y, reducción de costes y tiempos. Con la transferencia de conocimiento, se proporciona a los empleados de una organización la información en el momento adecuado para la toma de decisiones; así como el conocimiento necesario para poder realizar cada uno de las actividades o tareas de un proceso. Es por ello, que ***esta solución está soportada tecnológicamente por una plataforma software colaborativa que permite la comunicación, coordinación y compartición de conocimiento, independientemente del tiempo y el espacio.***

Este trabajo define un marco metodológico para la difusión, uso y reutilización de activos de procesos, mediante un artefacto definido en esta tesis doctoral, *Patrón de Producto*, y una estrategia corporativa, utilizando como soporte tecnológico una plataforma web colaborativa. Dicho marco metodológico describe:

1. El elemento clave para la reutilización de activos de proceso, *Patrón de Producto*, así como sus mecanismos de colaboración.
2. Un lenguaje de *Patrones de Producto* donde se van a describir los siguientes elementos: tipos de *patrones de producto*; el catálogo de *patrones de producto* desarrollado en esta tesis doctoral; la regla de búsqueda de *patrones de producto*; el metamodelo que describe la relación entre los *Patrones de Producto* y la lógica de la ejecución de un proyecto software; el proceso de retroalimentación y creación de *patrones de producto*.
3. La estrategia corporativa de despliegue de la solución que describe los pasos para realizar la difusión, uso y reutilización de los activos de proceso en una organización.
4. Una solución tecnológica basada en una plataforma colaborativa que da soporte a la solución planteada en esta tesis doctoral. Así mismo, se describe cómo se adapta la estrategia corporativa a la solución tecnológica planteada.

1.5 APORTACIONES DE LA INVESTIGACIÓN

Las aportaciones que obtienen las organizaciones al aplicar la solución propuesta en esta tesis doctoral se manifiesta en una mejora de los siguientes parámetros: reutilización, entendimiento del proceso, aprendizaje del proceso, comunicación entre el equipo de desarrollo, eficiencia de uso de los procesos, trazabilidad y calidad de los productos software. Estos parámetros se describen a continuación.

- **Reutilización:** la reutilización es un factor crucial, ya que el conocimiento que se genera en las organizaciones es el activo más valioso en la actual sociedad de la información, por lo que se debe almacenar y utilizar como activos reutilizables por otros proyectos en la organización. A través del artefacto definido, *Patrón de Producto*, y el mecanismo de colaboración que se propone en esta tesis doctoral, se podrá reutilizar el conocimiento de los ingenieros expertos así como las mejores prácticas de la Ingeniería del Software.

- **Entendimiento del proceso:** el entendimiento del proceso se mejora gracias a que se representan gráficamente los pasos que se tienen que realizar para desarrollar un producto software, de forma que en cada momento se sabe qué, quién y cómo se tiene que realizar las actividades del proyecto software.

- **Aprendizaje de los procesos:** los pasos sistemáticos que se proponen mejorarán el aprendizaje del proceso que se debe realizar para obtener un producto software utilizando las mejores prácticas de la Ingeniería del software, debido a que los pasos sistemáticos son repetibles.

- **Comunicación entre el equipo de desarrollo:** este factor es crucial debido a que es necesario que la transferencia de conocimiento se realice en las organizaciones. Gracias a esa transferencia y compartición del conocimiento en las organizaciones, los participantes implicados en el desarrollo de un proyecto, sabrán en cada momento qué actividad tienen que realizar y lo más importante, cómo tienen que realizarla. Para ello, en este trabajo se ha definido una arquitectura colaborativa y mecanismos de comunicación entre los *Patrones de Producto*.

- **Eficiencia de uso de los procesos:** la eficiencia de uso se mejorará disminuyendo el tiempo de desarrollo de los productos software. Éste parámetro se va a mejorar gracias a que

los *Patrones de Producto* van a proporcionar el conocimiento necesario para realizar las actividades del desarrollo software y a que gracias a la estrategia corporativa y a la plataforma colaborativa van a mejorar la transferencia y compartición del conocimiento.

-**Trazabilidad** hace referencia a la facilidad para seguir el proyecto. Se mejorará la trazabilidad en dos aspectos esenciales:

1.- los pasos sistemáticos que se proponen en el marco metodológico, mejorarán la trazabilidad de la implantación de los procesos software en las organizaciones.

2.- los pasos sistemáticos que proporciona el *Patrón de Producto*, mejorarán la trazabilidad en la realización de las actividades del desarrollo software.

- **Calidad:** se mejorarán parámetros de calidad en la primera versión de los productos software generados en el desarrollo software y se mejorará el tiempo empleado en hacer los productos software una y otra vez hasta que pasen el nivel de calidad exigido, gracias al conocimiento que aportan los *Patrones de Producto*.

1.6 VALIDEZ DE LA SOLUCIÓN

Este trabajo se enmarca en el Proyecto Nacional “GPS – Plataforma de Gestión de Procesos Software: Modelado, Reutilización y Medición” (TIN-2004-07083), desarrollado en la Universidad Carlos III de Madrid, a través del cual surge la necesidad de mejorar la eficiencia de uso de los procesos software y facilitar la difusión, uso y reutilización de activos de procesos y proyectos.

De entre los métodos de investigación existentes en el campo de la Ingeniería del Software, esta tesis doctoral utiliza el método empírico para validar la solución propuesta: se propone una teoría formal y es evaluada a través de estudios empíricos que se realizan mediante casos de estudio y experimentos.

La validación de la solución propuesta en esta tesis doctoral se descompone en tres fases:

1. Fase I: Realización de los proyectos software sin el uso de los *patrones de producto* ni del Marco Metodológico.

2. Fase II: Validación de los *patrones de producto* definidos en el Catálogo de Patrones de Producto realizado en esta tesis doctoral.
3. Fase II: Validación del Marco Metodológico a través del despliegue de la solución propuesta en esta tesis doctoral.

Fase I: Realización de los proyectos software sin el uso de los Patrones de Producto ni del Marco Metodológico.

El objetivo de la Fase I es el desarrollo de diez proyectos software sin la utilización de los patrones de producto ni del marco metodológico, únicamente con el conocimiento y la experiencia del jefe de proyecto y su equipo de trabajo. Los proyectos software se desarrollarán utilizando la metodología propuesta por Craig Larman y los procesos de Gestión de proyecto.

Fase II: Validación de los patrones de producto definidos en el catálogo de Patrones de Producto definidos en esta tesis doctoral.

El objetivo de esta fase es la validación de los patrones de producto creados y almacenados en el catálogo de patrones y accesible a través de la wiki desarrollada en esta tesis doctoral. En esta fase se validan los objetivos 1, 3 y 4 de la tesis doctoral.

El propósito de esta fase es:

1. Realizar los mismos proyectos software que en la fase I utilizando los *patrones de producto* definidos y almacenados en la wiki, con equipos de desarrollo distintos a los equipos que realizaron los proyectos software sin *patrones de producto* (Fase I).
2. Comprobar que la utilización de los *patrones de producto* facilita el acceso, uso y reutilización del conocimiento de las buenas prácticas en los procesos software; mejora la eficiencia de uso de dichos procesos software y la calidad de los productos obtenidos.

Para ello, se desarrollarán los mismos diez proyectos software por equipos de desarrollo distintos, siguiendo la metodología propuesta por Craig Larman y los procesos de Gestión de proyecto, utilizando las técnicas de Ingeniería del Software mediante los *patrones de producto* definidos en el catálogo de patrones de producto.

Durante la realización de los proyectos se realizará el seguimiento del proyecto y se pasará un cuestionario para obtener los datos necesarios para analizar si se ha facilitado el acceso, uso y reutilización del conocimiento de las buenas prácticas en los procesos software mediante los *patrones de producto*.

Fase III: Validación del Marco Metodológico a través del despliegue de la solución propuesta en esta tesis doctoral.

El objetivo de esta fase es validar que es posible mejorar la eficiencia de uso de los procesos software, utilizando el marco metodológico propuesto en esta tesis doctoral. Para ello se ha utilizado el elemento que permite encapsular y reutilizar el conocimiento, *patrón de producto*, y la estrategia corporativa, soportado por tecnología colaborativa. En esta fase se validan los objetivos 2, 3 y 4 de la tesis doctoral.

El propósito de esta fase es:

1. Puesta en práctica del Marco Metodológico descrito en esta tesis doctoral a través de una plataforma colaborativa.
2. Comprobar que la utilización del marco metodológico que incluye la utilización de los *patrones de producto* y la estrategia corporativa, a través de la plataforma colaborativa, facilita la difusión, uso y reutilización de los activos de procesos y proyectos en las organizaciones mejorando la eficiencia de uso de los procesos software y la calidad de los productos obtenidos.

Durante la realización de los proyectos, se llevará a cabo el seguimiento de los proyectos y se pasará un cuestionario para obtener los datos necesarios para analizar si se ha facilitado la transferencia, implantación y reutilización de activos de procesos mediante el Marco Metodológico.

1.7 ESTRUCTURA DE LA TESIS DOCTORAL

Esta memoria se ha estructurado en seis capítulos que se describen a continuación.

- **Capítulo 1. Introducción.** Es el capítulo actual, en el que se presenta el contexto de la investigación. Se define el problema en base a la problemática actual dentro del contexto, y se recoge la motivación de la investigación. Se describen los objetivos de investigación. Se expone brevemente una aproximación a la solución. Se detallan las aportaciones a la investigación y por último se describe cómo se realiza la validez de la solución propuesta en este trabajo de investigación.
- **Capítulo 2. Revisión crítica del estado de la cuestión.** Se presenta el estado de la cuestión realizando una revisión crítica de los trabajos existentes y comparando dichas soluciones con la solución propuesta en esta tesis doctoral.
- **Capítulo 3. Solución Propuesta.** Se presenta la solución propuesta en esta tesis doctoral, en el que se define un marco metodológico para la mejora de la eficiencia de uso de los procesos software.
- **Capítulo 4. Validación.** Se describe la planificación, ejecución y el análisis de los datos obtenidos en la validación experimental realizada en este trabajo de investigación.
- **Capítulo 5. Conclusiones y futuras líneas de investigación.** En este último capítulo se muestran las conclusiones de la presente tesis doctoral y las posibles líneas futuras en relación a la propuesta de la investigación.

2. REVISIÓN CRÍTICA DEL ESTADO DE LA CUESTIÓN

2. REVISIÓN CRÍTICA DEL ESTADO DE LA CUESTIÓN.....	19
2.1 Gestión del conocimiento	24
2.2 Reutilización: Patrones	43
2.3 Herramientas que contemplan las necesidades de la solución	59
2.4 Resumen del estado de la cuestión	69

Esta tesis doctoral plantea una solución práctica a la definición y el despliegue del proceso software con el objetivo de mejorar la eficiencia de uso de los procesos software y el escaso uso de la Ingeniería del Software en los proyectos software. Para resolver dicho problema, esta tesis doctoral se apoya en tres áreas de investigación:

- Information science: En este área se estudian la aplicación y uso del conocimiento en las organizaciones, la interacción entre las organizaciones y las personas así como los sistemas de información. Esta tesis se apoya concretamente en el área de la gestión de conocimiento, ya que uno de los usos de la gestión del conocimiento es soportar las actividades que apoyan la mejora de procesos software, proporcionando una interacción positiva que beneficia a la organización y a los programas de mejora (Burke et al, 2005) (Dingsoyr et al, 2009).

- Social software: Dentro de éste área, se centra en groupware, ya que la solución propuesta se apoya en funcionalidades y mecanismos de los entornos colaborativos como soporte tecnológico de la solución (transferir, usar colaborativamente, compartir, etc).

- Ingeniería del Software: La aportación de esta tesis doctoral se realiza en esta área de investigación, concretamente en el área de la mejora de procesos software, proponiendo un artefacto y marco metodológico para mejorar la eficiencia de uso de los procesos. Para ello, esta tesis doctoral se apoya en la línea de investigación de reutilización desde la perspectiva de la Ingeniería del Software. En esta tesis doctoral, la reutilización está enfocada en la identificación de los artefactos y mecanismos que sirvan para la reutilización del conocimiento de los ingenieros expertos y la reutilización de activos de proceso en proyectos software. Por ello, se necesita un artefacto que permita dar una solución práctica a un problema bajo un contexto y unas restricciones, y que tenga la característica principal de la reutilización. Los patrones software serán utilizados como artefactos para la solución de esta tesis doctoral.

La originalidad de este trabajo radica en la forma en que se aborda el problema de la mejora de la eficiencia de uso de los procesos software mediante la interrelación de tres disciplinas como son, la gestión de conocimiento, la reutilización desde la perspectiva de la ingeniería del software, y los entornos web colaborativos como plataforma propuesta para que la solución sea realmente práctica y usable en el sentido de proveer de accesibilidad a la solución teórica propuesta.

A continuación se pasa a detallar los trabajos más relevantes de cada una de las disciplinas identificados, comparando cada uno de ellos con el problema que se trata de resolver en este trabajo.

2.1 GESTIÓN DEL CONOCIMIENTO

En el contexto de la ingeniería del software, se define gestión de conocimiento como un conjunto de actividades, técnicas y herramientas que soportan la creación y transferencia del conocimiento a través de las organizaciones. El área de gestión de conocimiento es considerada por (Holsapple et al, 2004) como esfuerzos sistemáticos y deliberados de una entidad para expandir cultivar y aplicar conocimiento disponible para añadir valor a la entidad, en el sentido de resultados positivos para llevar a cabo sus objetivos o sus propósitos.

Muchos autores han investigado los estadios que conforman la gestión de conocimiento (Hey, 2004) y (Burke et al, 2005). En ambos trabajos se analizaron los estadios del conocimiento y cómo las organizaciones pueden pasar del estadio más simple, los datos, al estadio más alto de la gestión de conocimiento, la innovación. A continuación, se resumen dichos estadios, cómo se pasa de un estadio a otro y la forma en que se manejan estos estadios en esta tesis doctoral.

Los estadios identificados en la gestión de conocimiento son (Hey, 2004) (Burke et al, 2005):

- Dato: es la unidad o cantidad mínima de información no elaborada, sin sentido en sí misma.
- Información: Es un conjunto de datos que al relacionarse adquieren sentido.
- Conocimiento: El conocimiento es una apreciación de la posesión de múltiples datos interrelacionados que por sí solos poseen menor valor cualitativo. Existen dos tipos de conocimiento, el conocimiento explícito y el conocimiento tácito. El conocimiento explícito es el conocimiento que se puede transmitir formalmente y sistemáticamente porque este conocimiento se puede expresar en forma de datos, a través de formatos como formularios científicos, bases de datos, manuales, etc. El conocimiento tácito es el conocimiento que almacena el know-how de la empresa, las habilidades y experiencia de sus empleados, por tanto es un conocimiento que es

difícil de formalizar, comunicar y compartir entre las personas que forman la organización (Polanyi, 1966) (Nonaka et al, 1998) (Christensen, 2003). El conocimiento constituye el activo más valioso de cualquier organización en la Sociedad de la Información. La competitividad de las empresas, y por lo tanto su supervivencia, depende de que tanto el conocimiento explícito como tácito pueda preservarse y utilizarse de forma eficiente.

- Innovación: la innovación tiene su origen en el conocimiento y es la materialización de los avances que se derivan del conocimiento acumulado y que se concreta en la creación, introducción y difusión de nuevos y mejorados procesos. Hay diferentes definiciones de innovación, el informe Cotec 1999 define la innovación como “el resultado de un proceso complejo e interactivo” y como “el arte de transformar el conocimiento en riqueza y calidad de vida”.

En la Figura 2-1 se ilustra la jerarquía DIKW (Data, Information, Knowledge, Wisdom). En esta jerarquía, las empresas tienen que alcanzar el estadio de sabiduría o innovación a través del análisis y reflexión de los datos, información y conocimiento obtenidos a través de la experiencia. Este estadio de innovación se obtiene basándose en la originalidad, ya que se tiene que pasar del conocimiento adquirido a través de la experiencia, a la creación de nuevo conocimiento. De esta innovación, entre otros factores, dependen las empresas para poder obtener ventajas competitivas.

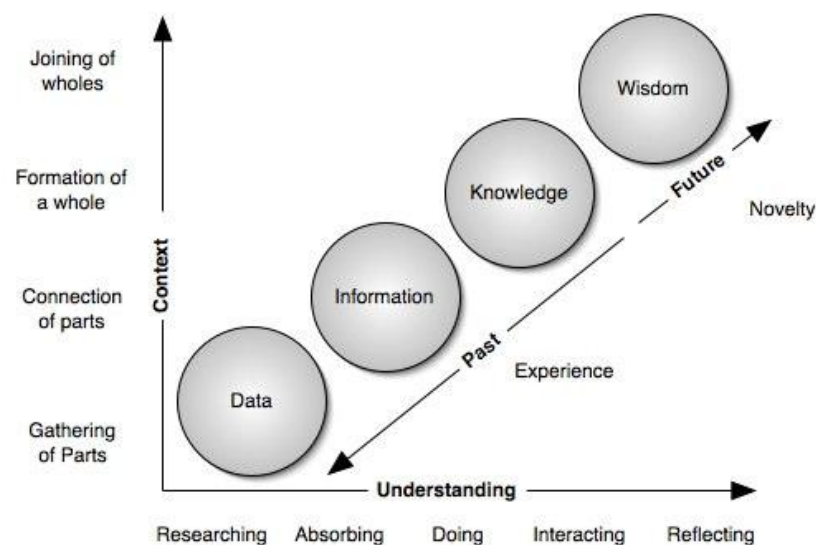


Figura 2-1 Visión de la jerarquía DIKW (Clark, 2004)

En el área de la gestión de conocimiento se han definido modelos de éxito para el uso de la gestión de conocimiento. La definición de un modelo de éxito no es el centro de este trabajo, pero se enuncian a continuación los más relevantes:

- El modelo definido por Bots y Bruin (Bots et al, 2002) en el que define una cadena de valor del conocimiento (Determinar estrategias y políticas - Hacer inventario y determinar el conocimiento que se necesita - Compartir el conocimiento - Aplicar el Conocimiento).
- El modelo Massey, Montoya-Weiss y O'Driscoll (Massey et al, 2002), en este modelo se refleja que el éxito de la gestión de conocimiento se basa en entender a la organización, sus usuarios y cómo ellos usan el conocimiento.
- El modelo Lindsey (Lindsey, 2002), en el que define la eficacia de la gestión de conocimiento en términos de dos principales características: la capacidad de la infraestructura del conocimiento (capital social y las relaciones entre las fuentes de conocimiento, sus usuarios, tecnología, estructura y cultura) y la capacidad del proceso de conocimiento (integración del proceso de gestión de conocimiento dentro de la organización, adquisición del conocimiento, adaptar dicho conocimiento a las necesidades, aplicación del conocimiento y la seguridad de dicho conocimiento).
- El modelo Jennex-Olfman (Jennex et al, 2004), este modelo evalúa el éxito como la mejora en la efectividad de la organización basándose en el uso y el impacto del sistema de gestión de conocimiento, a través de dimensiones como son: la calidad del sistema, calidad de la información y el conocimiento, satisfacción de usuario y el beneficio percibido.
- El modelo Maier (Maier, 2002), este modelo es muy similar al model Jennex-Olfman, se basa también en dimensiones como son, calidad del sistema, calidad de la información, comunicación y conocimiento, servicio conocimiento-específico, impacto individual, impacto en grupos de personas, impacto en la organización. En este modelo se añaden medidas en cada una de las dimensiones.

En un estudio realizado sobre el rol de la gestión de conocimiento en las organizaciones de desarrollo software, se identificaron seis tipos de actividades del conocimiento (Feher et al, 2006):

- Registrar y procesar la experiencia organizacional y personal de los proyectos de desarrollo software para compartir y reutilizar en el futuro.
- Recopilar la información y el conocimiento requerido para realizar las tareas de un proyecto software.
- Definir estándares, las mejores prácticas de los procesos, dar acceso a esas definiciones y transferir el conocimiento a los empleados.
- Extraer, almacenar y comunicar información y conocimiento del cliente y sobre los requisitos del cliente a desarrollar durante el proceso de desarrollo
- Examinar y explorar los actuales procesos de desarrollo software de una organización, monitorizar el proceso y basándose en los resultados determinar las fortalezas y debilidades, que pueden ser la base de mejoras de procesos posteriores.
- Adquirir e integrar conocimiento externo.

Al igual que en el estudio (Feher et al, 2006), en este trabajo doctoral se considera dichas actividades imprescindibles en el rol de la gestión de conocimiento, en las organizaciones de desarrollo software. En la solución propuesta en esta tesis doctoral, se proponen dos actividades más:

- Retroalimentar el conocimiento adquirido con el conocimiento que se obtiene de la realización de los proyectos
- Reutilización del conocimiento adquirido en proyectos futuros.

Existe un gran número de proyectos e investigaciones que se centran directamente en la definición de mecanismos de representación del conocimiento de los proyectos software. A continuación, se presentan las cuatro iniciativas más ampliamente difundidas, relacionadas con la representación del conocimiento de los proyectos software para su posterior reutilización.

2.1.1 Trabajos relacionados con la Representación del conocimiento

2.1.1.1 Web Semánticas

En (Antunes et al, 2007) se describe un sistema basado en sistemas de reutilización semánticas (SRS) para adquirir, gestionar y reutilizar el conocimiento de desarrollo software basado en tecnologías de Web Semánticas. En dicho trabajo se proponen tres capas lógicas:

- Capa de datos: corresponde con la base de conocimiento usando SRS que está compuesto por una ontología de representación, la ontología del dominio, un repositorio de los elementos que resultan de los procesos de desarrollo software (Software Development *Knowledge* Element - SDKE) y los datos del sistema y del usuario. Para representar dichos datos utiliza lenguajes para representar Web Semánticas como Resource Description Framework - RDF, el RDF Schema - RDFS y Web Ontology Language -OWL y son gestionados por un framework semántico.
- Capa de núcleo: que está compuesto por el módulo de indexación que implementa la submission y la indexación de los SDKE; el Módulo de Contexto que captura el contexto de trabajo del usuario; y el Módulo de Búsqueda que utiliza búsquedas semánticas para encontrar el conocimiento relevante.
- Capa de Interfaz que está compuesto por un conjunto de herramientas para gestionar el sistema y su base de conocimiento.

En (Antunes et al, 2007) se proporciona una forma de reutilizar y compartir el conocimiento a los desarrolladores software utilizando el contexto de los usuarios a través de tecnología basada en Web Semánticas. Con esta solución únicamente se almacena el conocimiento de los elementos realizados por los desarrolladores y su contexto. Es decir, únicamente almacenan los productos finales y el contexto a través del cual se han obtenido, sin proporcionar los activos de proceso basándose en el conocimiento de las mejores prácticas de la ingeniería del software y del conocimiento tácito que los ingenieros expertos de las organizaciones tienen en el uso de la ingeniería del software para obtener dichos activos de proceso.

Lo mismo pasa en (Brunner et al, 2007), en este trabajo se centran en cómo usar la tecnología de web semánticas para gestionar la información de los productos. Para ello definen una metamodelo y una arquitectura basada en un repositorio ontológico. El objetivo de este trabajo (Brunner et al, 2007) no es proporcionar el conocimiento para poder desarrollar dichos activos, sino la reutilización de los productos desarrollados.

2.1.1.2 Repositorios de Experiencia

Basili, en 1994 (Basili et al, 1994), creó el concepto de “Experience Factory” para aquellas organizaciones cuyo aprendizaje está basado en repositorios de experiencia. Desde entonces muchas investigaciones se han enfocado en cómo implementar los repositorios de experiencia (Dingsoyr, 2000) (Feldmann et al, 2003) (Basili et al, 2007).

En (Feldmann et al, 2003) utiliza un repositorio de experiencia para almacenar el conocimiento y experiencia de una organización para su posterior reutilización. En dicho estudio se centran en cómo implantar nuevos esquemas de repositorio de experiencia. No detallan cómo esos repositorios son usados ni cómo se reutiliza el conocimiento de los repositorios de experiencia.

El problema encontrado en los trabajos relacionados con los repositorios de experiencia es que sí almacenan el conocimiento y la experiencia extraída de los ingenieros expertos pero ese conocimiento no es reutilizado en los diferentes proyectos.

2.1.1.3 Wikis en ingeniería del software

En (Chau et al, 2005) se propone que los repositorios de experiencia sean representados por tecnología Wiki (Cunningham et al, 2001). De este estudio se extrae que tienen una gran aceptación las wiki como herramientas de compartición de conocimiento en las organizaciones. Además es notable la gran necesidad de que el conocimiento no estructurado de las organizaciones sea almacenado para su posterior reutilización.

Petter y Vaishnavi (Petter et al, 2008) investigaron cómo resolver que los jefes de proyecto software tuvieran la suficiente experiencia para gestionar un proyecto adecuadamente, ya que es una de las causas que motiva que solo el 29% de 9000 proyectos de desarrollo

software (Standish Group, 2004) sean desarrollados en tiempo, coste y con las funcionalidades deseadas. Para ello, se centraron en intentar facilitar la reutilización de la experiencia obtenida en el desarrollo de proyectos software entre los jefes de proyectos. Para solucionar dicho problema, propusieron crear bibliotecas de intercambio de experiencia (Experience Exchange Library) y una comunidad de práctica especializada en gestión de proyectos (Project Management Community):

- Las bibliotecas de intercambio de experiencia ayudan a capturar y almacenar el conocimiento basado en experiencias expresadas de forma narrativa. El conocimiento que se almacena en dichas bibliotecas es, el “problema” a resolver, las “lecciones aprendidas” y la “historia” en la que se detalla la experiencia obtenida de dicho desarrollo de forma textual.
- Las comunidades de práctica son una iniciativa para compartir el conocimiento tácito (Woo et al, 2004). Estas comunidades son una red de expertos en algo que trabajan juntos para aprender y resolver problemas complejos cuando se necesitan soluciones. En este caso, la comunidad de práctica está enfocada para el intercambio de conocimiento referente a la gestión de proyecto.

La plataforma tecnológica para el desarrollo de la biblioteca y de la comunidad fue una wiki debido a que permite que las personas que utilizan dicha tecnología puedan editar y modificar su contenido. Aunque en dicho trabajo (Petter et al, 2008) se hace énfasis en que la solución que aportan para la transferencia y reutilización del conocimiento está en formato textual y por tanto es un formato fácil de entender y de obtener; dicho formato no va a permitir automatizar la reutilización del conocimiento en proyectos futuros, ya que tal y como proponen en dicha solución, el jefe de proyecto tendrá que leer todas las páginas creadas de múltiples proyectos que tengan en común el problema para el cual se está buscando las experiencias pasadas, estudiar la solución que se llevó a cabo en cada una de ellas y extraer el conocimiento que pueda aportar a su proyecto. Sin embargo, lo que se propone en esta tesis doctoral es que según las características del contexto y las restricciones de la actividad a realizar para obtener un determinado producto, se proporcione los pasos a seguir para desarrollar dicha tarea, así como información adicional como por ejemplo las lecciones aprendidas de haber ejecutado dichos pasos en diferentes proyectos. Además de que esta

información se encuentre de forma sistemática, los componentes del equipo de desarrollo podrán acceder a una wiki en la que toda esta información está contenida.

2.1.1.4 Librería de Activos de Procesos

Las librerías de activos de proceso (PAL) son definidas (Suzanne, 2006) como un repositorio organizado, bien indexado, de activos de proceso que sean fácilmente accesible por cualquier persona que necesite información del proceso, como son ejemplos, datos, plantillas o cualquier otro material de los proyectos software.

Las librerías de activos de proceso proporcionan:

- un repositorio de conocimiento central para adquirir, definir y difundir las guías sobre procesos relacionados para las actividades de la organización.
- la reducción de duplicados innecesarios de activos de procesos de la organización.
- mecanismos para compartir el conocimiento sobre los activos de proceso y cómo éstos son usados.
- un entorno de aprendizaje de los procesos de la organización.

Existen algunos trabajos relacionados con las PAL en el área de la ingeniería del software, como son:

- Goddar Space Flight CenterGSFC (<http://software.gsfc.nasa.gov/process.cfm>)
- Sub-web of SSC San Diego Public Web Site (<http://sepo.spawar.navy.mil/>)

En estas PAL los tipos de activos que se almacenan son: guías, listas, formularios, plantillas, procedimientos, subprocesos, procesos y políticas.

Las funcionalidades que aporta una PAL a una organización, están incluidas en las características y objetivos que se ha definido en esta tesis doctoral, es por ello que en la solución propuesta en esta tesis doctoral, se define una PAL con los activos de procesos reutilizables de dicha solución.

2.1.1.5 Guías Electrónicas de Proceso

La guía electrónica de proceso (EPG) fue definida por (Kellner et al, 1998) como documentos orientados a flujo de trabajo, estructurados para poder llevar a cabo un proceso. Una guía electrónica de proceso detalla los siguientes campos de un proceso: sus actividades, artefactos, agentes, roles y recursos que están involucrados en el proceso y la relación entre ellos.

Las guías electrónicas de proceso han sido aplicadas en diferentes líneas de investigación. Una de las líneas de investigación es la mejora de procesos software, donde se encuentra el trabajo de Moe (Moe et al, 2006). En dicho trabajo (Moe et al, 2006) se realizó una encuesta a las personas involucradas en el desarrollo de un proyecto software para valorar cómo percibían varias características de calidad en las guías electrónicas de proceso (la eficiencia de uso, el aprendizaje del proceso y su utilidad). Las conclusiones a la que llegaron en dicho trabajo (Moe et al, 2006) es que las guías electrónicas de proceso hay que adaptarlas a los procesos de la organización ya que si no, no son útiles para quienes las utilizan. Sobre la facilidad de uso, se obtuvieron diferentes resultados dependiendo del perfil de los empleados, para los jefes de proyecto la EPG era fácil de utilizar, pero para los desarrolladores era difícil de utilizar y aprender la sistemática de la guía. Como conclusiones finales se puede extraer que una guía electrónica de proceso está enfocada a los procesos y no a los productos y además tienen que ser adaptadas a las necesidades de la empresa.

En esta tesis doctoral se propone utilizar una guía electrónica de proceso combinada con los *patrones de producto*, es decir, se le proporciona al equipo de desarrollo una guía de proceso con la información contenida en los *patrones de producto* para que el cada uno de los roles del equipo tenga en todo momento los pasos que tiene que seguir para poder obtener el producto deseado.

2.1.2 Trabajos relacionados con la Gestión de Conocimiento aplicado a desarrollos de proyectos.

2.1.2.1 Gestor de Conocimiento basado en actividades

Muchos ingenieros y expertos piensan que la gestión de conocimiento es necesaria y beneficia la construcción de un proyecto. Sin embargo no existen plataformas adecuadas que

asistan a los ingenieros seniors ni expertos en compartir y recoger su know-how y experiencias cuando se está ejecutando un proyecto. El propósito del estudio (Tseng et al, 2008) fue desarrollar un gestor de conocimiento basados en actividades (ABKM) para construir proyectos software. La información y el conocimiento de dominio son divididos y guardados como actividades y relacionados a los proyectos. La representación del conocimiento se realiza mediante mapas de conocimiento que es una representación gráfica de nodos en el que se define diferentes atributos de conocimiento, ver Figura 2-2.

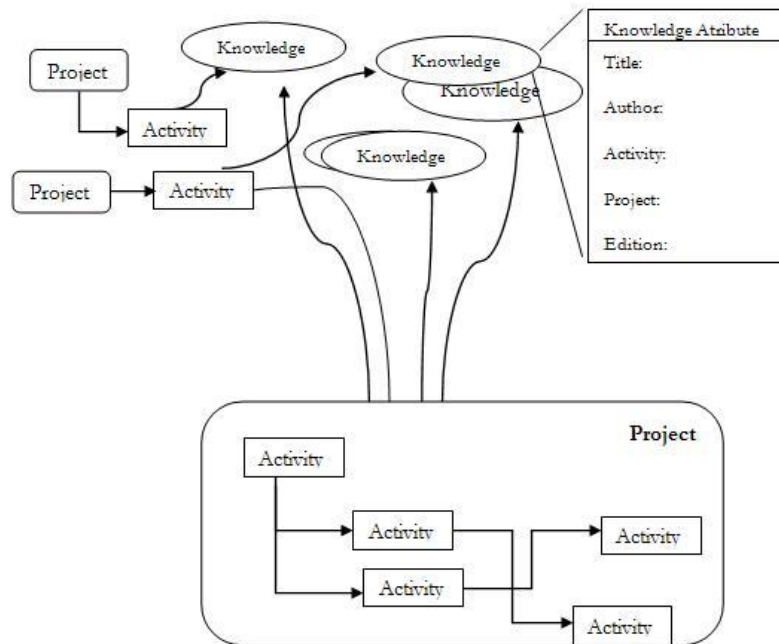


Figura 2-2 Mapa conceptual de la gestión de conocimiento basado en actividades.

Las fases que en este sistema se realiza son:

- Fase de adquisición del conocimiento: en el sistema ABKM se realiza la recolección de documentos, información y datos de todas las actividades relacionadas en un proyecto.
- Fase de extracción del conocimiento: se extraen el conocimiento de los ingenieros expertos de cada proceso definido en cada actividad relacionada.
- Fase de almacenamiento del conocimiento: el conocimiento es almacenado en mapas de conocimiento y relacionado con los proyectos.
- Fase de compartición del conocimiento: en un nuevo proyecto los ingenieros realizarán proceso de búsqueda para encontrar los mapas de conocimiento adecuados, estudiarlos para poder reutilizar dicho conocimiento en su propio proyecto.

- Fase de modificación del conocimiento: una vez aplicado el conocimiento en proyectos similares el conocimiento es almacenado.

Al igual que en (Tseng et al, 2008) en esta tesis doctoral se considera que el área de gestión de conocimiento es fundamental para la difusión, uso y reutilización de activos de procesos en la realización de un proyecto software. A diferencia que en (Tseng et al, 2008), en la presente tesis doctoral no se considera la actividad la unidad para la construcción de proyectos software, sino los productos. Como bien se explica en (Tseng et al, 2008) es fundamental tener el conocimiento para poder realizar una actividad, pero realmente en la ejecución de un proyecto y tras realizar una actividad o un conjunto de actividades, se obtienen productos. Es por ello que este trabajo ha identificado como unidad mínima que proporcione al sistema la característica de interoperabilidad entre diferentes proyectos software, los productos.

Las fases identificadas en el ciclo de la gestión de conocimiento de esta tesis doctoral son las enunciadas en (Tseng et al, 2008) con aportaciones a cada una de dichas fases. A continuación son detalladas las aportaciones realizadas:

- Fase de adquisición y extracción del conocimiento: en esta fase, además de realizar la recolección de documentos, información y datos de todas las actividades relacionadas en un proyecto, se ha realizado la adquisición y extracción del conocimiento de las mejores prácticas de la ingeniería del software de los ingenieros expertos.
- Fase de almacenamiento del conocimiento: el conocimiento es almacenado en artefactos denominados *patrones de producto* que proporciona a los proyectos de la capacidad de reutilización mediante parámetros propios del proyecto.
- Fase de compartición del conocimiento: al igual que en (Tseng et al, 2008) se podrá realizar proceso de búsqueda para encontrar los patrones de producto más adecuados. En (Tseng et al, 2008) para poder reutilizar el conocimiento de los expertos es necesario seguir el mapa de conocimiento y leer todos los documentos anexados, estudiar la solución dada de forma textual en dichos documentos y reutilizar dicho conocimiento en el proyecto a desarrollar. En esta tesis doctoral, al utilizar los patrones de producto, directamente se le proporciona al ingeniero la

solución al problema a resolver de una forma gráfica donde se le detallan todos los pasos que tiene que realizar para obtener el producto deseado en el proyecto.

- Fase de modificación del conocimiento: una vez aplicado el conocimiento en proyectos similares el conocimiento es almacenado.

2.1.2.2 Living Software Development Process

En (Gnatz et al, 2003) se presenta un framework de modelos de proceso que proporciona la capacidad de incorporar el conocimiento de los procesos de la organización mediante un proceso denominado Living Software Development Process. Este proceso está basado en el concepto de patrón de proceso que almacena el conocimiento de las metodologías y modelos de proceso para que puedan ser utilizadas en el desarrollo de proyectos software.

El framework propuesto permite al equipo de desarrollo interactuar para elaborar y mejorar los procesos de desarrollo software estándar mediante un proceso llamado Living Software Development Process, ver Figura 2-3. Dicho proceso permite definir y mantener un modelo de proceso que integre artefactos de trabajo de todos los modelos de procesos existentes.

- Seleccionar y personalizar procesos de desarrollo.
- Proporcionar los artefactos de trabajo de dichos procesos de desarrollo.
- Crear, modificar y almacenar artefactos de trabajo.
- Retroalimentar los procesos de desarrollo para su posterior reutilización.

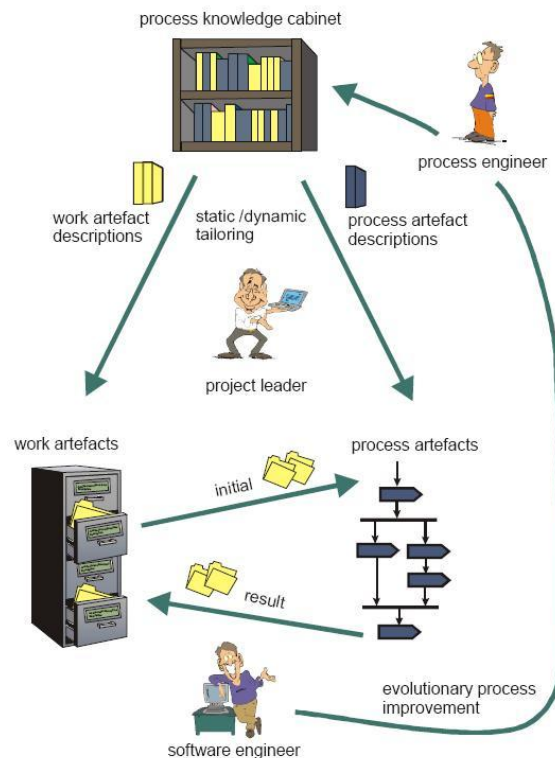


Figura 2-3 Proceso Living Software Development Process (Gnatz et al, 2003)

Los artefactos de trabajo son todo tipo de documento que se necesitan y producen en un proceso de desarrollo. Los artefactos de proceso son las tareas que se realizan en el desarrollo de un proceso de desarrollo y en el que son utilizados y producidos los artefactos de trabajo.

Los artefactos de proceso están basados en patrones de procesos. Cada patrón de proceso va a proporcionar un diagrama de actividad en el que se le muestra al usuario el conjunto de actividades que tiene que realizar en ese proceso.

En este trabajo de investigación se proporciona al equipo de desarrollo los procesos de desarrollo así como los patrones de proceso que les indican las actividades que tienen que realizar para realizar dicho proceso. Esta solución está enfocada a procesos y no a productos, es decir, los artefactos de proceso proporcionan al usuario las actividades que tiene que realizar para realizar un proceso de desarrollo, esta solución minimiza la interoperabilidad entre proyectos software.

2.1.3 Trabajos relacionados con áreas relacionadas con la gestión del conocimiento

La gestión de conocimiento es un área que está relacionada con muchas otras áreas, como la ciencia cognitiva, inteligencia artificial, sistemas expertos, entornos colaborativos, y otras muchas áreas que se pueden ver en (Gupta et al, 2008).

De todas estas áreas en la presente tesis doctoral la gestión de conocimiento se relaciona con la mejora de procesos software y con entornos colaborativos. A continuación se van a detallar los trabajos relacionados en los que dichas áreas están implicadas.

2.1.3.1 Gestión de Conocimiento y Mejora de Procesos

Desde el año 1984, cuando el Instituto de Ingeniería del Software (SEI) de la Universidad Carnegie Mellon y el departamento de defensa de EEUU proporcionaron modelos de procesos para la mejora de la calidad de sistemas software, muchos autores han investigado cómo mejorar dichos procesos. Diversas investigaciones muestran cómo los proyectos que han utilizado modelos de procesos software y estándares, ayudan a mejorar la calidad del software, reducir coste y tiempo, e incrementan la productividad. (Butler, 1995) (Yamamura, 1999) (Pitterman, 2000) (Niazi et al, 2006). A pesar de los beneficios que aportan el uso de las mejores prácticas en los proyectos, el 67% de los gestores de mejora de procesos demandan guías que describan cómo implementar las actividades de mejora en los proyectos ya que una de las grandes deficiencias de los programas de mejora es que detallan qué tienes que hacer pero no cómo hacerlo. Por ello, se considera en esta tesis doctoral imprescindible la interacción entre el área de mejora de procesos software y el área de la gestión de conocimiento. La solución propuesta en esta tesis doctoral apuesta por proporcionar el conocimiento de las buenas prácticas de la ingeniería software para que las personas involucradas en el desarrollo de proyectos y procesos software sepan cómo tienen que realizar dichas actividades.

Un uso de la gestión del conocimiento es soportar las actividades que apoyan la mejora de procesos software. Basados en la experiencia de la Federal Aviation Administration (FAA) de los EEUU, la gestión de conocimiento combinada con la mejora de proceso proporciona una interacción positiva que beneficia a la organización y a los programas de mejora. (Burke et al, 2005) (Feher et al, 2006) (Dingsoyr et al, 2009).

Algunos investigadores han centrado sus esfuerzos en investigar la interacción de la gestión de conocimiento y la mejora de procesos software; a continuación se detallan algunos trabajos.

En (Stelzer et al., 1998) se centraron en cómo los principios y técnicas de aprendizaje de la organización se pueden aplicar a proyectos de mejora de procesos software. Baskerville (Baskerville et al, 1998) sugirió incluir la práctica de gestión del conocimiento en modelos de madurez. Arent (Arent et al, 2000) describe como el conocimiento es creado, usado y difundido en el proceso del cambio organizacional con el objetivo de mejorar la eficiencia y calidad del software desarrollado. En esa investigación se analizaron tres proyectos de mejora de procesos software en organizaciones desde la perspectiva de la creación del conocimiento. Los resultados de la investigación concluyeron que el conocimiento se crea de la interacción entre el conocimiento tácito y el explícito. El cambio más inmediato ocurrió cuando los desarrolladores software combinaron las prácticas existentes con nuevas ideas y sugerencias del equipo de mejora de procesos. El problema encontrado fue la difusión del conocimiento en los diferentes niveles organizativos.

En (Komi-Sirviö et al, 2002), se confirma cómo la ingeniería del software y las técnicas de gestión fallan si no están basados en conocimiento. En dicho estudio se analizó el estado de la mejora de procesos software basado en el conocimiento y corroboraron que el conocimiento en la organización no era fácil de localizar y cuando se localizaba no se podía reutilizar. En dicha investigación se encontraron dos motivos por los que el conocimiento no se podía reutilizar: el proceso de capturar el conocimiento era demasiado informal y por tanto el conocimiento no era incorporado en los procesos de ingeniería ni era soportado por la estructura de la organización.

En (Falbo et al, 2004) se presenta una herramienta basada en la gestión de conocimiento para soportar la mejora de procesos software. Dicha herramienta está basada en dos ontologías, una ontología de procesos software y una ontología de métricas de software cuya estructura fue almacenada en repositorios. Algunos de los problemas encontrados en la herramienta fue la dificultad de encontrar proyectos similares. En estos trabajos se usa la gestión de conocimiento para capturar, almacenar, difundir y reutilizar el conocimiento creado durante el proceso software, lo que proporciona una mejora en la calidad y productividad.

En los trabajos de (Niazi et al, 2006) y (Brede et al, 2006) se pone de manifiesto la importancia que tiene la difusión del conocimiento de los procesos a los participantes en el desarrollo de los proyectos en la mejora de procesos software. En dichos trabajos se presentan las guías electrónicas de proceso como mecanismo para almacenar la información relacionada con los procesos.

En una organización, los ingenieros, para desarrollar su trabajo, buscan en fuentes de conocimiento como son libros, recursos on-line, bases de datos, etc, el conocimiento para poder realizar un determinado proceso. Pero en muchas ocasiones, el conocimiento necesario para poder desarrollar su proceso correctamente, está en el conocimiento tácito de la empresa, en el know-how que no ha sido guardado en formato electrónico y al cual no se puede acceder (Assimakopoulos et al, 2006). En dicho trabajo aunque se pone de manifiesto la importancia de tener un artefacto para almacenar el conocimiento tácito y cómo dicho artefacto tiene que permitir la reutilización del conocimiento tácito en los diferentes procesos de proyectos software, no proponen ningún artefacto para tal fin.

De los trabajos antes mencionados se puede extraer que la gestión de conocimiento mejora la calidad y productividad de los proyectos software. Dichos trabajos no proporcionan los procesos ni mecanismos para que las organizaciones puedan adquirir, difundir, usar y reutilizar el conocimiento de las mejores prácticas de la ingeniería del software y el conocimiento de los procesos de su propia organización. A diferencia de los trabajos anteriores, esta tesis doctoral define un artefacto y sus mecanismos de colaboración para el almacenamiento y reutilización del conocimiento de las mejores prácticas de la ingeniería software así como del conocimiento tácito de las organizaciones de una forma sistemática y eficiente. Este trabajo doctoral se centra en que las organizaciones puedan utilizar el conocimiento de los expertos a través de un elemento de conocimiento que se ha definido en el presente trabajo de investigación para que puedan reutilizarlo y llegar al estadio de innovación.

2.1.3.2 Gestión de Conocimiento y Entornos Colaborativos

El término sistemas colaborativos se creó en los años 80. El término “trabajo colaborativo” denominado en inglés, Computer Supported Cooperative Work (CSCW), lo inventaron Paul Cashman e Irene Greif (Greif, 1988) en un workshop en el Massachusetts Institute of Technology en 1984. El objetivo de este workshop fue reunir a personas de varias

disciplinas, que compartían un interés común en cómo la gente trabaja y cómo las tecnologías pueden soportar el trabajo cooperativo. Ellos utilizaron el término CSCW para describir este interés común. Desde 1986, el término CSCW ha sido debatido en múltiples investigaciones. En 1987 (Vygotsky, 1987) Vygotsky dijo que “la colaboración entre pares ayuda a desarrollar estrategias y habilidades generales de soluciones a problemas en el proceso cognitivo implícito en interacción y comunicación”. En el año 1992, Marca (Marca, 1992) señalaba que cada vez el trabajo colaborativo era más importante por los siguientes motivos:

- Los problemas son cada vez más complejos y de mayor tamaño.
- Las personas están cada vez más especializadas.
- Las soluciones cada vez requieren aptitudes diferentes, lo que provoca un cambio de actitud en la forma de trabajar. Esto se visualiza en que las personas necesitan intercambiar información cada vez más frecuentemente y en el que se está viendo la necesidad de trabajar como miembros de un equipo, cooperando entre ellos. Por lo tanto el éxito de un equipo de trabajo dependerá no sólo de las aptitudes individuales sino del nivel de colaboración.

El término CSCW es introducido como “una vía para describir cómo la tecnología de los computadores puede ayudar a los usuarios a trabajar juntos en grupos” (Crowe, 1994). De acuerdo con Carstensen and Schmidt, CSCW aborda cómo las actividades colaborativas y su coordinación se soportan mediante sistemas informáticos (Carstensen et al, 2002).

En la última década, los sistemas basados en entornos colaborativos están teniendo un importante auge debido a que sus capacidades les hacen especialmente útiles en tareas en las que la interacción persona-persona a través de un sistema informático es un factor crucial.

Los Sistemas basados en entornos colaborativos, son ideales para grupos de personas, dentro del ámbito profesional, industrias, investigación, etc., en los que existen necesidades importantes de interacción. Estos sistemas se han venido usando en entornos como la medicina (Pratt et al, 2004), la industria (D’Souza et al, 2003) y en diferentes ramas interdisciplinarias de la ingeniería civil (Garner et al, 2003). En el ámbito de la enseñanza, los entornos colaborativos se han aplicado a entornos de aprendizaje asíncrono (Dewiyanti et al, 2004) y a entornos virtuales que utilizan herramientas colaborativas (Dave et al, 2000).

Las herramientas colaborativas deben ser herramientas familiares e intuitivas de manejar, que permitan la gestión de la información de forma productiva y que permita la toma de decisión de forma segura. Coleman explicó las razones por las cuales se debería implantar un Groupware (Coleman, 1997):

- Mejor control de costos.
- Incremento de la productividad.
- Mejor servicio al cliente.
- Pocas reuniones.
- Automatización de procesos de rutina.
- Extensión de la organización: se incluye tanto al cliente como al proveedor.
- Integración de grupos geográficamente dispersos.
- Mejor coordinación globalmente.
- Proveer mejores servicios que diferencien a la organización.

Es necesario dinamizar el conocimiento a través de la interacción y la conexión entre distintas unidades de negocio, grupos funcionales, proveedores, etc, facilitando su conectividad tecnológica (creando las plataformas e infraestructuras necesarias como intranets, extranets, comunidades virtuales) y humana que inevitablemente conlleva un grado de negociación entre las distintas comunidades y grupos, y que se traduce en ciertas normas, valores sobreentendidos e intereses.

Fomentar el espíritu innovador en una organización supone facilitar el ciclo del conocimiento creando las plataformas tecnológicas y los mapas de conocimiento necesarios; y en lo que se refiere a la dimensión humana, rompiendo inercias y evitando la monopolización de los conocimientos clave para favorecer la retroalimentación. Se trata, en definitiva, de apoyar la construcción de un entorno de aprendizaje organizacional.

Entre sus principales tareas al servicio del aprendizaje en una organización se encuentran las de localizar y comunicar entre sí a los integrantes de una organización, establecer sistemas de gestión documental eficaces y accesibles a todos los miembros de la empresa, trabajar en un entorno de red óptimo para el intercambio de información interna y externa y, crear y desarrollar intranets, extranets y aulas virtuales de formación.

Para que se lleve a cabo el aprendizaje en las empresas es necesario tener una visión de conjunto de todos los elementos que intervienen en el conocimiento y en el aprendizaje, fomentar actitudes de colaboración, trabajo compartido y entender que el equipo es el elemento fundamental de aprendizaje.

Las organizaciones necesitan disminuir los costes y tiempos e incrementar la calidad de los desarrollos de proyectos software. Por lo que, las organizaciones necesitan aplicar el conocimiento ganado en proyectos previos, en proyectos futuros. Desafortunadamente, los equipos de desarrollo no se benefician de las experiencias anteriores y repiten los mismos errores, incluso cuando algunos individuos de la organización saben cómo evitarlos. La organización y los individuos en una organización pueden ganar mucho más si comparten el conocimiento. Las organizaciones necesitan definir procesos para compartir conocimiento, para que los empleados puedan tomar decisiones teniendo toda la información. Para ello, las organizaciones deben adquirir conocimiento sobre nuevas tecnologías que permitan la colaboración, coordinación y compartición de conocimiento a través de estos sistemas sin que el uso del mismo repercuta en tiempo y coste para los empleados.

Aunque las nuevas tecnologías permiten poder compartir conocimiento fácilmente, la cultura de la organización puede que no lo promueva. Algunas culturas empresariales defienden el individualismo y prohíben trabajar colaborativamente. Esta carencia es uno de los principales motivos de fracaso en la gestión del conocimiento. Si las organizaciones no fomentan una cultura de compartición, los empleados se sienten en posesión de su conocimiento sin llegar a compartirlo. Por lo que aunque el cambio es difícil se debe crear y reemplazar a una actitud positiva hacia la compartición (Rus, 2002).

La ventaja competitiva de las empresas reside en el capital intelectual o el conocimiento, siendo las personas las que pasan a ser el principal activo de las empresas. Es por ello, que la empresa tiene que aprovechar las capacidades intelectuales de la organización, fomentándolo, y potenciando la creación de nuevos conocimientos

2.2 REUTILIZACIÓN: PATRONES

Como se ha visto en el apartado anterior, el área de la gestión de conocimiento desde el punto de vista de la Ingeniería del Software, es relevante para esta tesis doctoral. Este trabajo pretende proporcionar una solución práctica a la difusión, uso y reutilización del conocimiento de las mejores prácticas así como el conocimiento tácito de las organizaciones, para facilitar la eficiencia de uso de los procesos en las organizaciones. Por ello, se necesita un artefacto que permita dar una solución práctica a un problema bajo un contexto y unas restricciones, y que tenga la característica principal de la reutilización, como son los patrones desde la perspectiva de la Ingeniería del Software.

El concepto de patrón ha sido definido por numerosos autores, pero todos ellos parten de la definición original dada por Alexander en 1979:

“Un patrón describe un problema que ocurre una y otra vez en nuestro entorno, describiendo el núcleo de la solución a dicho problema de tal forma que se puede usar esta solución un millón de veces sin hacerlo dos veces de la misma forma”. (Alexander, 1979).

En el campo del software, los patrones representan soluciones a problemas que surgen en el desarrollo de aplicaciones, permitiendo su reutilización como respuesta a conflictos similares surgidos en contextos diferentes. La clave está en que el patrón es una generalización o abstracción reusable que puede utilizarse como punto de partida para soluciones futuras. El hecho de tener en consideración este aspecto es importante ya que lo que se pretende con esta tesis doctoral es facilitar la difusión, uso y reutilización de activos de procesos, por ello, la utilización de patrones como el artefacto que almacene el conocimiento de las mejores prácticas de los ingenieros expertos y del conocimiento tácito de las organizaciones para su posterior reutilización, se adecua a las necesidades de esta tesis doctoral.

Los patrones han sido ampliamente utilizados en muchas áreas de la ingeniería. En la década de los 90 los patrones tuvieron gran popularidad en el campo de la programación. En dicho campo se han utilizado los patrones para capturar y organizar importantes soluciones de ingeniería. Actualmente, los patrones están siendo utilizados en muchas soluciones aplicadas al campo de la Ingeniería del Software (Coplien, 2004).

Dentro del amplio abanico que abre la investigación en el campo de los patrones software, se han clasificado los formatos más conocidos y los trabajos más relevantes según los tipos de patrones existentes, comparando cada uno de los tipos de patrones con el problema que se trata de resolver en este trabajo.

2.2.1 Descripción de Formatos de Patrones más comunes

La estructura de un patrón se define como el formato del patrón. Un formato de un patrón es simplemente una plantilla dividida en secciones fijas, con una estructura uniforme que facilita su aprendizaje, comparación y uso. No existe un único formato de patrón (Fowler, 2003), dicho formato dependerá del autor que lo cree, pero en todos ellos debería haber dos componentes vitales: el cómo (¿Cómo se implementa dicho patrón?) y el cuándo (¿Cuándo se puede utilizar dicho patrón?). Aunque no existe un único formato para la descripción del patrón, existen varios formatos muy extendidos, siendo los más habituales y conocidos: el formato Alexander (Alexander, 1977), el formato GoF (Gamma et al, 1994) y el formato canónico (Buschmann et al, 1996).

2.2.1.1 Formato Alexander

El formato Alexander (Alexander et al, 1977) consiste en la exposición en estilo narrativo del problema que se pretende resolver, el contexto en el cual se aplica el patrón y la solución propuesta. Un formato típico utilizado por Alexander se compone de los siguientes elementos:

- Name
- Problem
- Context
- Forces
- Solution

Nótese que la descripción de este formato es la original, definida por Alexander, creador del concepto de Patrón. Tal y como define Alexander, este formato de patrón es una exposición narrativa de los campos que lo forma. Ésta característica es una deficiencia para la aproximación de esta tesis doctoral, ya que se pretende una solución práctica que aporte el conocimiento de las mejores prácticas de la ingeniería del software y de los ingenieros expertos involucrados en el desarrollo de los procesos software de una forma eficiente y sistemática.

2.2.1.2 Formato GoF

Plantilla definida por los autores Erich Gamma, Richard Hel, Rala Johsson y John Vlissides, conocidos como el “Gand of Four” o “Club de los Cuatro” para describir los patrones catalogados en su libro (Gamma et al, 1994). La plantilla respeta el formato representado en la Tabla 2-1.

Sección	Descripción
Pattern name and classification	Expresa la esencia del patrón. Se define su propósito y ámbito.
Intent	Breve introducción que responde a las preguntas: ¿Qué hace el patrón? ¿Cuál es su razón? ¿Qué problema resuelve?.
Also Known As	Otros nombres dados al patrón.
Motivation	Escenario que ilustra el problema.
Applicability	Situaciones en las cuales el patrón puede ser aplicado. Pautas para reconocerlas.
Structure	Representación gráfica de las clases que conforman el patrón. Diagrama de clases de colaboración, etc.
Participants	Clases y/u objetos que participan en el patrón y sus responsabilidades.
Collaborations	La forma en la cual los participantes colaboran para realizar el cometido.
Consequences	¿Cómo soporta el patrón sus objetivos? ¿Cuáles son sus concesiones?.
Implementation	Trucos o técnicas que deben considerarse en la implementación.
Sample Code	Fragmentos de código que ilustran la implementación del patrón.
Related Patterns	Enumeración de otros patrones relacionados con el que se describe. Diferencias entre los mismos.

Tabla 2-1 Formato GoF.

2.2.1.3 Formato Canónico

Modelo definido por Buschmann (Buschmann et al, 1996) para la descripción de patrones. Básicamente, contempla las mismas secciones que el formato Alexander (motivo que ha llevado a algunos autores a utilizarlos indistintamente) aunque con mayor detalle y precisión. También es conocido como formato POSA (Tabla 2-2).

Sección	Descripción
Name	Nombre significativo. Puede ser una única palabra o una frase corta que haga referencia al patrón.
Problem	Descripción del problema indicando su propósito.
Context	Precondiciones bajo las cuales se puede aplicar el patrón.
Forces	Descripción de las restricciones relevantes y cómo interaccionan o entran en conflicto unas con las otras. Definición de los objetivos a conseguir.
Solution	Relaciones estáticas y reglas dinámicas que describen cómo conseguir el objetivo marcado.
Examples	Uno o más ejemplos de la aplicación del patrón.
Resulting Context	Estado o configuración del sistema después de haber aplicado el patrón. Describe tanto los efectos positivos como perniciosos.
Rationale	Explicación justificada de los pasos o reglas del patrón.
Related Pattern	Relaciones estáticas o dinámicas entre el patrón que se describe y otros.
Known uses	Descripción de ocurrencias conocidas del patrón y su aplicación dentro de sistemas existentes.

Tabla 2-2. Formato canónico.

2.2.2 Tipos de patrones software

En los últimos años el número de trabajos, estudios y publicaciones sobre los patrones se ha incrementado exponencialmente.

Dado que el concepto de patrón ha sido adoptado por diferentes disciplinas existen multitud de tipos de los mismos. Esta tesis doctoral describe los principales tipos de patrones clasificados en dos grandes grupos: los que han sido definidos en el marco de la disciplina de la Ingeniería del Software y los que salen de los márgenes de la misma. Además dependiendo del uso y del nivel de abstracción de cada tipo de patrón se han ubicado como muestra la Figura 2-4.

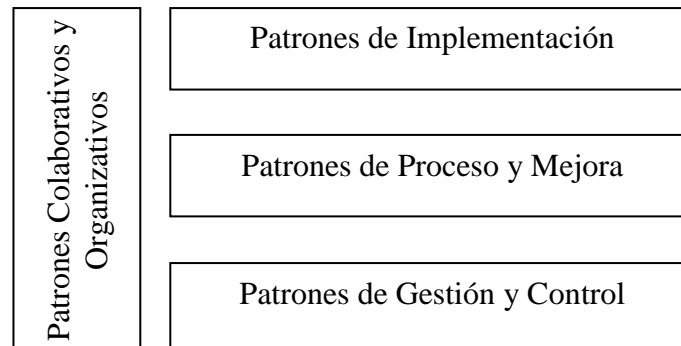


Figura 2-4 Marco de Patrones Estudiados

Dentro de la ingeniería de software se han identificado ocho tipos de patrones en tres niveles de abstracción diferentes y con dominios de conocimiento diferente:

1. Patrones de Implementación. El dominio de aplicación de cada tipo de estos patrones va siendo cada vez más detallado en estos cuatro tipos de patrones, desde subsistemas, pasando por estructuras de componentes, estructuras de datos y, objetos o clases, como puede verse en detalle en la Tabla 2-3.
 - a. Arquitecturas de referencia.
 - b. Patrones de Arquitectura.
 - c. Patrones de Análisis.
 - d. Patrones de diseño.

Tipos	Patrones de Diseño	Patrones de Análisis	Patrones de Arquitectura	Arquitecturas de Referencia
Nivel	Estructura dentro de un componente	Conocimiento del Dominio	Estructura entre componentes	Subsistemas
Entidades involucradas	Clases u objetos	Clases o estructuras de datos (DB)	Componentes	Subsistemas
Efecto en la estructura del sistema software	Pequeño	Medio	Grande	Muy Grande

Tabla 2-3 Nivel de Impacto de los patrones de Implementación en el desarrollo de un sistema.

2. Patrones de Proceso y Mejora. Los patrones catalogados en Proceso y Mejora tienen una perspectiva de desarrollo global de un sistema y también proporcionan guías específicas para llevar a cabo programas de mejora. Es por ello que podrían soportar a su vez cualquiera de los patrones de Implementación identificados como se muestra en la Figura 2-4. Estos patrones se clasifican en:

- a. Patrones de Proceso.
- b. Patrones de Mejora de Procesos Software.

3. Patrones de Gestión y Control. Estos patrones están pensados para albergar conocimiento que debe ser aplicado a lo largo de todo el desarrollo de un proyecto y es por ello que se encuentra en la base de todos los demás en la Figura 2-4. Estos patrones se clasifican en:

- a. Patrones de Gestión.
- b. Patrones de Gestión de la Configuración.

Fuera de los límites de la Ingeniería del software se van a contemplar dos tipos de patrones que se ubican en vertical respecto de todos los demás en la Figura 2-4, dado que se pueden utilizar para conseguir que todos los patrones identificados colaboren entre sí. Dichos patrones se clasifican en

4. Patrones Organizativos y Patrones de Colaboración.

A continuación se describirán cada uno de los patrones identificados en la Figura 2-4 en detalle.

2.2.2.1 Patrones de Implementación

a. Arquitecturas de Referencia

El nivel de abstracción más alto en el esquema de patrones corresponde a las Arquitecturas de Referencia. Una Arquitectura de Referencia es una arquitectura que ya ha

sido creada para un dominio de interés particular. Suele incluir varios estilos arquitectónicos, aplicados en diferentes áreas de su estructura.

Los componentes que contemplan y que manejan las Arquitecturas de Referencia son subsistemas, los cuales han podido ser diseñados según Patrones de Arquitectura (el nivel de abstracción justo por debajo), los cuales a su vez pueden estar compuestos por modelos y por componentes diseñados conforme a los patrones de nivel de abstracción inferiores (patrones de diseño y patrones de análisis).

Existen múltiples ejemplos de patrones de Arquitecturas de Referencia, si bien no están tan categorizados como los patrones de diseño o análisis. Una lista bastante amplia sería la siguiente: Dos capas (Two tier), Multi-Capa (Multi-tier), Workflow, Basada en Documentos (ej. Notes), Orientada a Formularios, Orientada a Hipertexto, Video/Imagen/Multimedia, Datawarehouse, Sistemas de Tiempo Real, Control de Procesos, Modelo Transaccional de Cliente Delgado.

Sin embargo, este patrón tiene inconvenientes tales como que el desarrollo es más difícil por la falta de herramientas totalmente integradas para los distintos subsistemas y tiene riesgos adicionales: la estandarización de las conexiones internivel no está del todo estabilizada, y además es obvio que cuanto más niveles, se puede esperar peor rendimiento.

b. Patrones de Arquitectura

Un patrón de arquitectura expresa un esquema estructural fundamental de una organización en los sistemas de software. Proporciona un conjunto de subsistemas predefinidos, especifica sus responsabilidades, e incluye reglas y guías para organizar las relaciones entre ellos.

El patrón Model-View-Controller (MVC) es uno de los ejemplos de patrón de arquitectura más conocido. Proporciona una estructura para sistemas software interactivo (Buschmann et al, 1996).

Un patrón de arquitectura define los siguientes elementos:

- Un vocabulario de componentes y conectores.

- Una topología que describe cómo se combinan los componentes y conectores.
- Un conjunto de requerimientos y restricciones sobre cómo deben ser esas combinaciones.
- Modelos semánticos que permitan especificar el comportamiento global a partir del comportamiento de las partes.

La característica de los Patrones de Arquitectura es que permiten el diseño de sistemas enfocándose en cualidades del sistema tales como: rendimiento, fiabilidad, disponibilidad, etc.

Hay dos enfoques básicos sobre Patrones de Arquitectura debidos respectivamente a Buschmann, que define el término de patrones de arquitectura (Buschmann et al, 1996) y Shaw, que define el término de estilos arquitecturales (Shaw et al, 1996). Ambos describen la misma lista de patrones, aunque con diferencias sutiles entre ellos.

c. Patrones De Análisis

El patrón de análisis fue definido por Fowler en 1997 (Fowler, 1997) como un grupo de conceptos que representa una construcción común en el modelado de negocio. Puede ser relevante para un solo dominio de problemas o para varios. Los patrones de análisis son el tipo de patrón junto a los patrones de diseño que más son utilizados por los desarrolladores. En 2003 fue definido por Purao, Storey y Han como un grupo de objetos genéricos relacionados (meta-clases) con atributos estereotipados (definiciones de datos), comportamiento (métodos) e interacciones esperadas definidas en un dominio neutral (Purao et al, 2003).

Existen varios tipos y categorías que proceden de procesos de abstracción similares a los usados para detectar patrones de diseño. En la Tabla 2-4 se describen los diversos tipos de patrones de análisis (Fowler, 1997).

Grupo	Descripción	Patrones
Dependencias de Personas	Cuando una persona u organización es responsable de otra u otras.	Party; Organization hierarchies; Hierarchies accountability; Operating scopes
Observaciones y Medidas	Graba información sobre medidas y definición de medidas	Quantity; Conversion ratio; Measurement; Observation
Referencia de Objetos	Identificación de objetos y entidades	Name; Identification scheme; Object equivalence
Contabilidad	Gestionar el dinero que fluye en la empresa	Account; Transactions; Summary account; Posting rules
Comercio	Compra y venta de bienes. Gestión de su valor.	Contract; Portfolio; Quote

Tabla 2-4 Patrones de Análisis (Fowler, 1997)

Estos patrones representan construcciones de negocio comunes que presentan una estructura interna muy parecida. Los patrones de análisis están ligados a los interfaces o tipos de datos (tipos), no a implementaciones (clases), siendo ésta una diferencia clara con otro tipo de patrones.

Unos de los grandes problemas de los patrones de análisis, es que aunque muchas metodologías de desarrollo software están en uso, ninguna de ellas tiene perfectamente integrados el uso de dichos patrones (Kodaganallur et al, 2006).

d. Patrones de Diseño

Siguiendo al “Club de los Cuatro” podemos decir que un patrón de Diseño define, motiva y explica sistemáticamente un diseño general dirigido a un problema recurrente de diseño en sistemas orientados a objetos. Describe el problema, la solución, cuándo aplicar la solución y sus consecuencias. Suministra trucos y ejemplos de implementación. La solución es una disposición ordenada de objetos y clases que resuelve el problema y está construida e implementada para resolver el problema en un contexto (Gamma et al, 1994).

Un patrón de Diseño hace una abstracción de una estructura recurrente de diseño. Puede estar formado tanto de clases como de objetos y ser definido por las dependencias, estructuras e interacciones de ellos y por una serie de convenciones. Es una "micro-arquitectura".

Un patrón de diseño define y especifica la estructura de diseño explícitamente y compila la experiencia de diseño de software. Es, así mismo, independiente del lenguaje y la implementación.

En el trabajo (Gamma et al, 1994) se definieron 23 patrones de diseño de software, independientes del lenguaje y la plataforma, aunque se utilizó básicamente ejemplos escritos en SmallTalk, y algunos en C++. Posteriormente se han escrito esos 23 patrones de diseño con los ejemplos en Java (Cooper, 1998).

Estos patrones se enfocan sobre problemas comunes en el desarrollo orientado a objetos (OO) y son ampliamente utilizados. Pueden dividirse en tres grupos según su funcionalidad: de creación, si tienen que ver con la creación de instancias de objetos o clases; de estructura, si definen disposiciones de objetos para tareas concretas; y de comportamiento si definen como interaccionan diversos grupos de objetos para tareas concretas (Tabla 2-5).

		Propósito		
		Creación	Estructura	Comportamiento
Alcance	Clase	Factory Method	Adapter (Clase)	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Adapter (Object) Bridge Composite Decorator Flyweight Facade Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer Strategy Visitor

Tabla 2-5 Patrones de Diseño

También pueden dividirse por su alcance, ya que algunos se aplican sobre clases y otros sobre objetos.

En los patrones de diseño el formato de patrón está formado por el nombre del patrón, la descripción del problema y de la solución y las consecuencias de haber utilizado dicho patrón. Para la descripción se usan diagramas UML, sobre todo de clases (estático) y de secuencia (dinámicos). Se suelen añadir ejemplos de aplicación en algún lenguaje orientado a objetos.

La ventaja de los patrones de diseño está en:

- Producir buen diseño, utilizan la experiencia acumulada, sirven para expertos y novatos e implementan abstracción en el diseño.
- Mejorar la comunicación dentro del ciclo de vida de desarrollo, genera explícitamente nombres de estructuras de diseño, estableciendo un vocabulario común, reduciendo la complejidad.
- Capturar y preservar la información de diseño, articulando las decisiones de diseño rápidamente y mejorando la documentación.
- Facilitar la reestructuración/reorganización añadiendo flexibilidad a los diseños.

Los patrones están interrelacionados, de manera que pueden usarse como piezas de una estructura mayor como por ejemplo los patrones definidos en J2EE (Alur et al, 2003) que permiten definir, ensamblando patrones, el esqueleto de diseño de una aplicación completa.

Sin embargo, existen aspectos negativos en los patrones de diseño. Entre ellos, está la tentación de llamar patrón a todo, lo que eliminaría su capacidad de abstracción y reutilización. Para evitarlo se debe aplicar estrictamente las reglas de selección de patrones (articular beneficios específicos, demostrar aplicabilidad extendida, existencia al menos dos ejemplos reales, etc). Los patrones a veces se aplican a ciegas, sin criterios adecuados, incrementando la complejidad y los costes. Así mismo, el diseño de patrones puede ser aun más costoso y duro que la propia orientación a objetos. Si diseñar los patrones no añade ahorro de complejidad y costes, pierden su sentido.

Nótese que ninguno de los patrones de implementación cumple con el objetivo de almacenar el conocimiento de las mejores prácticas para su reutilización.

2.2.2.2 Patrones de Proceso y Mejora

a. Patrones de Proceso

Un patrón de proceso describe una aproximación probada con éxito en un conjunto de acciones de desarrollo software (Ambler, 1998) (Ambler, 1999). Ambler, en ambos trabajos, se centra en patrones de proceso basados en el paradigma de orientación a objetos y propone tres niveles de patrones; 1) Patrones de proceso de tarea, 2) Patrones de proceso de nivel y 3) Patrones de proceso de fase. La unión de los tres niveles tiene como objetivo orientar al desarrollador en diferentes niveles a la hora de llevar a cabo un desarrollo software orientado a objetos. En los trabajos de Ambler, los patrones de proceso son descritos en detalle pero el grado de complejidad para entenderlos es alto, ya que dichos patrones no siguen una plantilla específica en el que el usuario pueda obtener de una forma rápida y esquemática los datos más importantes del patrón. El apartado “solution” en el que proporciona la solución del patrón, es explicada en lenguaje natural por lo que la sistematización de dicho patrón es casi imposible de realizar.

En (Iida, 1999) se introduce el concepto de evolución de los patrones de proceso, establecido para aplicar los patrones de proceso. En (Gnatz et al, 2003) se propone un framework para describir los procesos de desarrollo software modulado en base a patrones de proceso. En dicho trabajo se describe una aplicación web llamada LiSA para ayudar a definir y mantener los procesos de desarrollo software de las organizaciones. Dichos patrones de proceso se basan en una plantilla denominada “Plantilla para Patrones de Proceso de Desarrollo Software” (Gnatz, et al, 2003) que es mostrada en la Tabla 2-6. Dicha herramienta se desarrolló en 2003 y parecía prometedora, pero a día de hoy no se puede evaluar el avance de dicha herramienta porque no se ha publicado nada al respecto ni se tiene acceso a dicha herramienta.

La comunidad de investigadores dedicada a trabajar en patrones de proceso difiere en la forma en que estos patrones deben ser documentados, incluyendo variaciones de los elementos que a continuación se van a describir en la Tabla 2-6.

Sección	Descripción
Name	Nombre del patrón de proceso de desarrollo
Also Known As	Otros nombres por los que se conozcan al patrón
Author	El nombre del autor del patrón
Intent	Un resumen de la intención y lógica o heurísticas del patrón
Problem	El elemento que dirige el patrón, incluyendo una discusión de restricciones asociadas proporcionando un escenario o ejemplo del mundo real.
Context	Situación o estado de un proyecto en el cual el patrón de proceso pueda ser aplicable.
Solution	Estado o configuración del sistema después de haber aplicado el patrón. Describe tanto los efectos positivos como perniciosos. La solución puede ser descrita usando texto o técnicas de descripción gráfico.
Consequences	Los beneficios que el patrón proporciona y potenciales handicaps o problemas
Known uses	Descripción de ocurrencias conocidas del patrón y su aplicación dentro de sistemas existentes. Ejemplos que muestren la utilidad del patrón
See Also	Referencias a patrones que resuelvan problemas similares

Tabla 2-6 Plantilla de Patrones de Proceso

En (Garson, 2006) se proporciona una guía sobre cómo llevar a cabo tareas concretas en el marco de un proceso de desarrollo a través de patrones de proceso. Por lo que, los usuarios de un patrón de proceso se verán beneficiados de la experiencia de aquellos que hayan intentado de varias formas una determinada tarea, el saber derivado de dicha experiencia se destila, generaliza y encapsula en un patrón de proceso. La categorización e identificación apropiada de patrones de proceso permite diseminar el conocimiento y permite al equipo de trabajo focalizar los esfuerzos apropiadamente.

Diversas aproximaciones han sido realizadas para la formalización de los Patrones de Proceso, en (Ribo et al, 2002) se describe el lenguaje PROMENADE que describe el Patrón de Proceso como una plantilla de proceso parametrizable y propone un conjunto de operadores para manipular los modelos de procesos. En (Hagen et al, 2004), se define un lenguaje basado en Unified Modeling Language (UML) denominado PROPEL para poder describir de una forma semiformal los Patrones de Proceso y las relaciones entre ellos. En (Tran et al, 2006) se

describe un meta-modelo que permite representar modelos de procesos y patrones de procesos en diferentes tipos de niveles de abstracción.

En estos trabajos, no se refleja la implementación de los Patrones de Proceso en herramientas software, de modo que su uso se sigue restringiendo al procedimiento manual de lectura y aprendizaje de los mismos para posteriormente aplicarlos pero sin soporte tecnológico alguno, lo que conlleva a la pérdida del conocimiento del uso de dichos patrones así como la retroalimentación de los patrones para la mejora de los mismos. El único trabajo de investigación en el que se muestra la interfaz de lo que sería la introducción de los campos de patrones de proceso es la herramienta LiSA antes mencionada. Aunque dicha herramienta describe los campos, no muestra cómo los patrones de proceso se reutilizan en un desarrollo software.

b. Patrones de Mejora de Procesos Software

Los patrones de mejora de procesos software son patrones que mejoran la calidad aplicando las buenas prácticas. (Appleton, 1997). El formato para definir estos patrones son: nombre, contexto, problema, restricciones, solución, contexto resultante, base lógica, patrones relacionados y usos conocidos.

Los patrones de mejora de procesos fueron clasificados por Appleton en dos categorías: patrones organizacionales que están enfocados a la organización de la empresa, y patrones de proceso y comunicación (Tabla 2-7).

Patrones Organizativos	Patrones de Proceso y Comunicación
Local Heroes	Process is Product (process)
Center PEG	Virtual Forum (communication)
PIT also Practices	Process follows Practice (process)
Dedicated Improvement Processors	Improvement follows Process (process)
Improvement Action Teams	Improvement follows Spiral (process)

Tabla 2-7 Clasificación de Patrones de Mejora de Procesos

Estos patrones son apropiados para una organización o departamento cuya gestión tiene un comité de mejora de procesos y han realizada parte del esfuerzo que hay que aplicar a los

programas de mejora de proceso. La principal deficiencia de estos patrones es la ausencia de herramientas que soporten su implementación.

2.2.2.3 Patrones de Gestión y Control

En esta categoría de patrones únicamente se encontraron trabajos relevantes dentro de la categoría de la gestión de la configuración del software.

En (Berczuk et al, 2003) se define un lenguaje de patrones que permiten aplicar apropiadamente el proceso de la Gestión de Configuración del Software (SCM). Los patrones identificados recogen unas heurísticas que permiten llevar a cabo las tareas de SCM. Además enumera una serie de herramientas que soportan dichos patrones. Aun así la descripción de los mismos no es formal, ni siquiera se han identificado campos que estructuren dichos patrones aunque si almacenan información sobre cómo ejecutar los patrones de SCM.

2.2.2.4 Patrones Organizativos y Colaborativos

a. Patrones Organizativos

Un patrón organizativo según (Ambler, 1999) es un patrón que describe una técnica de gestión o una potencial estructura organizativa. En (Isla-Montes et al, 2004) los patrones organizativos son utilizados para el modelado de la estructura y el comportamiento social de los actores que intervienen en un sistema dentro de una metodología de desarrollo. Proponen la modelización de tres estructuras organizativas en patrones organizativos. De la utilización de dichos patrones validaron que los patrones facilitan no sólo el diseño de los sistemas si no también la modelización de los mismos. En este estudio queda patente las ventajas de utilizar un patrón organizativo aunque dicho patrón no está definido a nivel conceptual.

Coplien define los patrones organizativos como estructuras que definen la cultura de una organización de desarrollo software (Coplien, 2004). En dicho trabajo se focaliza en el desarrollo software a través de metodologías ágiles y proporciona un catálogo de cien patrones organizativos. Una vez revisados dichos patrones se puede concluir que la descripción de los patrones se realiza en lenguaje natural especificando un determinado problema, su explicación y una posible solución. Los patrones no son fácilmente entendibles por lo que no resulta fácil

de reutilizar sin tener una herramienta software que nos facilite la sistematización de dichos patrones.

b. Patrones Colaborativos

En el área de patrones colaborativos se ha identificado un catálogo de 10 patrones (Sommerville, 2004), visibles en web, pero no estrictamente relacionados con la ingeniería del software.

En (Campbell, 2004), se propone una herramienta, CoDiagram, que permite diseñar colaborativamente. Dicha herramienta se soporta sobre Visio y proporciona algunas funcionalidades para soportar el comportamiento colaborativo.

En (Lukosch et al, 2006) utilizan los patrones colaborativos para el desarrollo de plataformas groupware. En la Tabla 2-8 se muestra la estructura del patrón colaborativo definida por Lukosch (Lukosch et al, 2004).

Sección	Descripción
Name	Nombre del patrón colaborativo.
Intent	Un resumen de la intención y lógica o heurísticas del patrón.
Context	Situación o estado de un proyecto en el cual el patrón de proceso pueda ser aplicable.
Problem	El elemento que dirige el patrón, incluyendo una discusión de restricciones asociadas proporcionando un escenario o ejemplo del mundo real.
Solution	Describe las mejoras proporcionadas después de haber aplicado el patrón.
Scenariio	Descripción concreta de una situación donde el patrón puede ser utilizado para que se vea de forma más tangible el problema.
Symptoms	Ayuda a identificar la necesidad de los patrones describiendo aspectos más abstractos del patrón.
Collaborations	Indica los principales componentes o actores que interactúan con el patrón y explica cómo se relacionan entre sí.
Rationale	Explica por qué las restricciones son resueltas por el patrón.
Danger spots	Descripción de las restricciones en contra que se puede encontrar al aplicar el patrón colaborativo.
Known uses	Descripción de ocurrencias conocidas del patrón y su aplicación dentro de sistemas existentes. Ejemplos que muestren la utilidad del patrón.
Related Patterns	Referencias a patrones que resuelvan problemas similares.

Tabla 2-8 Estructura del Patrón Colaborativo (Lukosch et al, 2004)

Los patrones colaborativos definidos en (Lukosch et al, 2006) están divididos en categorías:

- Distribución de Datos: el objetivo de los patrones colaborativos en este dominio es solucionar cómo distribuir y compartir los datos.
- Consistencia de Datos: en este dominio los patrones se centran en conservar la consistencia de los datos compartidos.
- Gestión de sesión: los patrones de esta categoría se enfocan en cómo organizar una sesión colaborativa.
- Gestión de observación del entorno: este dominio proporciona patrones en el nivel de interfaz de usuario de las aplicaciones colaborativas. A los usuarios no les gusta desarrollar actividades si no saben quienes les observa. Por lo que un ejemplo de patrón de esta categoría le mostraría el nombre de todos los usuarios quienes están en la misma sesión.

2.3 HERRAMIENTAS QUE CONTEMPLAN LAS NECESIDADES DE LA SOLUCIÓN

En este apartado se describen las características de algunas herramientas y prototipos que existen en el mercado y son comparadas con la solución propuesta en esta tesis doctoral. Fichas herramientas son utilizadas por los ingenieros software para gestionar los programas de mejora de procesos software y la ejecución de proyectos.

Las funcionalidades estudiadas en cada una de las herramientas fueron las siguientes:

- gestión de conocimiento.
- reutilización del know-how.
- gestión de proyectos.
- despliegue de modelos de procesos software.
- plataforma colaborativa.
- capacidad de búsquedas de conocimiento.

2.3.1 CodeBeamer

CodeBeamer (Intland Software, 2007) es una plataforma de desarrollo colaborativo soportada por tecnología J2EE que ofrece aplicaciones con características de gestión de ciclo de vida de una aplicación. Esta plataforma permite a los miembros del grupo, monitorizar el

progreso de un proyecto en tiempo real de forma colaborativa de una manera más eficiente. Gracias a los informes que genera la plataforma se puede mantener la gestión del proyecto como una comunidad externa que esté informada acerca del progreso del proyecto.

Las características principales son:

- Gestión de proyectos empleando áreas de trabajo: los datos de codeBeamer se organizan en proyectos. Los proyectos están en espacios de trabajo colaborativos donde los usuarios pueden compartir, discutir y encontrar información.
- Gestión de tareas, cambios, proyectos y defectos: se podrá realizar el seguimiento de tareas, bugs, requisitos, cambios, histórico de dichas tareas y proyectos.
- Gestión de documentos: permite realizar gestión de versiones y bloqueos sobre documentos, búsqueda e indexación sobre archivos word, rtf, pdf, html, wiki y texto plano. Permisos de acceso a ficheros basados en roles. Notificación de cambios en ficheros a través de E-mail. Permisos de Lectura/Escritura a nivel de directorios.
- Wiki Corporativo.
- Gestión de configuración del software.
- Tablón de anuncios.
- Foros, informes, seguridad SSL, LDAP y Active Directory, acceso vía web, permisos a través de roles y grupos.
- Despliegue de proyectos de manera rápida gracias a plantillas.
- Api para la obtención de estadísticas con distintos lenguajes de programación.
- Multiplataforma.

Esta herramienta carece de formalismo en la gestión de conocimiento; aunque el conocimiento es gestionado a través de una wiki, y éstas son consideradas sistemas de gestión de contenido (Cunningham et al, 2001), esta herramienta no tiene un artefacto formal para la gestión de conocimiento que le proporcione capacidad para reutilizar el conocimiento tácito de una forma automática en las organizaciones. Además, carece de un despliegue de modelos de procesos software que ayude al jefe de proyecto a determinar qué modelo de proceso utilizar ni proporciona una guía electrónica de proceso para poder desarrollar las actividades del proyecto.

2.3.2 IRIS Process Author

IRIS Process Author (Osellus, 2007) es un sistema de gestión de proceso visual que permite la personalización de activos de proceso colaborativamente. Esta herramienta se enfoca en la gestión de proceso aunque permite la gestión de proyectos exportando el contenido de los procesos a terceras herramientas de gestión de proyecto como Microsoft Project.

Aunque esta aplicación tiene algunas características de gestión de conocimiento, carece de poder reutilizar el conocimiento de la organización y proyectos. Para la representación del know-how usa plantillas y wikis en vez de utilizar una representación formal como patrones, ontologías o tesauros. Algunas de sus deficiencias son la planificación y seguimiento de un proyecto.

2.3.3 Microsoft Visual Studio Team System

Microsoft Visual Studio Team System (VSTS) (Microsoft Corporation, 2007). Es una solución para la gestión de aplicaciones de ciclos de vida, procesos y guías que ayudan a un equipo de desarrollo mejorar sus capacidades y hacer más efectivo su trabajo.

La principal carencia de esta herramienta es que no soporta funcionalidades de gestión de proyecto viendo el proyecto como un todo, y la falta de gestión de conocimiento, no tiene un artefacto para la representación formal del conocimiento lo que conlleva el no poder reutilizar el conocimiento.

2.3.4 Select Solution Factory

Select Solution Factory (Select Business Solutions, 2006) es una herramienta que conecta componentes de desarrollo. Permite a las organizaciones gestionar los proyectos de desarrollo software así como reutilizar componentes.

Aunque esta aplicación facilita el desarrollo e implantación de proyectos software, carece de características de gestión de conocimiento como la reutilización de conocimiento en los proyectos software.

2.3.5 Building an Organization Repository of Experiences

Building an Organization Repository of Experiences (Bore) (Henninger, 2003) es un prototipo destinado a mejorar la calidad del desarrollo software mediante el aprovechamiento de las experiencias pasadas de los desarrolladores que ha sido propuesta por el Dr. Scott Henninger de la Universidad de Nebraska-Lincoln, especializado en el ámbito de los procesos de reutilización del software y la Ingeniería del Software en general (Figura 2-5).

El objetivo de la herramienta Bore consiste en usar las soluciones que otros han encontrado para ayudar a resolver los problemas con los que se encuentra a la hora de desarrollar software y de esta manera mejorar la calidad final del producto a la vez que incrementamos la agilidad del proceso de desarrollo. Esto se consigue mediante la construcción de un repositorio de experiencias basado en casos. Este repositorio organiza jerárquicamente las actividades de un proyecto, cada caso es una instancia concreta de una actividad del proyecto que almacena el problema, la solución dada y la información documental específica del contexto de ese problema. La construcción del repositorio se realiza incrementalmente, basándose en las actividades que se han ido seleccionando para un dominio concreto. De hecho, el Dr. Henninger propone la flexibilización de los procesos de desarrollo, adaptándolos al caso concreto en base a la experiencia recopilada anteriormente. De esta manera se consigue que los modelos de ciclo de vida se adecúen más al caso concreto con el que se trata, evitando actividades superfluas o innecesarias que puedan retrasar el proyecto. Para dar soporte a esta visión, el Dr. Henninger propone un sistema basado en reglas que capture formalmente las decisiones tomadas en un proyecto, de forma que puedan servir fácilmente para ayudar en las futuras decisiones que puedan plantearse en otros proyectos. Al final de todo, lo que se consigue es una herramienta que, basándose en el dominio del problema, proponga una serie de actividades provenientes de metodologías estandarizadas, que se adapten al proyecto de la mejor forma posible.

Cuando el proyecto se crea consta de un conjunto mínimo de tareas (que se muestran en el Case Manager). La actividad “Chose initial methodology” tiene opciones asociadas con ella, que proporciona un conjunto de cuestiones estándar que ayudan a la elección. El sistema propondrá una metodología a seguir, con las actividades que más se ajusten a nuestro problema. Los miembros del equipo con los permisos apropiados pueden modificar en cualquier momento las tareas asignadas, borrándolas o asignando otras nuevas. Estas acciones desencadenan un proceso en escalada. Se le pide al usuario una explicación racional del porqué de su decisión y ésta se incorpora de manera provisional al proyecto. Una vez comprobado el beneficio de la decisión se aprueba y queda definitivamente incorporada al sistema. Para crear estas nuevas reglas el sistema provee de un interface sencillo que permite formalizar correctamente las reglas. El usuario tiene a su disposición una ventana donde puede escoger las precondiciones y las acciones que implica la regla. Las precondiciones se representan como una pareja pregunta/respuesta igual que aparece en la imagen anterior. De esta manera se consigue de forma rápida y sencilla una adaptación de las metodologías estandarizadas a los problemas concretos que se puede encontrar a la hora de afrontar un proceso de desarrollo software.

Aunque esta herramienta ofrece una amplia funcionalidad de gestión de conocimiento, su principal carencia es la falta de gestión de proyecto, no tiene una visión del proyecto como un todo (Figura 2-5).

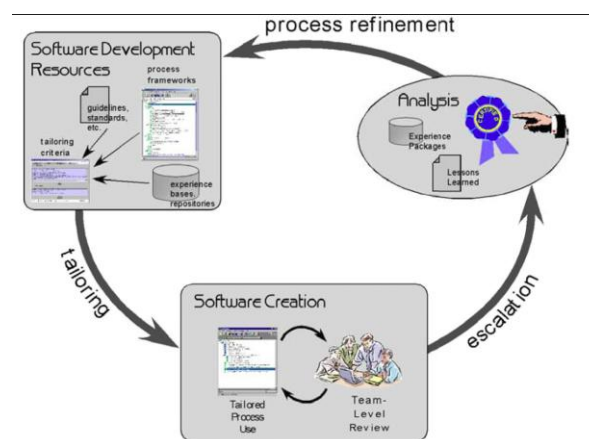


Figura 2-5 Herramienta BORE

2.3.6 Ontology Supported Software Process Knowledge Representation

La herramienta OnSSPKR (Ontology Supported Software Process Knowledge Representation) (He, 2007), es una herramienta basada en la gestión de conocimiento para soportar mejora de procesos software.

La arquitectura de esta herramienta se muestra en la Figura 2-6:

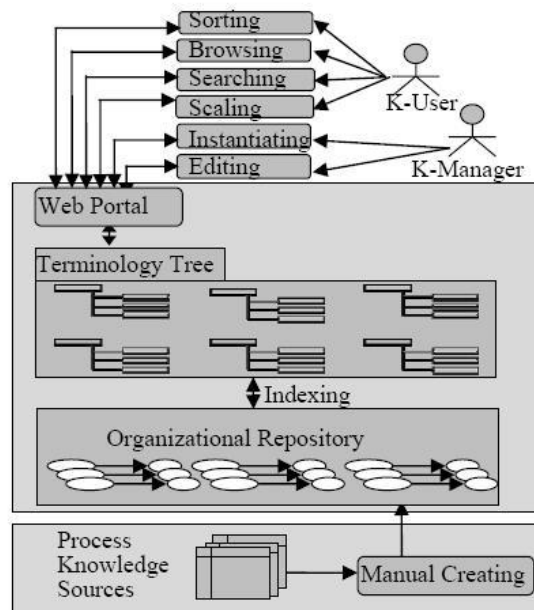


Figura 2-6 Arquitectura ONSSPKR

Sus principales funcionalidades son:

- componer y organizar activos de procesos dentro de un repositorio de conocimiento basado en ontologías.
- aportar mecanismos para poder realizar la recuperación de conocimiento de su repositorio.

Sus principales carencias son que carece de funcionalidad para la ejecución de la gestión de proyecto y que no es soportada por un framework colaborativo a pesar de tener un portal web implementado.

2.3.7 ProKnowHow

La herramienta ProKnowHow, (Silveira et al, 2002), es una herramienta basada en la gestión de conocimiento para soportar la mejora de procesos software.

Las funcionalidades más relevantes de la herramienta permiten:

- personalizar los procesos estándar para los proyectos software;
- almacenar y difundir el conocimiento adquirido durante la instanciación de los procesos estándar.
- soportar la modificación de los procesos estándar basados en la retroalimentación de los proyectos.
- seleccionar un ciclo de vida.
- estimar un proyecto.
- almacenar el conocimiento formal e informal en una base de datos de procesos software de una organización, mediante dos ontologías.
- ofrece gestión de proyecto y algunas funcionalidades para acceder a la planificación de procesos pasados y a sus lecciones aprendidas.

La arquitectura de esta herramienta se muestra en la Figura 2-7.

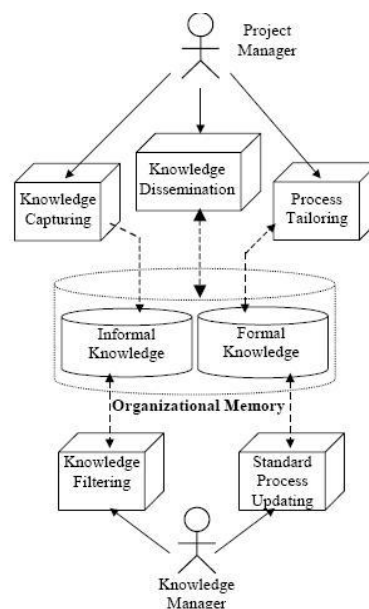


Figura 2-7 Arquitectura ProKnowHow

La herramienta ProKnowHow permite la selección y personalización de las actividades de un ciclo de vida a través de su base de datos de conocimiento formal; además proporciona las lecciones aprendidas de dichas actividades que están almacenadas en la base de datos de conocimiento informal.

Las principales carencias de dicha herramienta son:

- No almacena el Know-how de cómo se realizan cada una de las actividades del ciclo de vida. La reutilización que proporciona no está formalizada, ni es total, ya que únicamente proporciona qué actividades se realizan en cada ciclo de vida así como las lecciones aprendidas de haber realizado dichas actividades en proyectos.
- No ofrece ni guías electrónicas de procesos software ni un entorno colaborativo para la ejecución y gestión de los proyectos software. Aunque en (Silveira et al, 2002) se puede observar que la plataforma tecnológica es Web, ni en dicho artículo ni en otros relacionados con esta herramienta se han encontrado datos en los que se detallen la arquitectura tecnológica utilizada.

2.3.8 TaskNavigator

La herramienta TaskNavigator (Holz et al, 2006) soporta la reutilización del Know-how de procesos basado en tareas y la entrega de información proactiva.

Dicha herramienta está basada en la aproximación PERL (Propose, Enrich, Reuse, Learn) que está en el área de la gestión de conocimiento orientado a procesos de negocio (Business Process-Oriented Knowledge Management – BPOKM). La aproximación PERL permite almacenar, usar y reutilizar el conocimiento que los trabajadores requieren y desarrollan en sus tareas diarias. A continuación se detallan cada uno de los pasos del ciclo PERL, que se muestran en la Figura 2-8.

- Propose: el sistema le proporciona al trabajador el conocimiento que se tenga almacenado de la tarea a realizar.
- Enrich: el trabajador le dará información informal al sistema y éste le devolverá todo lo relacionado con dicha información.

- Reuse: el sistema soportará la reutilización sugiriéndole tareas similares, guías de proceso, definición de los procesos formalmente. El sistema ayudará al trabajador a crear las tareas a reutilizar en el repositorio con el know-how obtenido de la ejecución de dichas tareas.
- Learn: el sistema aprende ya que almacena los nuevos documentos y conocimiento originado en la ejecución de las nuevas tareas.

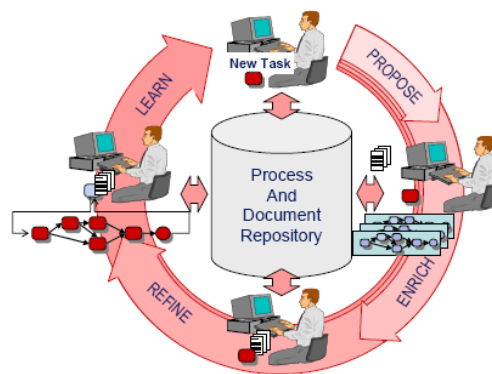


Figura 2-8 Ciclo PERL

Las funcionalidades más relevantes de la herramienta permiten realizar las siguientes operaciones:

- Gestionar las tareas de una forma ágil: los usuarios pueden crear y editar tareas, especificar las fechas, estado y prioridad de las tareas. Permite mantener una lista de documentos y urls organizadas en carpetas. Se puede definir un flujo de trabajo especificando las relaciones de precedencias de las tareas.
- Recuperación y entrega de información proactiva.
- Reutilizar el know-how de los procesos.
- Permite tener un espacio documental colaborativo a través de una wiki.

La herramienta está compuesta de la herramienta TaskNavigator que es una herramienta soportada bajo una plataforma Web que interactúa con la herramienta BrainFiler que es el módulo donde se accede al know-how de cada una de las tareas y se basa en (Figura 2-9):

- un sistema de ficheros.
- una wiki.
- Protocolo de Acceso a Mensajes Internet.

- WebDAV: que es un protocolo que permite la edición y versionado de contenido distribuido sobre la web. Este protocolo pertenece al grupo de trabajo en ingeniería de internet IETF, es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, tales como transporte, encaminamiento, seguridad. Fue creada en EEUU en 1986. Es una institución formada básicamente por técnicos en Internet e informática cuya misión es velar porque la arquitectura de la red y los protocolos técnicos que unen a millones de usuarios de todo el mundo funcionen correctamente. Es la organización que se considera con más autoridad para establecer modificaciones de los parámetros técnicos bajo los que funciona la red.

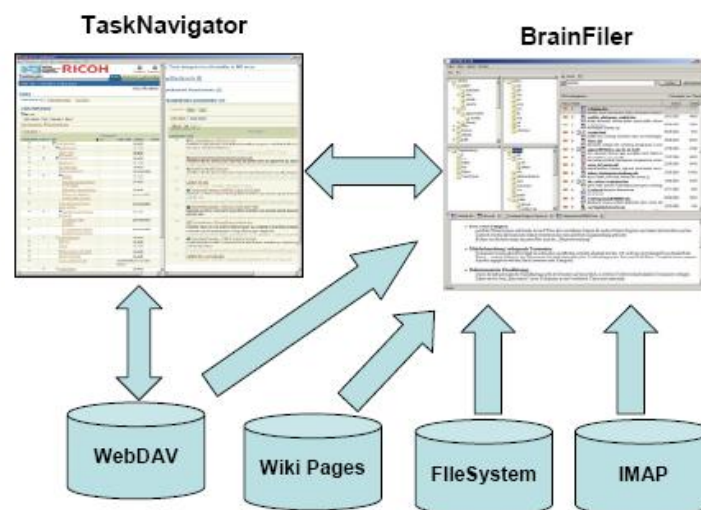


Figura 2-9 Módulos de la herramienta TaskNavigator

Las principales carencias de dicha herramienta son:

- No tiene una visión total del proyecto.
- El know-how se almacena en un sistema de ficheros por lo que la reutilización que proporciona no está formalizada.
- No ofrece ni guías electrónicas de procesos software ni un entorno colaborativo para la ejecución y gestión de los proyectos software.

2.4 RESUMEN DEL ESTADO DE LA CUESTIÓN

Los trabajos analizados en el estado de la cuestión son trabajos relacionados con las áreas de conocimiento en las que se apoya esta tesis doctoral. Dichos trabajos se pueden clasificar en tres grupos:

1.- Trabajos relacionados con el área de la gestión de conocimiento: estos trabajos se catalogan en cuatro secciones:

A. Trabajos que analizan técnicas, modelos y tecnologías para la representación del conocimiento.

B. Trabajos que estudian las arquitecturas para compartir y gestionar activos de desarrollo

C. Trabajos más cercanos a los objetivos de esta tesis doctoral.

D. Trabajos relacionados en la interacción de las áreas de conocimiento implicadas en esta tesis doctoral.

2.- Trabajos relacionados con la línea de investigación de reutilización desde el punto de vista de la Ingeniería del Software.

3.- Trabajos relacionados con las herramientas y prototipos existentes en el mercado y que son utilizadas por los ingenieros software para gestionar los programas de mejora de procesos software y la ejecución de proyectos.

A continuación se describe una revisión sintetizada de los trabajos analizados.

1.- Trabajos relacionados con el área de la gestión de conocimiento

A. Trabajos que analizan técnicas, modelos y tecnologías para la representación del conocimiento.

1 Web Semánticas

Dentro del grupo de trabajos relacionados con la representación del conocimiento mediante Web semánticas (Nakayama et al, 2005) (Falbo et al, 2004) (Brunner et al, 2007), el trabajo de Antunes (Antunes et al, 2007) es el que más se aproxima a este trabajo de

investigación ya que proporciona una forma de reutilizar y compartir el conocimiento entre los desarrolladores software utilizando el contexto de los usuarios a través de tecnología basada en Web Semánticas. El problema identificado en este trabajo es que únicamente almacena los elementos de conocimiento realizados por los desarrolladores y su contexto. Es decir, únicamente almacenan los productos finales y el contexto a través del cual se han obtenido. *La aportación de la solución propuesta en el ámbito de la representación del conocimiento es el artefacto definido en esta tesis doctoral, denominado patrón de producto. Con este artefacto, además de almacenar los productos finales y el contexto a través el cual se han obtenido dichos productos, se permite almacenar el conocimiento de las mejores prácticas de la ingeniería del software y el conocimiento tácito que los ingenieros expertos de las organizaciones tienen en el uso de la ingeniería del software, para obtener dichos activos de procesos.*

2 Repositorios de Experiencia

De todos los trabajos relacionados con repositorios de experiencia (Basili et al, 1994), esta tesis doctoral se ha centrado en los trabajos enfocados en cómo implementar dichos repositorios de experiencia (Dingsoyr, 2000) (Feldmann et al, 2003) en las organizaciones. El trabajo realizado por Feldmann (Feldmann et al, 2003) es uno de los trabajos más rigurosos encontrados de todos ellos. *Dicho estudio se centra en cómo implantar nuevos esquemas de repositorios de experiencia a través de un repositorio de experiencia y patrones de proceso. Esta solución está enfocada al proceso y no al producto, es decir, se almacena el conocimiento para la realización de un proceso a través de patrones de proceso en el repositorio de experiencia. La aportación de la solución propuesta está enfocada a los productos que se desarrollan en un proceso o proyecto de software, de esta forma la solución es más interoperable entre proyectos software. La solución que propone Feldman es incompleta ya que únicamente permite almacenar el conocimiento y la experiencia extraída de los ingenieros expertos para realizar un determinado proceso. Sin embargo, la solución propuesta en esta tesis doctoral, proporciona un marco metodológico para llevar a cabo todo el ciclo de la gestión de conocimiento, adquirir el conocimiento, almacenar y difundir el conocimiento a través de un artefacto definido en este trabajo y la transferencia de conocimiento a través de un soporte tecnológico para su posterior utilización y reutilización en el desarrollo de proyectos.*

3 Wikis en ingeniería del software

Cunningham y Chau (Cunningham et al, 2001) (Chau et al, 2005) proponen las wikis como herramienta de compartición de conocimiento en las organizaciones. El trabajo que más se aproxima a la solución propuesta es la realizada por Petter y Vaishnavi (Petter et al, 2008) que se centra en facilitar la reutilización de la experiencia obtenida en el desarrollo de proyectos software entre los jefes de proyectos a través de bibliotecas de intercambio de experiencia y una comunidad de práctica (Woo et al, 2004) especializada en gestión de proyectos, mediante tecnología wiki. *Aunque en este trabajo se hace énfasis en que la solución que aportan para la transferencia y reutilización del conocimiento está en formato textual y por tanto es un formato fácil de entender y de obtener; dicho formato no va a permitir automatizar la reutilización del conocimiento en proyectos futuros. Esta tesis doctoral propone que según las características del contexto y las restricciones de la actividad a realizar para obtener un determinado producto, se proporcione los pasos a seguir para desarrollar dicha tarea e información adicional como por ejemplo las lecciones aprendidas de haber ejecutado dichos pasos en diferentes proyectos. Esta búsqueda del conocimiento se realizará de forma sistemática porque dichos pasos se encontrarán en el artefacto definido en esta tesis doctoral. A pesar del inconveniente encontrado en las wikis, solucionado con el uso del artefacto definido en esta tesis doctoral, se proporcionará una wiki a los componentes del equipo de desarrollo donde esté contenida toda la información referente al proyecto y sus activos de proceso.*

4 Librería de Activos de Procesos

Las librerías de activos de proceso (PAL) (Garcia, 2006) proporcionan: un repositorio de conocimiento central para adquirir, definir y difundir las guías sobre procesos relacionados para las actividades de la organización, reducción de duplicados innecesarios de activos de procesos de la organización; mecanismos para compartir el conocimiento sobre los activos de proceso y cómo éstos son usados. *Las funcionalidades que aporta una PAL a una organización, están incluidas en las características y objetivos que se ha definido en esta tesis doctoral. La utilización de la PAL está combinada con la utilización del artefacto de conocimiento para que la solución sea reutilizable.*

5 Guías Electrónicas de Proceso (EPG)

Las guías electrónicas de proceso (Kellner et al, 1998) han sido aplicadas en diferentes trabajos de investigación, algunos de ellos son (Niazi et al, 2006) (Brede et al, 2006) (Bjørnson et al, 2005) (Moe et al, 2006) (Phongpaibul et al, 2007). De los resultados obtenidos en (Moe et al, 2006), se llegó a la conclusión que las EPG hay que adaptarlas a los procesos de la organización y que una EPG está enfocada a los procesos y no a los productos. *En esta tesis doctoral se considera útil la utilización de una guía electrónica de proceso combinada con los patrones de producto para proporcionar al equipo de desarrollo una guía de proceso con la información contenida en dichos patrones de producto, para que cada uno de los roles del equipo tenga en todo momento los pasos que tiene que seguir para obtener el producto deseado.*

B. Trabajos más cercanos a los objetivos de esta tesis doctoral.

Los trabajos de (Tseng et al, 2008) y de (Gnatz et al, 2003) son los que se encuentran más cercanos a la aproximación de esta tesis doctoral.

En el trabajo de (Tseng et al, 2008) se desarrolló un gestor de conocimiento basados en actividades (ABKM) para construir proyectos software. *Al igual que en este trabajo, en la solución propuesta se considera que la Ingeniería del Software se tiene que apoyar en el área de gestión de conocimiento para la difusión, uso y reutilización de activos de procesos en la realización de un proyecto software. En dicho trabajo (Tseng et al, 2008), se considera la actividad como la unidad para la construcción de proyectos software. En la presente tesis doctoral es fundamental tener el conocimiento para poder realizar una actividad, pero realmente en la ejecución de un proyecto y tras realizar una actividad o un conjunto de actividades, se obtienen productos. Es por ello, que se ha identificado al producto como unidad mínima que aporta interoperabilidad al sistema. A continuación se detallan las aportaciones de la tesis doctoral, a las fases del ciclo de la gestión de conocimiento enunciadas en (Tseng et al, 2008):*

- *Fase de adquisición y extracción del conocimiento: se realiza la adquisición y extracción de los ingenieros expertos del conocimiento de las mejores prácticas de la ingeniería del software.*

- *Fase de almacenamiento del conocimiento: el conocimiento es almacenado en artefactos denominados patrones de producto que proporciona a los proyectos de la capacidad de reutilización e interoperabilidad.*
- *Fase de compartición del conocimiento: al utilizar los patrones de producto, directamente se le proporciona al ingeniero la solución al problema a resolver de una forma gráfica donde se le detallan todos los pasos que tiene que realizar para obtener el producto deseado en el proyecto.*
- *Fase de modificación del conocimiento: la retroalimentación se hace de forma sistemática en la solución propuesta.*

En el trabajo de (Gnatz et al, 2003) se presenta un framework que proporciona la capacidad de incorporar el conocimiento de los procesos de la organización mediante el concepto de patrón de proceso que almacena el conocimiento de las metodologías y modelos de proceso para que puedan ser utilizadas en el desarrollo de proyectos software. *En este trabajo de investigación, se proporciona al equipo de desarrollo, los procesos de desarrollo así como los patrones de proceso que les indican las actividades que tienen que realizar para realizar dicho proceso. Esta solución (Gnatz et al, 2003) está enfocada a procesos y no a productos, es decir, los artefactos de proceso proporcionan al usuario las actividades que tiene que realizar para realizar un proceso de desarrollo, esta solución minimiza la interoperabilidad entre proyectos software.*

C. Trabajos relacionados en la interacción de las áreas de conocimiento implicadas en esta tesis doctoral.

Gestión de Conocimiento y Mejora de Procesos Software.

Son muchos los trabajos que relacionan las áreas de gestión de conocimiento y mejora de procesos software (Butler, 1995) (Pitterman, 2000) (Yamamura, 1999), (Niazi et al, 2006) (Burke et al, 2005) (Stelzer et al., 1998) (Baskerville, 1998) (Arent et al, 2000) (Komi-Sirviö et al, 2002). De estos trabajos se puede concluir que en las organizaciones: existen problemas para difundir el conocimiento entre los diferentes niveles organizacionales; el conocimiento no es fácil de localizar y no se reutiliza; la captura del conocimiento es demasiado informal y no es incorporado en los procesos de ingeniería. En (Assimakopoulos et al, 2006) identifica la necesidad de un artefacto para almacenar el conocimiento tácito y que permita la reutilización

del conocimiento pero no propone ningún artefacto para tal fin. *Los trabajos antes mencionados no proporcionan los procesos ni mecanismos para que las organizaciones puedan adquirir, difundir, usar y reutilizar el conocimiento de las mejores prácticas de la ingeniería del software y el conocimiento de los procesos de su propia organización para mejorar sus procesos. La solución propuesta define un artefacto y sus mecanismos de colaboración para el almacenamiento, difusión y reutilización del conocimiento en las organizaciones de una forma sistemática y eficiente.*

Gestión de Conocimiento y Entornos Colaborativos.

De los trabajos analizados en el contexto de entornos colaborativos, (Vigotsky, 1987) (Greif, 1988) (Marca, 1992) (Crowe, 1994) (Coleman, 1997). (Dave et al, 2000) (Carstensen et al, 2002) (Rus, 2002). (Garner et al, 2003). (D'Souza et al, 2003) (Pratt et al, 2004) (Dewiyanti. et al, 2004), (Lau et al, 2003) se extraen las ventajas competitivas por lo que se debe utilizar un entorno colaborativo en las organizaciones para la difusión y transferencia de conocimiento *Del análisis de los trabajos, se extrae la conclusión que la solución propuesta en esta tesis doctoral tiene que soportarse mediante tecnología web colaborativa para que el conocimiento adquirido, generado en el uso de los procesos software pueda ser transferido y reutilizado en los proyectos software.*

2.- Trabajos relacionados con reutilización de software

La gestión de conocimiento requiere mecanismos para acceder a los componentes y reutilizar dicho conocimiento. En el área de la Ingeniería del Software existe una línea de investigación que es la reutilización. Algunos trabajos en esta línea son: la reutilización de componentes (Ramachandran, 2006), reutilización de procesos, reutilización de software (Succi et al, 2001), familias de productos software (Deelstra et al, 2004). Hay muchos trabajos de investigación relacionados con estas líneas pero no son objetivo de esta tesis doctoral. En esta tesis doctoral, la reutilización está enfocada en la identificación de los artefactos y mecanismos que sirvan para la reutilización del conocimiento de los ingenieros expertos y la reutilización de activos de proceso en proyectos software. Por ello, se necesita un artefacto que permita dar una solución práctica a un problema bajo un contexto y unas restricciones, y que tenga la característica principal de la reutilización, como son los patrones software.

El concepto de patrón ha sido definido por numerosos autores, pero todos ellos parten de la definición original dada por Alexander en 1979 (Alexander, 1979). En el campo del

software, los patrones representan soluciones a problemas que surgen en el desarrollo de aplicaciones, permitiendo su reutilización como respuesta a conflictos similares surgidos en contextos diferentes. La clave está en que el patrón es una generalización o abstracción reusable que puede utilizarse como punto de partida para soluciones futuras. *El hecho de tener en consideración este aspecto es importante ya que lo que se pretende con esta tesis doctoral es facilitar la difusión, uso y reutilización de activos de procesos. Por ello, la utilización de patrones como el artefacto que almacene el conocimiento de las mejores prácticas de los ingenieros expertos y del conocimiento tácito de las organizaciones para su posterior reutilización, se adecua a las necesidades de esta tesis doctoral.*

Actualmente, los patrones están siendo utilizados en muchas soluciones aplicadas al campo de la Ingeniería del Software (Coplien, 2004). A continuación se muestran los trabajos relacionados con los diferentes tipos de patrones clasificados en el marco de la disciplina de la Ingeniería del Software (patrones de implementación, patrones de proceso y mejora y, los patrones de gestión y control) y los que salen de los márgenes de la misma y que pudieran estar relacionados con la solución propuesta (patrones colaborativos y organizativos)

A continuación se van a detallar los trabajos más relevantes de cada tipo de patrones identificados (ver Tabla 2-9).

Categoría	Tipo de Patrón	Referencias
Patrones de Implementación	Arquitectura de Referencia	(Buschmann et al, 1996)
	Patrones de Arquitectura	(Buschmann et al, 1996) (Shaw et al, 1996)
	Patrones de Análisis	(Fowler, 1997); (Purao et al, 2003) (Kodaganallur et al, 2006)
	Patrones de Diseño	(Gamma, 1994)
Patrones de Proceso y Mejora	Patrones de Proceso	(Ambler, 1998) ; (Ambler, 1999) (Ribo et al, 2002); (Hagen et al, 2004) (Garson, 2006); (Tran et al, 2006)
	Patrones de Mejora	(Appleton, 1997)
Patrones de Gestión y Control	Patrón de Gestión de la Configuración	(Berczuk, 2003)
Patrones Colaborativos y Organizativos	Patrones de Colaboración	(Isla et al, 2004); (Coplien, 2004)
	Patrones Organizativos	(Sommerville, 2004); (Campbell, 2004) ; (Lukosch et al, 2006)

Tabla 2-9 Tipos de Patrones

Ninguno de los patrones estudiados excepto el patrón de proceso, está enfocado a aportar el conocimiento necesario para el desarrollo de proyectos software, por lo que todos los patrones identificados excepto el patrón de proceso no son aplicables a esta solución.

Los patrones de proceso proporcionan una solución para resolver un problema como es la realización de un determinado proceso. Esta solución es muy restrictiva en el ámbito del desarrollo de proyectos software porque únicamente dichos patrones van a proporcionar el conocimiento (un conjunto de actividades) para realizar un determinado proceso, como por ejemplo el proceso de análisis. Sin embargo, los proyectos pueden utilizar diferentes modelos de proceso de los cuales se obtienen los mismos productos, por ejemplo un diagrama de casos de uso. Los procesos utilizados en ambos modelos de procesos son diferentes pero el producto a realizar es el mismo, sin embargo, el mismo patrón de proceso no se podría utilizar por ser procesos diferentes. Es por ello que los patrones de procesos son restrictivos y poco interoperables entre proyectos software. Los patrones de producto proporcionan máxima interoperabilidad entre proyectos software ya que dichos patrones proporcionan el conocimiento para obtener un producto software.

3.- Trabajos relacionados con las herramientas y prototipos existentes en el mercado laboral para gestionar los programas de mejora de procesos software y la ejecución de proyectos.

CodeBeamer

CodeBeamer (Intland Software, 2007) es una plataforma de desarrollo colaborativo que ofrece aplicaciones con características de gestión de ciclo de vida de una aplicación. Las características principales son: Gestión de proyectos, Gestión de tareas, Gestión de documentos, Wiki Corporativo, Gestión de configuración del software, Plataforma colaborativa, Capacidad de búsquedas de conocimiento. *Esta herramienta carece de formalismo en la gestión de conocimiento; aunque el conocimiento es gestionado a través de una wiki, y éstas son consideradas sistemas de gestión de contenido (Cunningham et al, 2001), esta herramienta no tiene un artefacto formal para la gestión de conocimiento y es por ello que no tiene capacidad de reutilizar el conocimiento tácito de una forma automática en las organizaciones. Además, esta herramienta carece de un despliegue de modelos de procesos software que ayude al jefe de proyecto a determinar qué modelo de proceso utilizar,*

ni proporciona una guía electrónica de proceso para poder desarrollar las actividades del proyecto.

IRIS Process Author

IRIS Process Author (Osellus, 2007) es un sistema de gestión de proceso visual que permite la personalización de activos de proceso colaborativamente. Esta herramienta se enfoca en la gestión de proceso aunque permite la gestión de proyectos exportando el contenido de los procesos a terceras herramientas de gestión de proyecto. *Aunque esta aplicación tiene algunas características de gestión de conocimiento, no permite reutilizar el conocimiento de la organización y proyectos. Para la representación del know-how usa plantillas y wikis en vez de utilizar una representación formal. Algunas de sus deficiencias son la planificación y seguimiento de un proyecto.*

Microsoft Visual Studio Team System

Microsoft Visual Studio Team System (VSTS) (Microsoft Corporation. 2007). Es una solución para la gestión de aplicaciones de ciclos de vida, procesos y guías que ayudan a un equipo de desarrollo mejorar sus capacidades y hacer más efectivo su trabajo. *Su principal carencia es que no proporciona un artefacto para la representación formal del conocimiento, por lo que el conocimiento tácito no se almacena ni se reutiliza; además no soporta funcionalidades de gestión de proyecto viendo el proyecto como un todo.*

Select Solution Factory

Select Solution Factory (Select Business Solutions, 2006) es una herramienta que conecta componentes de desarrollo. Permite a las organizaciones gestionar los proyectos de desarrollo software así como reutilizar componentes. *Aunque esta aplicación facilita el desarrollo e implantación de proyectos software, carece de características de gestión de conocimiento como la reutilización de conocimiento en los proyectos software.*

Building an Organization Repository of Experiences (Bore)

Bore (Henninger, 2003) es un prototipo destinado a mejorar la calidad del desarrollo software mediante el aprovechamiento de las experiencias pasadas de los desarrolladores

mediante la construcción de un repositorio de experiencias basado en casos. Para dar soporte a esta visión, el Dr. Henninger propone un sistema formal basado en reglas. Proporciona una herramienta que en base al dominio del problema proponga una serie de actividades provenientes de metodologías estandarizadas, que se adaptan a nuestro proyecto de la mejor forma posible. *Aunque esta herramienta ofrece una amplia funcionalidad de gestión de conocimiento, su principal carencia es la falta de gestión de proyecto. No tiene una visión del proyecto como un todo.*

OnSSPKR

La herramienta OnSSPKR (Ontology Supported Software Process Knowledge Representation)(He, 2007) es una herramienta basada en la gestión de conocimiento para soportar mejora de procesos software. Las principales funcionalidades son: componer y organizar activos de procesos dentro de un repositorio de conocimiento basado en ontologías y aportar mecanismos para poder realizar la recuperación de conocimiento de su repositorio. *Sus principales carencias: son que carece de funcionalidad para la ejecución de la gestión de proyecto y que no es soportada por un framework colaborativo a pesar de tener un portal web implementado.*

ProKnowHow

La herramienta ProKnowHow, (Silveira et al, 2002), es una herramienta basada en la gestión de conocimiento para soportar la mejora de procesos software. Esta herramienta permite: personalizar procesos estándar; almacenar y difundir el conocimiento; modificar la modificación de los procesos estándar basados en la retroalimentación de los proyectos; seleccionar un ciclo de vida; estimar un proyecto; almacenar el conocimiento formal e informal; ofrece gestión de proyecto y algunas funcionalidades para acceder a la planificación de procesos pasados y a sus lecciones aprendidas. *Esta herramienta no almacena el Know-how de cómo se realizan cada una de las actividades del ciclo de vida. La reutilización que proporciona no está formalizada, ni es total. No ofrece ni guías electrónicas de procesos software ni un entorno colaborativo para la ejecución y gestión de los proyectos software. No soporta mecanismos de colaboración.*

TaskNavigator

La herramienta TaskNavigator (Holz et al, 2006) soporta la reutilización del Know-how de procesos basado en tareas y la entrega de información proactiva. Las funcionalidades más relevantes de la herramienta permiten realizar las siguientes operaciones: gestionar las tareas de una forma ágil; reutilizar el know-how de los procesos; permitir un espacio documental colaborativo a través de una wiki. *Esta herramienta no proporciona una visión total del proyecto. El Know-how se almacena en un sistema de ficheros por lo que la reutilización que proporciona no está formalizada y no ofrece ni guías electrónicas de procesos software ni un entorno colaborativo para la ejecución y gestión de los proyectos software.*

3. SOLUCIÓN PROPUESTA

3 SOLUCIÓN PROPUESTA	81
3.1 Elemento clave y su mecanismo de creación y retroalimentación	86
3.2 Lenguaje de patrones de producto	94
3.3 Estrategia corporativa de despliegue de la solución	103
3.4 Componentes y adaptación de la estrategia corporativa a la solución tecnológica....	118

La solución propuesta en esta tesis doctoral es una solución práctica para mejorar la eficiencia de uso de los procesos software con el fin último de facilitar la difusión, uso y reutilización de activos de procesos y proyectos, basados en la gestión del conocimiento y en mecanismos de colaboración.

En este trabajo se define un marco metodológico para la difusión, uso y reutilización de activos de procesos, mediante un artefacto definido en esta tesis doctoral, denominado *Patrón de Producto*, utilizando como soporte tecnológico una plataforma web colaborativa. Dicho marco metodológico describe:

1. El elemento clave para la reutilización de activos de proceso, *Patrón de Producto*, así como el mecanismo de creación y retroalimentación de *patrones de producto* (sección 3.1).
2. Un lenguaje de *Patrones de Producto* donde se van a describir los siguientes ítems: el catálogo de *patrones de producto* desarrollado en esta tesis doctoral; la regla de búsqueda de *patrones de producto*; el metamodelo que describe la relación entre los *patrones de producto* y la lógica de la ejecución de un proyecto software; el mecanismo de colaboración entre *patrones de producto* (sección 3.2).
3. La estrategia corporativa de despliegue de la solución que describe los procesos para realizar la difusión, uso y reutilización de los activos de proceso en una organización (sección 3.3).
4. Una solución tecnológica basada en una plataforma colaborativa que da soporte a la solución planteada en esta tesis doctoral. Así mismo, se describe cómo se adapta la estrategia corporativa a la solución tecnológica planteada (sección 3.4).

A continuación se dedica una sección a la descripción de cada uno de estos elementos.

3.1 ELEMENTO CLAVE Y SU MECANISMO DE CREACIÓN Y RETROALIMENTACIÓN

3.1.1 Elemento Clave: Patrón de Producto

En 1964, el arquitecto Christopher Alexander, creó el concepto de patrón, definiéndolo en el año 1979 como:

Un patrón describe un problema que ocurre una y otra vez en nuestro entorno, describiendo el núcleo de la solución a dicho problema de tal forma que se puede usar esta solución un millón de veces sin hacerlo dos veces de la misma forma. (Alexander, 1979)

Esta idea fue utilizada por los autores Ward Cunningham y Kent Beck en el campo de la Ingeniería del Software. Como se ha visto en el capítulo 2, los patrones software representan soluciones a problemas que surgen en el desarrollo de aplicaciones, permitiendo su reutilización como respuesta a conflictos similares surgidos en contextos diferentes. El patrón proporciona una solución reusable que puede utilizarse como punto de partida en soluciones futuras. Esta característica es fundamental para poder facilitar la difusión, uso y reutilización de activos de procesos, es por ello, que se ha elegido la utilización de patrones como el artefacto que almacene el conocimiento de las mejores prácticas de los ingenieros expertos y del conocimiento tácito de las organizaciones para su posterior reutilización.

En el área de la Ingeniería del Software existen muchos tipos de patrones (ver capítulo 2) pero ninguno proporciona una solución que se adecue a las necesidades de la solución de esta tesis doctoral. El único patrón que se asemeja es el patrón de proceso, pero la solución que proporcionan los patrones de proceso es más restringida y menos portable, ya que proporciona los pasos a realizar para llevar a cabo dicho proceso y no está enfocado en proporcionar una solución que proporcione los pasos para la obtención de un producto software en el desarrollo de un proyecto o proceso que es el objetivo a perseguir en esta tesis doctoral. Por ello, este trabajo propone un tipo de patrón, denominado *Patrón de Producto*.

El *Patrón de Producto* permite almacenar el conocimiento de las mejores prácticas de la Ingeniería del Software y la experiencia de los ingenieros expertos para obtener un producto

software específico. Se considera un producto como el elemento mínimo en la Ingeniería del Software que se obtiene de la ejecución de uno o varios modelos de proceso.

Los patrones son descritos formalmente en una plantilla en el que se describen los atributos que definen al patrón y que proporciona su aprendizaje, comparación y uso. Hay tres formatos básicos, el formato de Alexander (Alexander, 1979), el formato GOF (Gamma et al, 1994) y el formato canónico (Buschmann et al, 1996). El formato del *Patrón de Producto*, está basado en el formato canónico, modelo definido por Buschmann (Buschmann et al, 1996) para la descripción de patrones. Dicho formato es descrito en la Tabla 3-1.

Sección	Descripción
Name	Nombre significativo. Puede ser una única palabra o una frase corta que haga referencia al patrón.
Problem	Descripción del problema indicando su propósito.
Context	Precondiciones bajo las cuales se puede aplicar el patrón
Forces	Descripción de las restricciones relevantes y cómo interaccionan o entran en conflicto unas con las otras. Definición de los objetivos a conseguir.
Solution	Relaciones estáticas y reglas dinámicas que describen cómo conseguir el objetivo marcado.
Examples	Uno o más ejemplos de la aplicación del patrón.
Resulting Context	Estado o configuración del sistema después de haber aplicado el patrón. Describe tanto los efectos positivos como perniciosos.
Rationale	Explicación justificada de los pasos o reglas del patrón.
Related Pattern	Relaciones estáticas o dinámicas entre el patrón que se describe y otros.
Known uses	Descripción de ocurrencias conocidas del patrón y su aplicación dentro de sistemas existentes.

Tabla 3-1 Formato Canónico (Buschmann et al, 1996)

El *Patrón de Producto* requiere de la adaptación de los campos del formato canónico para ajustarse a la funcionalidad para la que ha sido creado, que es almacenar el conocimiento de las mejores prácticas en ingeniería del software. A continuación se muestra en detalle en la Tabla 3-2 cada uno de los campos del *Patrón de Producto* indicando por cada uno de ellos, su nombre, descripción y la aportación de dicho campo.

Nombre del Campo	Descripción	Aportación
<i>Name</i>	Nombre del <i>Patrón de Producto</i> .	Nombre significativo. Puede ser una única palabra o una frase corta que haga referencia al <i>Patrón de Producto</i> .
<i>Related Patterns</i>	Nombre de los <i>Patrones de Producto</i> relacionados con el que se está describiendo.	La relaciones entre los <i>Patrones de Producto</i> relacionados pueden ser: <ul style="list-style-type: none"> • Tipo de Patrón (Por taxonomía): que pertenezcan al mismo tipo de Patrón de Producto, por ejemplo, que sean dos patrones que pertenezcan a la fase de análisis en un desarrollo software. • Precedencia: que tengan algún tipo de relación por precedencia. Un Patrón de Producto precederá a otro cuando la salida de uno sea la entrada del otro patrón.
<i>Initial Context</i>	Contexto Inicial del <i>Patrón de Producto</i>	Precondiciones bajo las cuales se aplica el <i>patrón de producto</i> . El contexto inicial es el estado del sistema en el que el <i>patrón de producto</i> se puede aplicar
<i>Result Context</i>	Contexto resultante de haber ejecutado el <i>Patrón de Producto</i>	Contexto o estado en el que se queda el sistema después de haber aplicado el <i>patrón de producto</i> .
<i>Problem</i>	Descripción del problema.	Descripción del problema que se resuelve al aplicar el <i>Patrón de Producto</i> .
<i>Forces</i>	Restricciones bajo las cuales se ejecuta el <i>Patrón de Producto</i> .	Las restricciones pueden ser provenientes de: <ul style="list-style-type: none"> • Características de las organizaciones: Tamaño de la organización, Infraestructura de las comunicaciones, Número de empleados, Madurez empresarial, Capas de gestión organizativa, Facturación, Distribuido geográficamente. • Características de los proyectos: Experiencia del equipo de desarrollo en el lenguaje y herramientas; Experiencia en el tipo de aplicación a desarrollar; Experiencia en la plataforma; Volatilidad de la plataforma; Restricción en el tiempo de ejecución; Reutilización de requisitos; Continuidad del personal; Paradigma.
<i>Solution</i>	Relaciones estáticas y reglas dinámicas que describen cómo conseguir el producto.	Este campo describe el proceso para poder obtener el producto deseado. La descripción del proceso se realiza de forma gráfica a través de un diagrama de actividad. De esa forma, el usuario tendrá especificado de forma sintetizada y clara, cada uno de los pasos que tiene que realizar para obtener el producto. Además, se proporciona el tiempo estimado que se tarda en realizar dicho proceso.
<i>Roles</i>	Roles implicado en el desarrollo del producto.	En este campo se detallan los roles que participan en la solución del <i>Patrón de Producto</i> para obtener el producto software.
<i>Entries</i>	Productos que son requeridos para poder aplicar el <i>Patrón de Producto</i> .	En este campo se detallan los productos software que se necesitan para poder desarrollar el proceso descrito en el campo "Solution" del <i>Patrón de Producto</i> . Sin estos productos software el <i>Patrón de Producto</i> no se puede aplicar.
<i>Lessons Learned</i>	Lecciones aprendidas	Campo que detalla las lecciones aprendidas de haber utilizado el <i>Patrón de Producto</i> . Este campo es modificado tras la realización del <i>Patrón de Producto</i> .
<i>Templates</i>	Plantillas	En este campo se detalla una plantilla que permitirá realizar el proceso software de una forma más usable.
<i>Examples</i>	Ejemplos	En este campo se proporcionan ejemplos de haber aplicado el <i>Patrón de Producto</i> en otros desarrollos software.
<i>Exit</i>	Producto que se obtiene de la realización del <i>patrón de producto</i>	Producto que se obtiene de la realización de los pasos descritos en el campo "Solution".
<i>Capability Level</i>	Nivel de Madurez	En este campo se detalla el nivel de madurez que se obtiene una vez realizado el <i>Patrón de Producto</i> . Los niveles de madurez están basados en dos modelos de procesos, CMMI y SPICE.
<i>Informations Resources</i>	Referencias bibliográficas	En este campo se detallan las referencias bibliográficas que puede utilizar el usuario a la hora de desarrollar los pasos descritos en el campo Solution.
<i>Basic Knowledge</i>	Conocimiento Básico	En este campo se proporciona el conocimiento básico que tiene que tener el usuario que quiera utilizar el <i>Patrón de Producto</i> . Este campo fue añadido por petición de los usuarios que validaron los <i>Patrones de Producto</i> .

Tabla 3-2 Descripción de los campos del *Patrón de Producto*

En la Tabla 3-3 se muestra un ejemplo de un *Patrón de Producto* del catálogo de *Patrones de Producto*.

Sección	Descripción								
Name	Patrón de Producto Casos de Uso en formato alto nivel								
Related Patterns	Patrón de Producto Diagrama de Casos de uso Patrón de Producto Casos de Uso en formato expandido								
Initial Context	Se requiere la especificación de la funcionalidad del sistema.								
Result Context	Se obtendrá los casos de uso en formato alto nivel para representar cómo se interactúa con el sistema, cómo se va a usar el sistema y obtener una descripción textual de la dinámica del sistema.								
Problem	Representar la interacción con el sistema y la funcionalidad del mismo.								
Forces	<ul style="list-style-type: none"> • Organización: Pymes. • Sistema: Cualquier tipo. • Paradigma: Orientado a objetos. 								
Solution	<ul style="list-style-type: none"> • Tiempo de Realización: <ul style="list-style-type: none"> ◦ T.Realización: 8 horas • Diagrama de Actividades: 								
Roles	Jefe de Proyecto, Analista y Cliente								
Entries	Diagrama de Casos de uso Tipo de Información (.doc.) Se aplica gestión de la configuración del software? sí								
Lessons Learned	En el campo "descripción" detallar la interacción entre los actores identificados y el sistema.								
Templates	<table border="1"> <tr> <td>Nombre del Caso de uso</td> <td></td> </tr> <tr> <td>Actores</td> <td></td> </tr> <tr> <td>Tipo</td> <td></td> </tr> <tr> <td>Descripción</td> <td></td> </tr> </table>	Nombre del Caso de uso		Actores		Tipo		Descripción	
Nombre del Caso de uso									
Actores									
Tipo									
Descripción									
Examples	<table border="1"> <tr> <td>Nombre del Caso de uso</td> <td><i>Añadir Consumición</i></td> </tr> <tr> <td>Actores</td> <td><i>Camarero</i></td> </tr> <tr> <td>Tipo</td> <td><i>Primario</i></td> </tr> <tr> <td>Descripción</td> <td><i>El camarero escribe en su dispositivo el número de la consumición que ha pedido el cliente. Después pulsa el botón aceptar para confirmarlo.</i></td> </tr> </table>	Nombre del Caso de uso	<i>Añadir Consumición</i>	Actores	<i>Camarero</i>	Tipo	<i>Primario</i>	Descripción	<i>El camarero escribe en su dispositivo el número de la consumición que ha pedido el cliente. Después pulsa el botón aceptar para confirmarlo.</i>
Nombre del Caso de uso	<i>Añadir Consumición</i>								
Actores	<i>Camarero</i>								
Tipo	<i>Primario</i>								
Descripción	<i>El camarero escribe en su dispositivo el número de la consumición que ha pedido el cliente. Después pulsa el botón aceptar para confirmarlo.</i>								
Exit	Nombre: Casos de Uso en Formato Alto Nivel Tipo: Documento doc Se aplica gestión de la configuración del software? sí								
Capability Level	Nivel 2								
Informations Resources	- ANALISIS Y DISEÑO ESTRUCTURADO Y ORIENTADO A OBJETOS DE SISTEMAS INFORMÁTICOS. Antonio de Amescua Seco, Juan José Cuadrado Gallego, Emilio Errica Lafuente, Javier García Guzmán, Luis García Sánchez, Paloma Martínez Fernández, M ^a Isabel Sánchez Segura Ed. McGraw Hill. - INGENIERÍA DEL SOFTWARE. Un enfoque práctico. Roger S. Presuman Ed. McGraw Hill - http://www.cikear.com/manuales/uml/								
Basic Knowledge	Actor: toda entidad externa al sistema que guarda una relación con este y que le demanda una funcionalidad. Caso de uso: tarea de negocio.								

Tabla 3-3 Ejemplo Patrón de Producto

3.1.2 Mecanismo de Retroalimentación y Creación de Patrones de Producto

Cuando los *patrones de producto* son instanciados y utilizados en la realización de una actividad en un proyecto software, el conocimiento y la experiencia adquirida de la ejecución de dicha actividad enriquece y retroalimenta al conocimiento tácito del *patrón de producto*. De esta forma se contemplan los cuatro estadios del conocimiento (jerarquía DIKW (Clark, 2004)) alcanzando el último estadio, innovación, transformando el conocimiento adquirido a través de la experiencia en nuevo y mejorado conocimiento que retroalimenta o crea nuevos *patrones de producto* que se aplicarán en futuros proyectos software.

La retroalimentación o creación de un nuevo *patrón de producto* depende de si el conocimiento y la experiencia adquirida añade nuevas heurísticas de uso del *patrón de producto*. Si no se añade nuevas heurísticas de uso, el conocimiento retroalimentará algunos campos del *patrón de producto* como son: “*Lessons Learned*”, “*Templates*” y “*Examples*”.

Si el conocimiento y la experiencia adquirida añaden una nueva heurística de uso al *patrón de producto*, se creará un nuevo *patrón de producto* similar al de origen pero con determinadas particularidades. Esto ocurre en las siguientes situaciones:

- cuando aparece una restricción nueva (“Force”)
- se enriquece o cambia el contexto (“Initial o Result Context”)
- se enriquece el problema (“Problem”)

En la Figura 3-1 se esquematizan ambos mecanismos. Si al patrón P1 se le añade una nueva heurística de uso, se creará un nuevo *patrón de producto* P1’. En caso contrario, simplemente se retroalimenta el *patrón de producto* P1 con el conocimiento obtenido del uso del patrón.

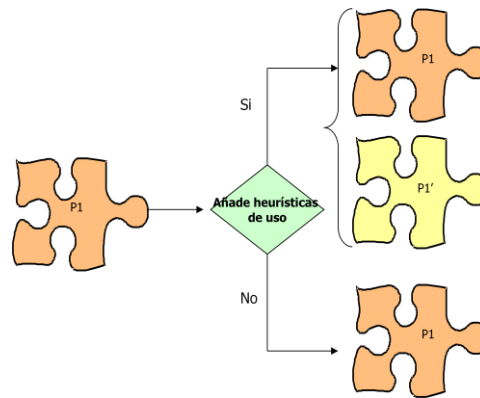


Figura 3-1 Retroalimentación y creación de un *Patrón de Producto*

A continuación se muestra un ejemplo de los mecanismos de retroalimentación y creación de un nuevo *patrón de producto*. El *patrón de producto* del que se parte es el “Patrón de Producto Casos de Uso en Formato de Alto Nivel”, ilustrado en la Tabla 3-3.

- *Mecanismo de Retroalimentación:* Una vez que el *patrón de producto* se ha instanciado y utilizado, el conocimiento de dicho *patrón de producto* se retroalimenta con el conocimiento adquirido de la experiencia de uso del patrón. En el ejemplo expuesto, se retroalimenta el campo “Example” con la instanciación de dicho *patrón de producto* en un proyecto software específico, ver Tabla 3-4.
- *Mecanismo de Creación de un nuevo Patrón de Producto:* Una vez aplicado el “Patrón de Producto Casos de Uso Alto Nivel” en un proyecto de “business intelligence”, se observó que era necesario una nueva heurística de uso que modificaba el *patrón de producto* original y creando un nuevo *patrón de producto*, “Patrón de Producto Casos de Uso Alto Nivel para Bussiness Intelligenge”. Dicha heurística de uso añadió una nueva restricción al *patrón de producto* en el campo “Forces”, y una nueva tarea a realizar en el campo “Solution”, ver Tabla 3-5.

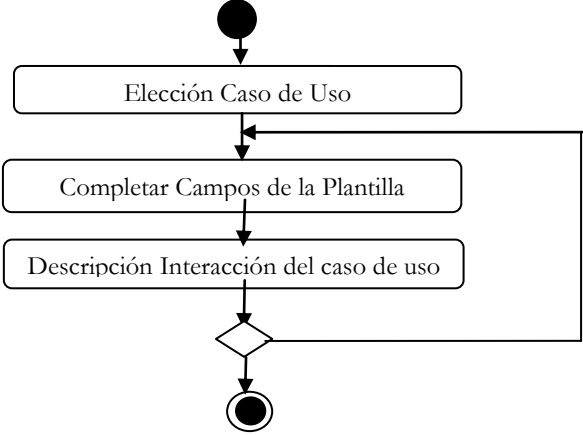
Sección	Descripción										
Name	Patrón de Producto Casos de Uso en formato alto nivel										
Related Patterns	Patrón de Producto Diagrama de Casos de uso Patrón de Producto Casos de Uso en formato expandido										
Initial Context	Se requiere la especificación de la funcionalidad del sistema.										
Result Context	Se obtendrá los casos de uso en formato alto nivel para representar cómo se interactúa con el sistema, cómo se va a usar el sistema y obtener una descripción textual de la dinámica del sistema.										
Problem	Representar la interacción con el sistema y la funcionalidad del mismo.										
Forces	<ul style="list-style-type: none"> • Organización: Pymes. • Sistema: Cualquier tipo. • Paradigma: Orientado a objetos. 										
Solution	 <pre> graph TD Start(()) --> A[Elección Caso de Uso] A --> B[Completar Campos de la Plantilla] B --> C[Descripción Interacción del caso de uso] C --> D{ } D --> B D --> End((())) </pre>										
Roles	Jefe de Proyecto, Analista y Cliente										
Entries	Diagrama de Casos de uso Tipo de Información (.doc.) Se aplica gestión de la configuración del software? sí										
Lessons Learned	En el campo “descripción” detallar la interacción entre los actores identificados y el sistema.										
Templates	<table border="1" data-bbox="529 1294 1321 1491"> <tr> <td>Nombre del Caso de uso</td> <td></td> </tr> <tr> <td>Actores</td> <td></td> </tr> <tr> <td>Tipo</td> <td></td> </tr> <tr> <td>Descripción</td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>	Nombre del Caso de uso		Actores		Tipo		Descripción			
Nombre del Caso de uso											
Actores											
Tipo											
Descripción											
Examples	<table border="1" data-bbox="529 1529 1353 1832"> <tr> <td>Nombre del Caso de uso</td> <td>Añadir Consumición</td> </tr> <tr> <td>Actores</td> <td>Camarero</td> </tr> <tr> <td>Tipo</td> <td>Primario</td> </tr> <tr> <td>Descripción</td> <td></td> </tr> <tr> <td></td> <td>El camarero escribe en su dispositivo el número de la consumición que ha pedido el cliente. Después pulsa el botón aceptar para confirmarlo.</td> </tr> </table>	Nombre del Caso de uso	Añadir Consumición	Actores	Camarero	Tipo	Primario	Descripción			El camarero escribe en su dispositivo el número de la consumición que ha pedido el cliente. Después pulsa el botón aceptar para confirmarlo.
Nombre del Caso de uso	Añadir Consumición										
Actores	Camarero										
Tipo	Primario										
Descripción											
	El camarero escribe en su dispositivo el número de la consumición que ha pedido el cliente. Después pulsa el botón aceptar para confirmarlo.										
....										

Tabla 3-4 Mecanismo de Retroalimentación

Sección	Descripción										
Name	Patrón de Producto Casos de Uso en formato alto nivel para Business Intelligence										
Related Patterns	Patrón de Producto Diagrama de Casos de uso Patrón de Producto Casos de Uso en formato expandido										
Initial Context	Se requiere la especificación de la funcionalidad del sistema.										
Result Context	Se obtendrá los casos de uso en formato alto nivel para representar cómo se interactúa con el sistema, cómo se va a usar el sistema y obtener una descripción textual de la dinámica del sistema.										
Problem	Representar la interacción con el sistema y la funcionalidad del mismo.										
Forces	<ul style="list-style-type: none"> • Organización: Pymes. • Sistema: Business Intelligence. • Paradigma: Orientado a objetos. 										
Solution	<pre> graph TD Start(()) --> A[Elección Caso de Uso] A --> B[Delimitar Indicadores de Análisis] B --> C[Completar Campos de la Plantilla] C --> D[Descripción Interacción del caso de uso] D --> E{ } E --> A E --> End((())) </pre>										
Roles	Jefe de Proyecto, Analista y Cliente										
Entries	Diagrama de Casos de uso Tipo de Información (.doc.) Se aplica gestión de la configuración del software? sí										
Lessons Learned	En el campo “descripción” detallar la interacción entre los actores identificados y el sistema.										
Templates	<table border="1"> <tr> <td>Nombre del Caso de uso</td> <td></td> </tr> <tr> <td>Actores</td> <td></td> </tr> <tr> <td>Tipo</td> <td></td> </tr> <tr> <td><i>Universo/Clase/Objeto</i></td> <td></td> </tr> <tr> <td>Descripción</td> <td></td> </tr> </table>	Nombre del Caso de uso		Actores		Tipo		<i>Universo/Clase/Objeto</i>		Descripción	
Nombre del Caso de uso											
Actores											
Tipo											
<i>Universo/Clase/Objeto</i>											
Descripción											
....										

Tabla 3-5 Mecanismo de Creación de un Nuevo *Patrón de Producto*

3.2 LENGUAJE DE PATRONES DE PRODUCTO

Un lenguaje de patrones (Alexander, 1979) se define como el conjunto de patrones enlazados, de tal forma que cuando se usa un patrón en un proyecto, se desencadena el uso de otros patrones relacionados. En la solución propuesta en esta tesis doctoral es necesario tener un lenguaje de *patrones de producto* para poder tener una referencia conceptual del dominio del problema. El lenguaje de *patrones de producto* no sólo nos va a permitir tener un conjunto de patrones enlazados sino que va a servir como marco conceptual en el diseño de la arquitectura de la solución propuesta. Por ello, se ha extendido el concepto de lenguaje de patrones, definiendo el lenguaje de *patrones de producto* con los siguientes ítems: mecanismos de colaboración entre patrones de producto, el catálogo de *patrones de producto* desarrollado, la regla de búsqueda de *patrones de producto*, el metamodelo para representar *patrones de producto*.

3.2.1 Mecanismos de colaboración entre *Patrones de Producto*

Los mecanismos de colaboración hacen referencia a dos formas de interacción: la manera en que los *patrones de producto* interactúan entre sí y la manera en que el usuario se comunica con el sistema a la hora de realizar un proceso software a través del *patrón de producto*. Gracias a estos dos mecanismos de colaboración se obtendrá la comunicación total entre los miembros de un equipo en el desarrollo de los proyectos software. Los mecanismos de colaboración entre *patrones de producto* identificados en esta tesis doctoral son:

- *Colaboración entre patrones de producto*. Este mecanismo modela la forma en el que los *patrones de producto* interactúan entre sí, definiendo el orden de precedencia entre los *patrones de producto*. Esto es posible ya que los *patrones de producto* son independientes del método o proceso y por tanto son independientes e interoperables. Para definir el orden de precedencia entre dos *patrones de producto*, se realizará la siguiente comprobación (Figura 3-2):

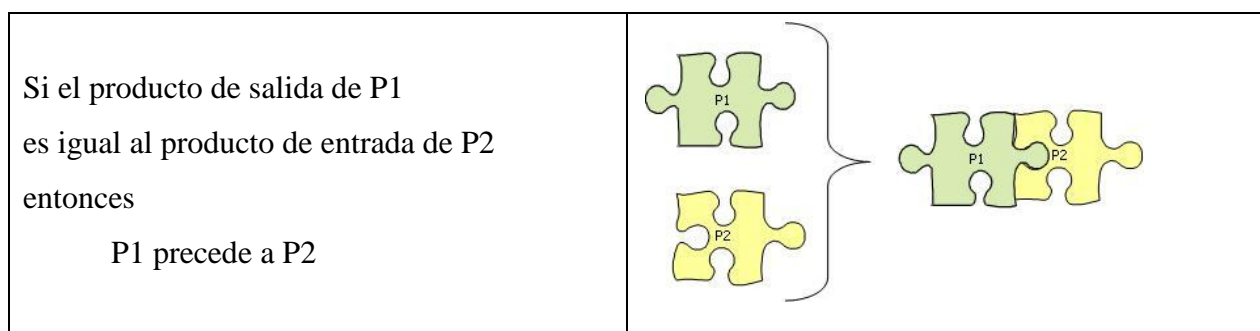


Figura 3-2 Mecanismo de Colaboración entre *Patrones de Producto*

- *Colaboración entre los roles involucrados en el desarrollo de un patrón de producto.* Este mecanismo modela cómo interactúan e intervienen los recursos humanos recomendados en el campo “Roles” del *patrón de producto* para realizar una actividad. Dicha colaboración está modelada en el diagrama de actividades del campo “Solution” del *patrón de producto*. Esta interacción está automatizada mediante una herramienta colaborativa que proporciona a los usuarios un entorno colaborativo para el desarrollo de las actividades y procesos de los proyectos software.

3.2.2 Catálogo de Patrones de Producto

La solución propuesta proporciona la reutilización de activos de procesos a través de *patrones de producto* soportado por tecnología colaborativa para mejorar la eficiencia de uso de sus procesos software y la calidad de sus productos. Para ello, esta solución proporciona una librería de activos de proceso que proporciona un repositorio de conocimiento central para adquirir, definir y difundir las guías y el conocimiento de los ingenieros expertos para la realización de los productos software, facilitando mecanismos para compartir el conocimiento sobre los activos de proceso y cómo éstos son usados. Esta PAL ha sido implementada mediante una wiki (PIBOK-PB-wiki: Process Improvement base don Knowledge – Patterns based- wiki) que contiene el repositorio de conocimiento de activos de procesos de los proyectos software. Las wikis son propuestas por (Cunningham et al, 2001) (Chau et al, 2005) (Gonzalez-Reinhart, 2005) como una herramienta de compartición de conocimiento en las organizaciones ya que por sus características tecnológicas permite la comunicación, colaboración y la documentación del conocimiento.

La estructura de la wiki definida en esta tesis doctoral está compuesta por los siguientes elementos:

- **Modelos de Procesos Software:** en esta sección de la wiki están los modelos de procesos software de la ingeniería del software. De cada uno de los modelos se describen las actividades del modelo. Dichas actividades van a utilizar los *patrones de producto* como base de conocimiento para realizar la actividad y obtener los productos software.
- **Patrones de Producto:** almacenan el conocimiento tácito de los ingenieros expertos para obtener productos software. Esta sección está compuesta por los diferentes tipos de *patrones de producto*: patrones de producto de las fases de análisis, diseño y gestión de

un proyecto software y los *patrones de producto* de diferentes metodologías. Por cada uno de los tipos de *patrones de producto* se especifican todos los *patrones de producto* de cada una de las fases y/o metodología y se describe cada uno de los *patrones de producto* con todos sus campos (definidos en el formato de *patrón de producto*). El *patrón de producto* proporcionará una solución mediante un diagrama de actividades con los pasos para realizar dicha actividad, plantillas para tener como base para la realización de la actividad y ejemplos que son los productos de las actividades realizadas mediante dicho *patrón de producto*.

- **Proyectos Software:** esta sección contiene los proyectos software desarrollados. Para cada uno de los proyectos se especifica las actividades llevadas a cabo en cada proyecto junto con los productos software obtenidos de cada una de las actividades. Además, se indica si la actividad ha sido desarrollada con el conocimiento proporcionado por algún *patrón de producto*, y si es así, se indica qué *patrón de producto* ha sido utilizado.

En el mapa conceptual de la wiki, ver Figura 3-3, se ilustra la interrelación entre los conceptos que componen la wiki. La wiki como se ha explicado anteriormente, está compuesta de Modelos de Procesos Software, *Patrones de Producto* y Proyectos Software. En la sección de los proyectos software se van a almacenar todos los proyectos software que son desarrollados en la organización. Para desarrollar cada uno de estos proyectos se va a utilizar un modelo de proceso que está almacenado en la sección de la wiki denominada Modelos de Procesos. Cada modelo de proceso está compuesto de un conjunto de actividades. Por tanto, el proyecto software instanciará dichas actividades del modelo de proceso para obtener sus productos software. Dichas actividades se van a ejecutar con la ayuda de los *patrones de producto*, que almacenan el conocimiento tácito de cómo se realiza cada actividad y proporciona un diagrama de actividad que contiene los pasos que se tiene que seguir para obtener dicho producto software, de plantillas que ayudan a la realización de dicha actividad y de ejemplos, que a su vez son los productos software obtenidos de la ejecución de esa actividad con el *patrón de producto*, en otros proyectos software.

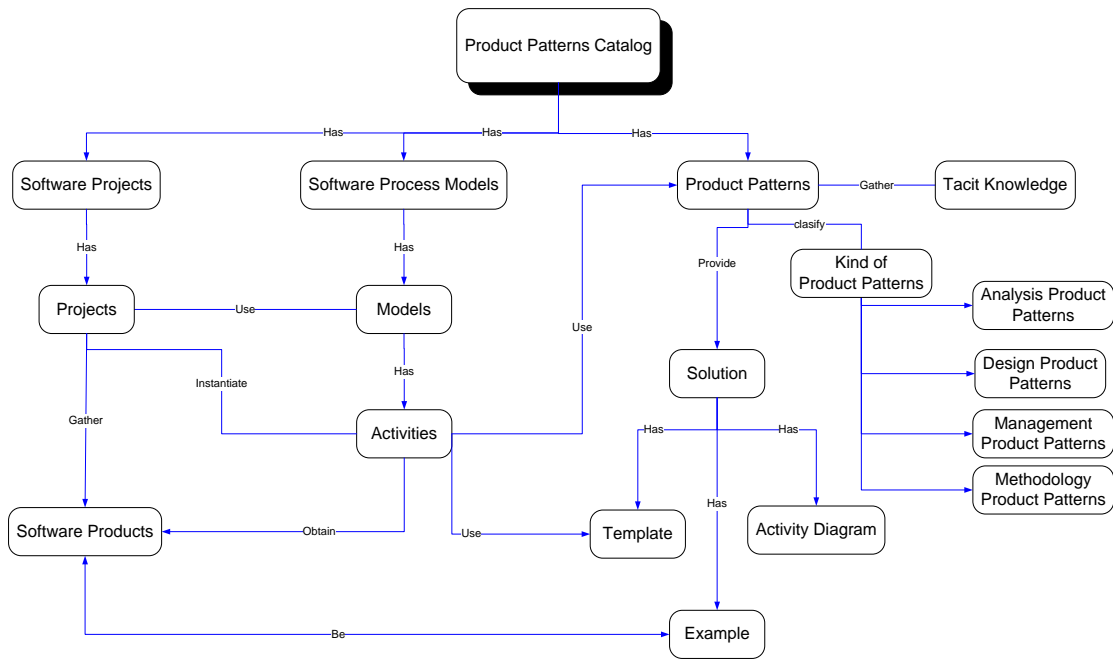


Figura 3-3 Mapa Conceptual de la wiki

La wiki desarrollada en esta solución se ha implementado mediante la herramienta de open source MindTouch Core. La url de la página web de la wiki es: <http://productpatterns.sel.inf.uc3m.es>. En la Figura 3-4 se muestra el *patrón de producto* del Diagrama de Casos de Uso.

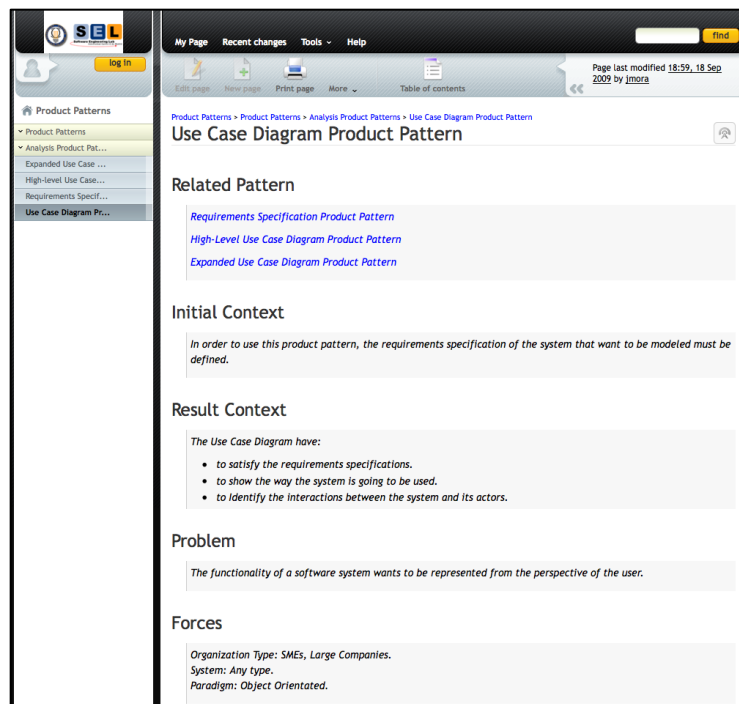


Figura 3-4 PIBOK-PB- wiki

En la Figura 3-5 se muestra el mapa navegacional de la wiki.

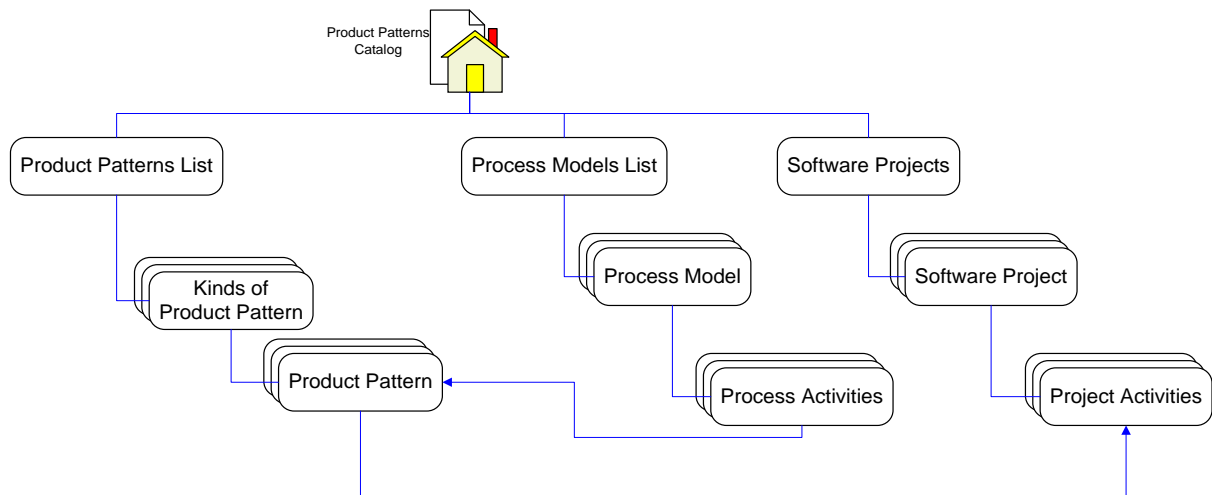


Figura 3-5 Mapa Navegacional de la wiki

En el catálogo de *patrones de producto* se han definido 38 *patrones de producto*, que se han clasificado en:

- Fases genéricas del ciclo de desarrollo software:

Fase de Análisis: 4 *patrones de producto*

Fase de Diseño: 4 *patrones de producto*

Fase de Gestión: 4 *patrones de producto*

- Métodos del desarrollo software:

Craig Larman: 8 *patrones de producto*

Extreme Programming: 26 *patrones de producto*

3.2.3 Búsqueda de Patrones de Producto

Los *patrones de producto* proporcionan el conocimiento de las mejores prácticas de la ingeniería del software para obtener un producto software específico así como el conocimiento tácito de los ingenieros expertos en los procesos de desarrollo de proyectos software en las organizaciones para su posterior reutilización.

Para poder reutilizar el conocimiento, se tiene que realizar un proceso de búsqueda de los patrones de producto que mejor se adapten a los diferentes parámetros de contexto y restricciones (“forces”) de la organización y del proyecto. Es decir, se va a realizar la búsqueda de los patrones de producto que proporcione como salida los pasos sistemáticos que se tienen que realizar para la consecución del proceso y poder obtener el producto software. Por tanto, el sistema tendrá que ejecutar la siguiente regla para poder realizar la búsqueda en el catálogo de patrones de producto: si para realizar una actividad (“problem”), la actividad se encuentra en un contexto determinado (“context”) y cumple unas determinadas restricciones (“forces”), entonces se identifica el/los *patrones de producto* que cumplan con dichas premisas.

En formato de premisas, la regla es la siguiente:

<p>If you find yourself in this context (and) with this problem (and) entailing these forces then map a product pattern in your project (and) look for more product patterns</p>

Una vez ejecutada dicha regla, se obtendrán todos los *patrones de producto* que cumplan dichas premisas. La decisión de qué *patrón de producto* se instanciará de todos los proporcionados por esta regla, la adoptará el jefe de proyecto con la información obtenida de cada uno de los *patrones de producto* y de su propia experiencia.

3.2.4 Metamodelo para representar Patrones de Producto

El metamodelo define un lenguaje que sirve para modelar la relación lógica entre los *patrones de producto* y la lógica del desarrollo de proyectos en las organizaciones. En la Figura 3-6 se ha representado dicho meta-modelo mediante el lenguaje unificado de Modelado UML (Unified Modeling Language).

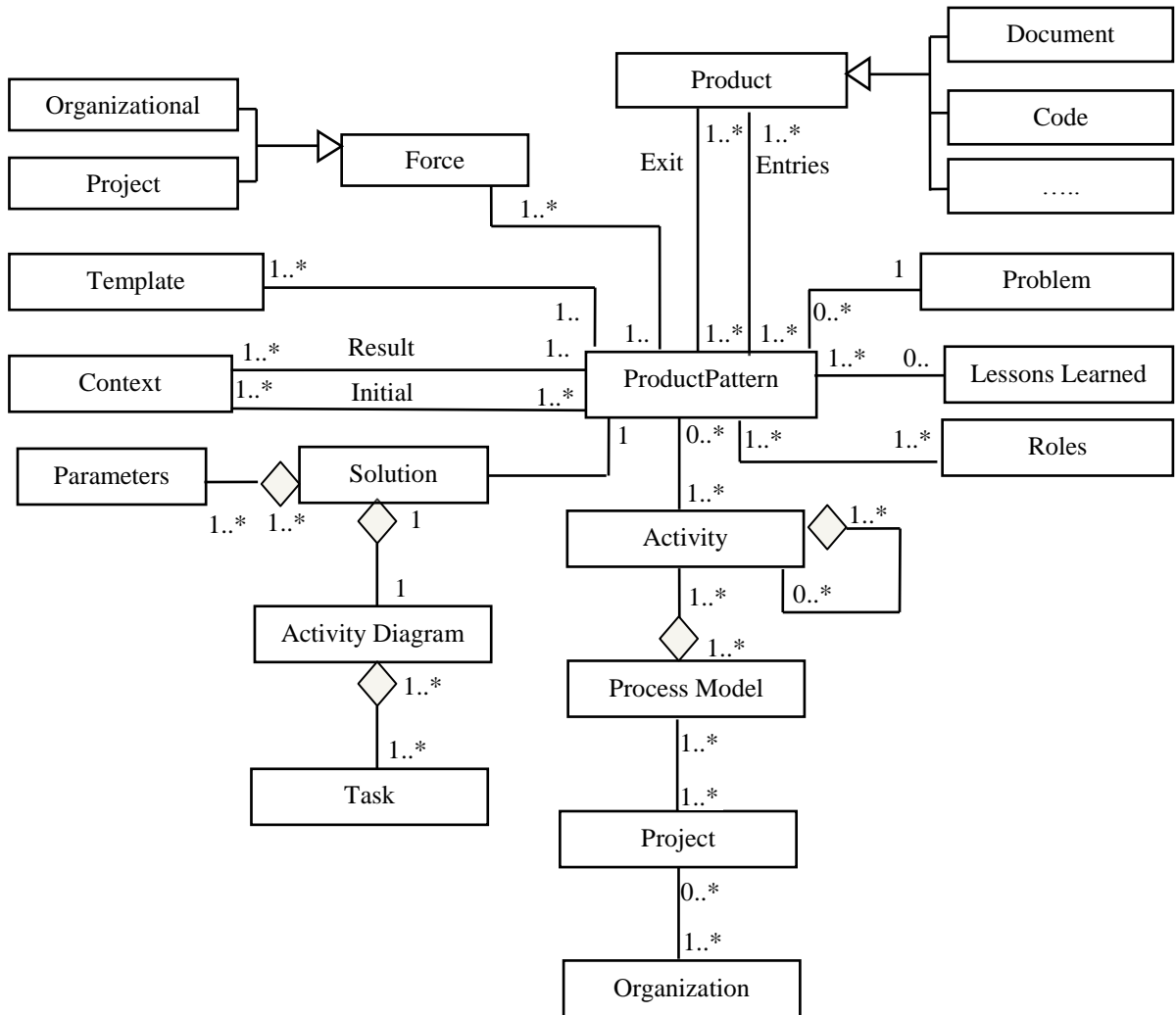


Figura 3-6 Metamodelo *Patrones de Producto*

A continuación se va a describir dicho meta-modelo por partes.

Las organizaciones van a desarrollar de 0 a n proyectos, considerando un proyecto como el esfuerzo temporal (con un inicio y un fin definido) realizado para la creación de un producto o servicio único. Los proyectos se realizarán siguiendo unos modelos de proceso, es decir, unos marcos de referencia que contienen los procesos, las actividades y sub-actividades o tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de sus procesos, ver Figura 3-7.

Para poder llevar a cabo estos procesos, se le proporcionan al sistema las mejores prácticas de la ingeniería del software y el conocimiento tácito que poseen los expertos de la

organización sobre cómo realizar cada uno de los procesos y actividades de los modelos de proceso. Por tanto, cada una de las actividades podrá tener asociado *patrones de producto*. El uso de un *patrón de producto* en un proyecto software depende de su conexión a una actividad del proyecto, ver Figura 3-7.

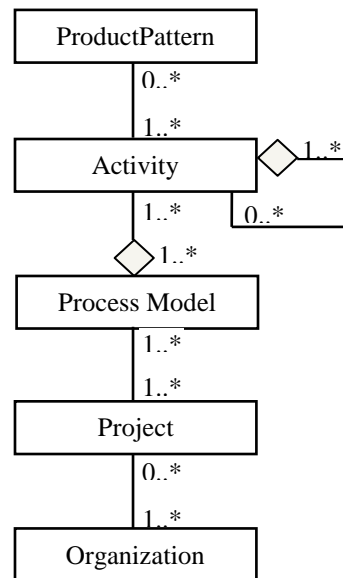


Figura 3-7 Relación entre *Patrones de Producto*, *Actividad*, *Organización*, *Proyecto* y *Modelos de Proceso*

El *patrón de producto* se va a aplicar en un problema, bajo un contexto inicial del sistema y unas restricciones de la organización y del proyecto a desarrollar. Siguiendo con el ejemplo, las restricciones del proyecto sería que este *patrón de producto* se puede aplicar en proyectos que se estén desarrollando bajo un paradigma orientado a objeto tanto en grandes organizaciones como en pequeñas y medianas empresa. El producto de salida también saldrá bajo un contexto resultante del sistema de haber aplicado dicho patrón. Ver Figura 3-8.

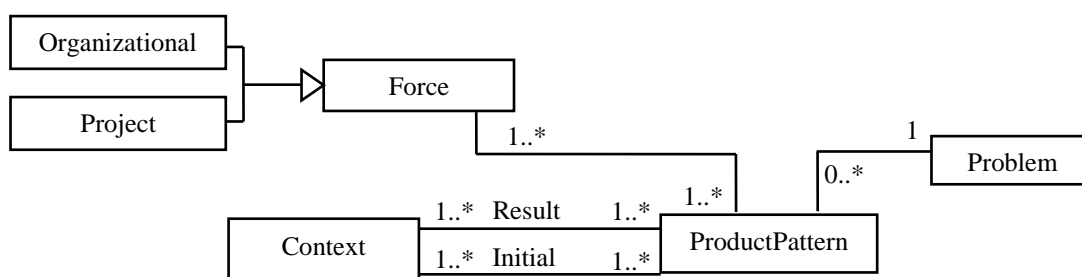


Figura 3-8 Relación entre los campos: *Problema*, *Forces* y *Contexto* del *Patrón de Producto*

El *patrón de producto* necesita productos de entrada para la realización del mismo y proporciona como salida un producto de salida. Por ejemplo, en el caso del *patrón de producto Patrón Casos de Uso en formato alto nivel*, el patrón necesita el producto “diagrama de casos de uso” para poder realizar dicha actividad. Como salida, produciría otro producto, que en este ejemplo sería el documento de “diagrama de casos de uso de alto nivel”. Ver Figura 3-9.

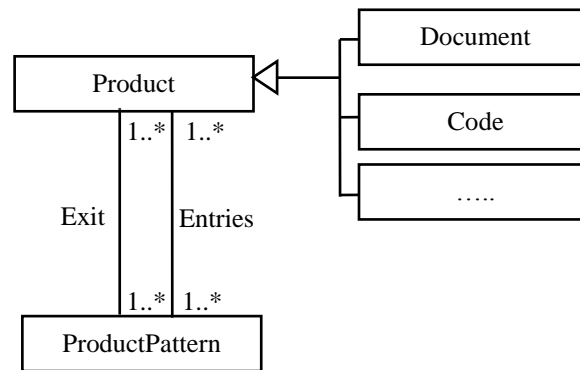


Figura 3-9 Relación entre los campos Entrada y Salida del *Patrón de Producto*

Para obtener los productos de salida, el *patrón de producto* tiene asociado una única solución que está compuesta de un diagrama de actividades y de parámetros relacionados con el desarrollo de la solución de dicho patrón en proyectos similares. El diagrama de actividades está compuesto de todas las tareas que tiene que realizar los roles implicados en dicho patrón para la obtención de los productos resultantes. En todos aquellos *patrones de producto* en los que se pueda aportar una plantilla para facilitar la realización de dicho patrón, se añadirá la plantilla como base para que el usuario tenga el formato con el que tiene que realizar los pasos del patrón. Estas plantillas proporcionarán el formato de salida del producto en los casos que sean posibles. En el ejemplo expuesto, se puede observar cómo el producto de salida, un documento Word, estará formado por la plantilla proporcionada en el patrón, ya que el jefe de proyecto para realizar dicho patrón tiene que ir rellenando la plantilla proporcionada en el mismo. Además, el *patrón de producto* proporcionará las lecciones aprendidas de haber realizado dicha actividad en proyectos previos. Ver Figura 3-10.

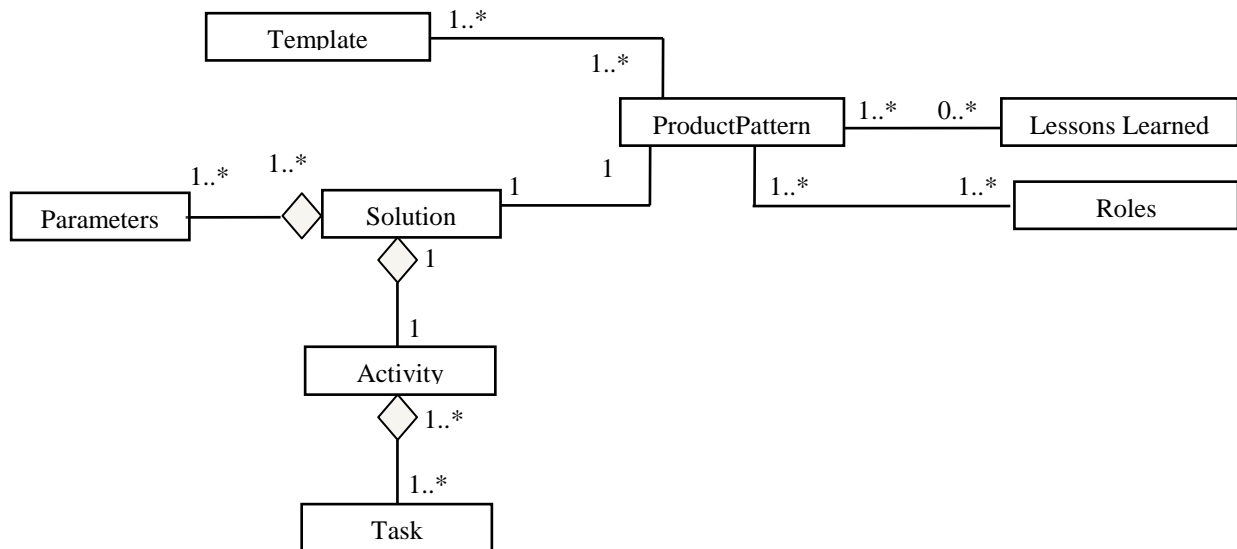


Figura 3-10 Relación entre los campos Plantilla y Solución del *Patrón de Producto*

3.3 ESTRATEGIA CORPORATIVA DE DESPLIEGUE DE LA SOLUCIÓN

3.3.1 Visión general

Esta tesis doctoral propone una estrategia corporativa que proporciona un conjunto de procesos necesarios para poder realizar la difusión, uso y reutilización de los activos de proceso en las organizaciones. Dicha estrategia corporativa consta de dos fases:

- La primera fase es previa a la difusión, uso y reutilización de los procesos software en las organizaciones. En esta fase se realiza el proceso de “Educción de las Mejores Prácticas en la Ingeniería del Software”.

- La segunda fase es la implantación de los procesos software necesarios para poder realizar la difusión, uso y reutilización de los activos de proceso en las organizaciones.

En la Figura 3-11 se muestra la estrategia corporativa del despliegue de la solución.

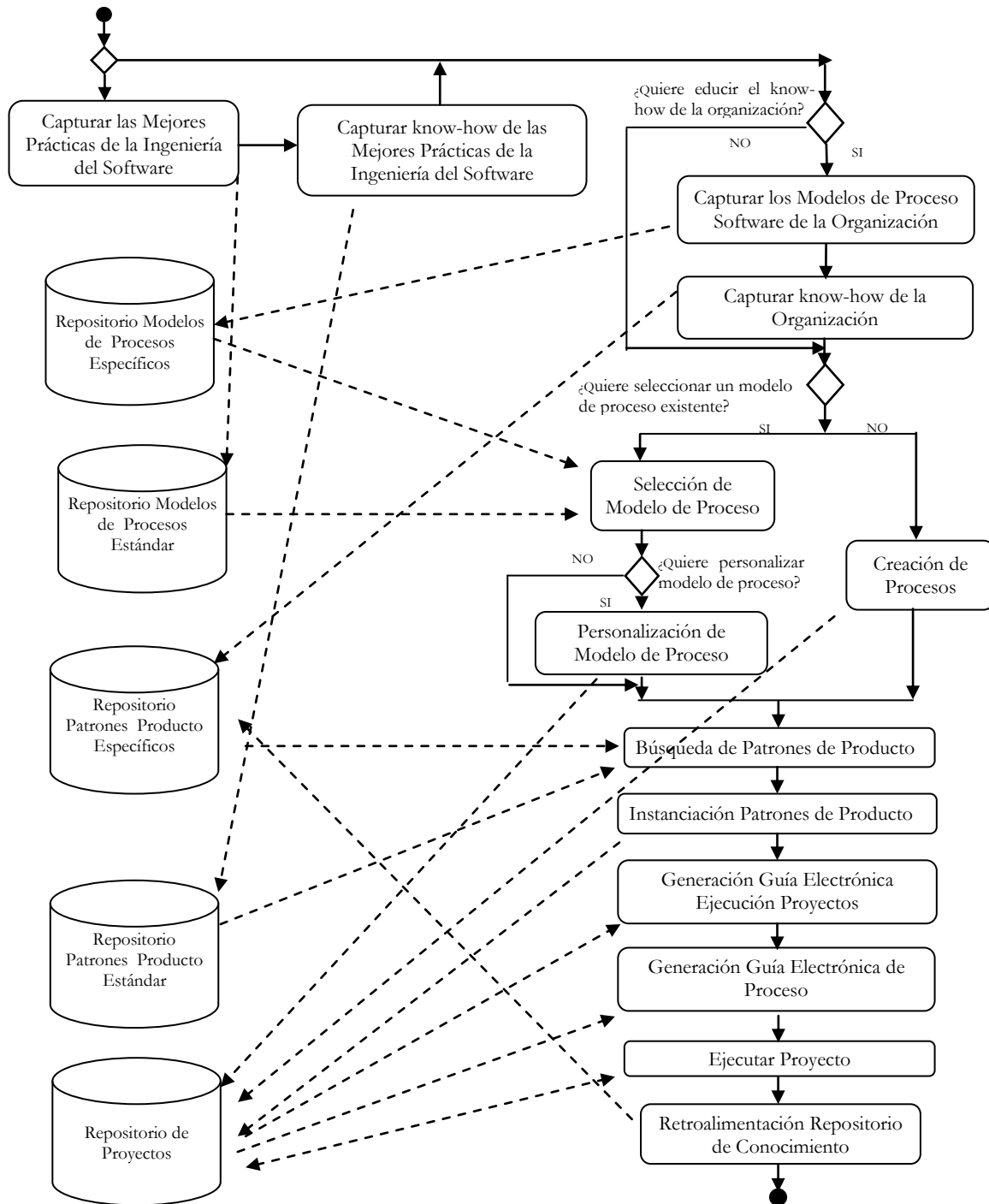


Figura 3-11 Estrategia Corporativa del despliegue de la solución

3.3.2 Fase I: Educción de las Mejores Prácticas en la Ingeniería del Software

Esta fase va a proporcionar el conocimiento explícito y tácito de las mejores prácticas que existen en el campo de la Ingeniería del Software. Esta fase está compuesta de dos procesos:

- Proceso Capturar las mejores prácticas de la Ingeniería del Software
- Proceso Capturar el know-how de las Mejores Prácticas de la Ingeniería del Software

3.3.2.1 Proceso Capturar las Mejores Prácticas de la Ingeniería del Software.

Descripción

Este proceso captura las mejores prácticas de la Ingeniería del Software. Ha sido un proceso sistemático de recogida de información de modelos de procesos y metodologías en el campo de la Ingeniería del Software. Las actividades que se han llevado a cabo en el proceso han sido: la búsqueda de referencias de estándares, modelos de procesos y metodologías; el análisis y extracción de sus actividades, roles, técnicas para realizar las actividades y el almacenamiento de dicha información en el repositorio de Modelos de Procesos Estándar. Una vez realizado este proceso, en el repositorio se tiene almacenado el conocimiento explícito de los modelos de procesos y metodologías analizadas. Los pasos de este proceso se recogen en la Figura 3-12.

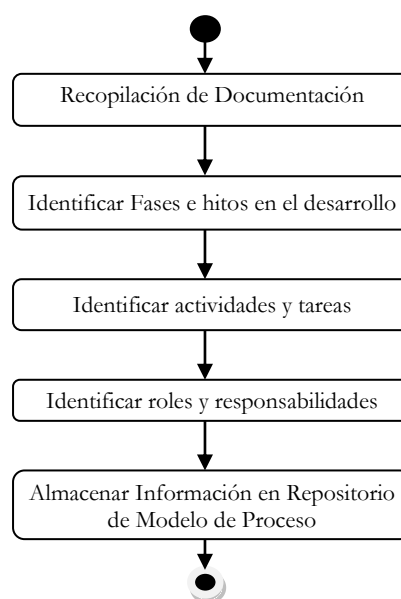


Figura 3-12 Proceso Capturar las Mejores Prácticas de la Ingeniería del Software

Aportación

Actualmente se siguen estudiando cuales son los problemas que influyen en la pobre calidad del software. Uno de los indicadores que se manifiestan es el poco uso de las mejores prácticas en los proyectos software debido a que este conocimiento no es almacenado de forma electrónica. Hay trabajos de investigación donde recogen la importancia de llevar las mejores prácticas de la Ingeniería del Software a las organizaciones.

La aportación que se realiza en esta actividad es el análisis y almacenamiento de los modelos de procesos y metodologías de las mejores prácticas definidas en la Ingeniería del Software. Dicha información se proporcionará al usuario de forma electrónica.

3.3.2.2 Proceso Capturar el know-how de las Mejores Prácticas de la Ingeniería del Software

Descripción

Este proceso define los pasos sistemáticos que se tiene que realizar para ejecutar los procesos y actividades identificadas en los modelos de proceso y metodologías de desarrollo de procesos software. Por cada una de las actividades identificadas en los Modelos de procesos y metodologías de desarrollo, y representado en un árbol de proceso, se estudia y analiza toda la información referente a dicha actividad, así como sus lecciones aprendidas. A partir de este conocimiento se realizará un proceso de educación del conocimiento del experto en el área de la Ingeniería del Software y se representará dicho conocimiento tácito en el *patrón de producto*, mediante un diagrama de actividad con los pasos que un usuario tiene que realizar para poder llevar a cabo las actividades del diagrama de actividad y poder obtener el producto de salida de dicha actividad. Los pasos de este proceso se recogen en la Figura 3-13.

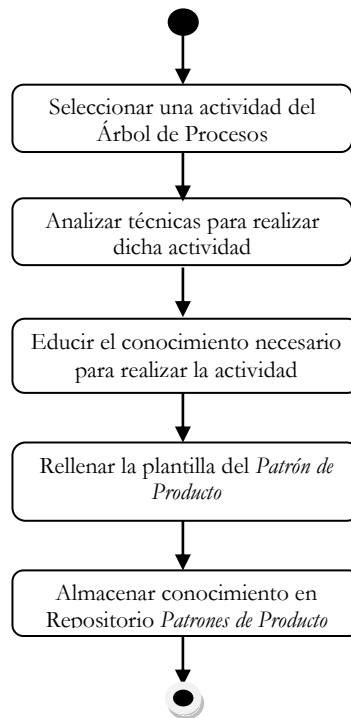


Figura 3-13 Proceso Capturar el know-how de las Mejores Prácticas de la Ingeniería del Software.

Aportación

La aportación de esta actividad es la utilización del *patrón de producto* como artefacto para almacenar el conocimiento para poder obtener un producto software. La utilización de los patrones para almacenar un determinado proceso ha sido utilizada en algunos trabajos de investigación. Aunque en todos esos trabajos únicamente se llegaba a realizar un catálogo de patrones de forma textual. La aportación que se realiza con esta actividad es tener almacenado de forma electrónica las mejores prácticas de la Ingeniería del Software haciéndolas más usable en los procesos software.

3.3.3 Fase II: Implantación de los Procesos Software

Esta fase va a proporcionar los procesos necesarios para que las organizaciones puedan realizar la difusión, uso y reutilización de los activos de proceso.

Esta fase está compuesta por los siguientes procesos:

- Proceso Capturar los Modelos de Proceso Software de la Organización.
- Proceso Capturar el Know-How de las Organizaciones.
- Proceso Selección de Modelo de Proceso.
- Proceso de Personalización de Modelo de Proceso.

- Proceso de Creación de Procesos.
- Proceso Búsqueda de Patrones de Producto.
- Proceso de Instanciación de Patrones de Producto.
- Proceso Generación de una Guía electrónica de ejecución de Proyecto.
- Proceso Generación de una Guía electrónica de Procesos.
- Proceso de Ejecución de Proyecto.
- Proceso de Retroalimentación del Repositorio de Conocimiento.

3.3.3.1 Proceso Capturar los Modelos de Proceso Software de la Organización

Descripción

En este proceso se recogen los modelos de proceso y metodologías de la organización. Para ello, se tendrán reuniones con los ingenieros expertos del cliente para recopilar la documentación donde se definen los modelos de proceso así como toda la información relacionada: las fases e hitos, roles y responsabilidades, tareas, entregables, otros documentos (entregables internos) y registros.

La técnica para llevar a cabo esta actividad es la entrevista. A partir de la documentación de la metodología y modelos de procesos de la organización, se llevan a cabo la identificación de las actividades y se almacenará dicha información en el repositorio de Modelos de Procesos Específicos. Los pasos de este proceso se recogen en la Figura 3-14.

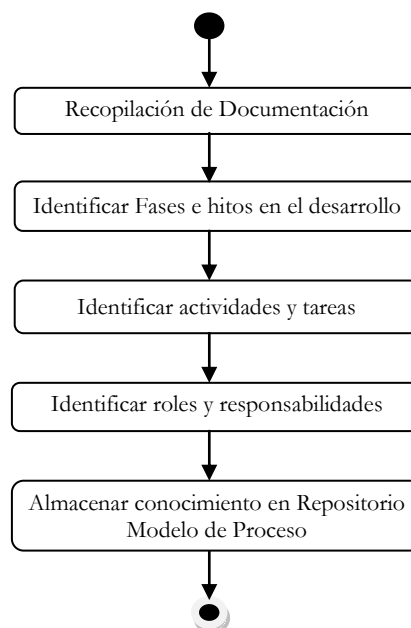


Figura 3-14 Proceso Capturar los Modelos de Proceso Software de la Organización

Aportación

La aportación de esta actividad es el análisis de la información capturada en las organizaciones para llevarlo de una forma estructurada a un repositorio de conocimiento. Muchas empresas tienen definido sus propias metodologías y modelos de proceso de forma textual en documentos difíciles de leer y que el propio empleado no es capaz de entender ya que no tiene un formato usable, lo que provoca que dichos modelos y metodologías no sean utilizadas por los empleados en los desarrollos software.

Este proceso proporciona un árbol de procesos y actividades donde los usuarios de forma electrónica pueden encontrar toda la información relacionada con el modelo de proceso o metodología a seguir en su organización.

3.3.3.2 Proceso Capturar el Know-How de las organizaciones

Descripción

Este proceso proporciona el conocimiento que los diferentes roles tienen que realizar para poder llevar a cabo sus actividades. Para extraer este conocimiento, se analiza cómo se realizan cada una de las actividades para poder proporcionar y reutilizar dicho conocimiento en todos los proyectos software con un máximo de productividad y mínimo coste y tiempo.

Para realizar la captura de este conocimiento se realizan: reuniones con el cliente, a diferentes niveles de jerarquía organizativa para poder adquirir la metodología utilizada y, un ingeniero experto trabaja in-situ con la persona que realiza dicha actividad; de esta forma, se puede extraer todo el conocimiento del rol implicado, ya que no siempre se transmite en una reunión las tareas que uno lleva a cabo para realizar dicha actividad. Es en este paso de la metodología donde se da valor al proceso, ya que se detalla cómo se tiene que realizar una actividad. Este conocimiento se almacena en el *patrón de producto*. Los pasos de este proceso se recogen en la Figura 3-15.

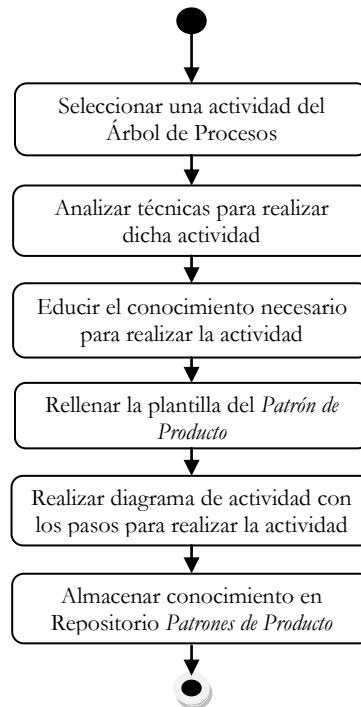


Figura 3-15 Proceso Capturar el Know-How de las Organizaciones

Aportación

La aportación de esta actividad es la utilización del *patrón de producto* como artefacto para almacenar el conocimiento para poder realizar un producto software. El *patrón de producto* proporciona el conocimiento tácito de los ingenieros para su reutilización en otros proyectos software. Como se ha mencionado anteriormente, dicho conocimiento se almacena en un catálogo de patrones de producto accesible por los usuarios que están desarrollando el proyecto software.

3.3.3.3 Proceso de Selección de Modelo de Proceso

Descripción

Este proceso permite que el usuario, en este caso jefe de proyecto, pueda seleccionar un modelo de proceso de los propuestos por el sistema en base a características de la organización y del proyecto a desarrollar. Para ello, el sistema mostrará al jefe de proyecto parámetros de la organización y del proyecto para que el sistema en función de heurísticas proporcione al jefe de proyecto los modelos de procesos que más se adecuen a dichos parámetros. El jefe de proyecto

con la información que le proporciona el sistema de los modelos de procesos seleccionados y con su experiencia, será el encargado de tomar la decisión de qué modelo de procesos instanciar de los seleccionados. La información que se proporciona de los modelos de proceso es: un árbol de procesos, actividades y tareas, en el que se muestra gráficamente las actividades y tareas que se deben realizar en cada modelo de proceso; e información adicional a la que puede acceder para poder tomar la decisión adecuada de qué modelo de proceso elegir para realizar un determinado proyecto. Los pasos de este proceso se recogen en la Figura 3-16.

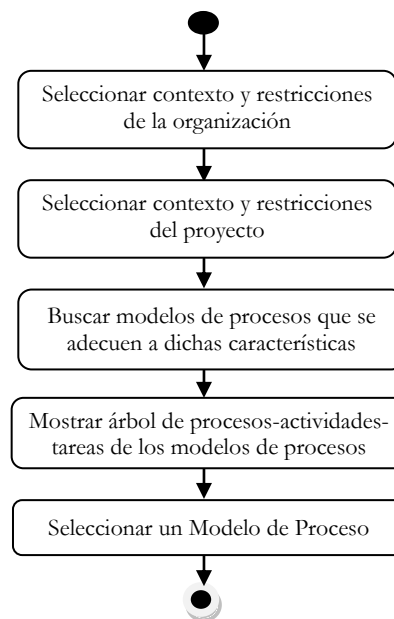


Figura 3-16 Proceso Selección de Modelo de Proceso

Aportación

Actualmente, existen softwares, como por ejemplo Virtual Team System, que también proporcionan modelos de proceso para el desarrollo de proyectos software. La diferencia fundamental es que dichos software, no proporciona al jefe de proyecto qué modelo de proceso se adecua más a sus necesidades dependiendo de las características del contexto y las restricciones, tanto de la organización que va a desarrollar el software como del proyecto a desarrollar proporcionándole al jefe de proyecto el árbol de procesos, actividades y tareas del modelo de proceso software o metodología seleccionada.

3.3.3.4 Proceso de Personalización de Modelo de Proceso

Descripción

Una vez seleccionado el modelo de proceso, el jefe de proyecto podrá personalizar dicho modelo, para ello, podrá añadir y eliminar actividades y tareas a dicho modelo de proceso. Los pasos de este proceso se recogen en la Figura 3-17.

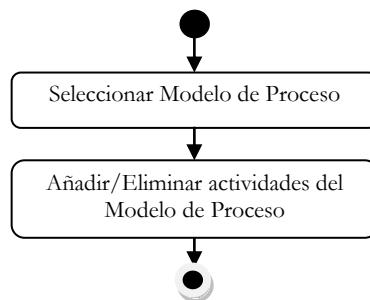


Figura 3-17 Proceso Personalización de Modelo de Proceso

Aportación

El sistema le proporciona una base de conocimiento de metodologías y técnicas que puede utilizar para la personalización del modelo de proceso. El jefe de proyecto tendrá que definir el árbol de actividades y tareas de su nuevo modelo de proceso, así como definir el producto que se obtiene en cada uno de ellos. Si el jefe de proyecto selecciona alguna metodología o técnica del repositorio de modelo de proceso, toda la información referente al mismo se asignará al nuevo modelo de proceso.

3.3.3.5 Proceso de Creación de Procesos

Descripción

Esta tarea proporciona la opción de creación de los procesos y actividades de la organización personalizando y priorizando cada uno de los procesos y actividades. Los pasos de este proceso se recogen en la Figura 3-18.

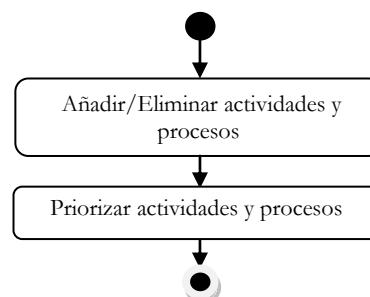


Figura 3-18 Proceso Creación de Procesos y Actividades de la Organización

Aportación

Este proceso permite que los jefes de proyecto no se tengan que limitar a los modelos de procesos estándar, si no que aquellas organizaciones que tienen sus propios modelos de procesos y metodologías puedan desarrollar sus productos en base a ésta teniendo todos los datos e información de dicha metodología o modelo de proceso de forma electrónica.

3.3.3.6 Proceso Búsqueda de Patrones de Producto

Descripción

Esta tarea realiza la el proceso de búsqueda de patrones de producto de cada una de las actividades que forman el árbol de actividades y tareas del modelo de proceso seleccionado o definido en las tareas anteriores. Los pasos de este proceso se recogen en la Figura 3-19.

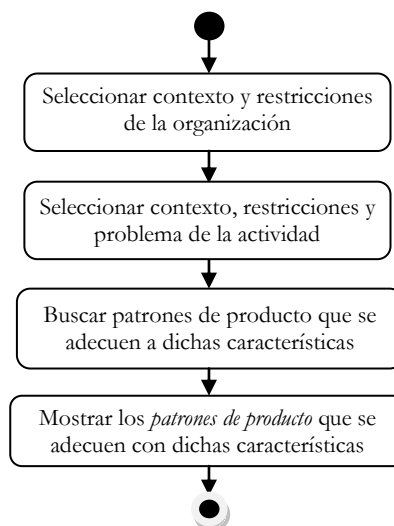


Figura 3-19 Proceso Búsqueda de Patrones de Producto

Aportación

El sistema proporcionará para cada una de las actividades y tareas a realizar, los *patrones de producto* encontrados para poder realizar dicha actividad, pudiendo ser más de un *patrón de producto* para una misma actividad. La búsqueda de los *patrones de producto*, se basa en el problema a resolver, el contexto inicial y las restricciones. El contexto inicial y las restricciones se comprobarán con las características de la organización y de la empresa que el jefe de proyecto ha proporcionado al sistema.

El sistema proporcionará información relativa a cada uno de los *patrones de producto* asociados a cada una de la actividad, entre ellos estará, el diagrama de actividad que ilustra de forma gráfica que pasos tiene que seguir el usuario para obtener un determinado producto, el tiempo estimado para realizar dicho patrón, los productos de entrada necesarios para la realización de este patrón así como información que ayuda a realizar el *patrón de producto*.

3.3.3.7 Proceso de Instanciación de Patrones de Producto

Descripción

Este proceso realiza la instanciación de los *patrones de producto*. Si en una actividad hubiera más de un *patrón de producto* se define cuál de ellos se va a utilizar para la realización de la actividad. Los pasos de este proceso se recogen en la Figura 3-20.

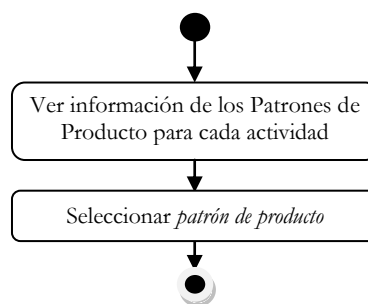


Figura 3-20 Proceso de Instanciación de Patrones de Producto

Aportación

En el caso de que varios *patrones de producto* se puedan instanciar para realizar una misma actividad, el jefe de proyecto seleccionará uno de los patrones de producto pudiéndose realizar esta elección de forma manual o automática. Si se realiza de forma automática, se aplicará un grado de ajuste que mide cómo de adecuado es cada uno de los patrones para esa actividad teniendo en cuenta las restricciones del sistema y de la organización así como el contexto en el que se encuentra.

El sistema proporcionará para cada una de las actividades el conocimiento almacenado en dichos patrones de producto, indicando qué pasos se tienen que hacer y cómo se realizan para poder obtener el producto de cada actividad. Se mostrará de forma electrónica y gráficamente los patrones instanciados.

3.3.3.8 Proceso Generación de una Guía electrónica de ejecución de Proyecto

Descripción

Esta tarea proporciona una guía electrónica de ejecución de proyecto. Para ello se realizará mediante una técnica de planificación de gestión de proyecto, la planificación del proyecto software a desarrollar. Así mismo, la guía proporcionará la información del tiempo estimado de cada actividad del proyecto y los roles asignados a cada actividad con toda la información necesaria para la ejecución del proyecto. Dichos datos se importarán a una plataforma colaborativa donde se va a desarrollar el proyecto. Esta plataforma colaborativa proporcionará todas las funcionalidades colaborativas que se requiere para la realización de las actividades de forma on-line y off-line. Los pasos de este proceso se recogen en la Figura 3-21.



Figura 3-21 Proceso Generación de una Guía electrónica de ejecución de Proyecto

Aportación

Mediante la guía electrónica de ejecución de proyecto se proporciona una visión global del proyecto a todo el equipo de desarrollo. Esto no siempre se proporciona en los proyectos software, donde por lo general, cada rol tiene sus actividades pero no tiene la visión de cómo su actividad encaja con el resto de las actividades del proyecto. Además, proporcionará de funcionalidad colaborativa a las actividades del proyecto.

3.3.3.9 Proceso Generación de una Guía electrónica de Procesos

Descripción

El sistema proporcionará una guía electrónica de proceso por cada actividad del proyecto. Para ello, se habilitará un área de trabajo colaborativo para la realización de dicha actividad. En dicho área, se debe proporcionar acceso directo al repositorio de activos de proceso (PIBOK-PB-PAL) para que en todo momento los encargados de realizar dicha actividad puedan tener el conocimiento accesible. Además, la plataforma proporcionará las funcionalidades colaborativas que se requiera para la realización de la actividad, como editor de texto en el que puedan trabajar colaborativamente en un producto software, repositorios de documentos al que puedan acceder los miembros del equipo, control de versiones, aprobación y/o rechazo de documentos, etc.

Los pasos de este proceso se recogen en la Figura 3-22.

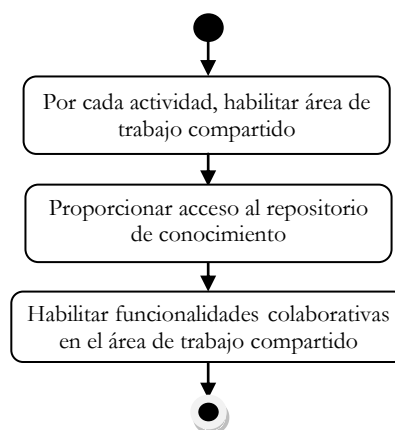


Figura 3-22 Proceso Generación de una Guía electrónica de Procesos

Aportación

Mediante la guía electrónica de proceso se proporciona un área donde los miembros del equipo puedan trabajar colaborativamente de forma on-line y off-line en el desarrollo de las actividades. En dicho área se proporciona el conocimiento explícito y tácito de los activos de proceso y proyectos de la organización para su reutilización en el desarrollo de dicha actividad. Además, la plataforma colaborativa proporcionará las funcionalidades colaborativas que se requiera en cada una de las actividades.

3.3.3.10 Proceso de Ejecución de Proyecto

Descripción

La ejecución del proyecto se realizará en la plataforma colaborativa que proporcionará las funcionalidades colaborativas y de las guías electrónicas de ejecución de proyecto y de proceso, para el desarrollo de las actividades del proyecto de forma colaborativa.

Los pasos de este proceso se recogen en la Figura 3-23.

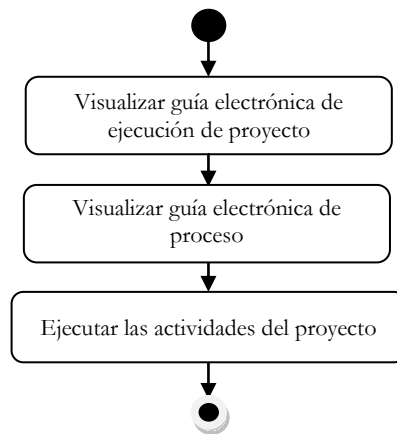


Figura 3-23 Proceso Ejecución de Proyecto

Aportación

Esta solución proporciona que todos los miembros del equipo de desarrollo software puedan desarrollar sus actividades en una plataforma colaborativa con el conocimiento de los ingenieros software. Para ello, se proporciona de las funcionalidades colaborativas que aporta la plataforma así como de las guías electrónicas de ejecución de proyecto y de proceso en áreas de trabajo compartido para el desarrollo de las actividades.

3.3.3.11 Proceso de Retroalimentación del Repositorio de Conocimiento

Descripción

Este proceso retroalimenta el repositorio de conocimiento con el conocimiento generado del desarrollo del proyecto software. Será el jefe de proyecto la persona encargada del proyecto software que retroalimentará dicha base de conocimiento. Los pasos de este proceso se recogen en la Figura 3-24.

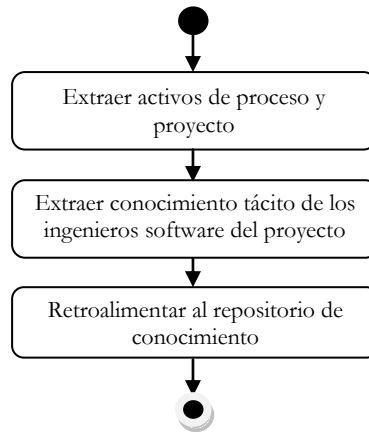


Figura 3-24 Proceso Retroalimentación del Repositorio de Conocimiento

Aportación

Este proceso enriquece el repositorio de conocimiento con los activos del proyecto y procesos de los proyectos que se desarrollen y con el conocimiento tácito de los ingenieros software de dichos proyectos para su posterior reutilización en proyectos software.

3.4 COMPONENTES Y ADAPTACIÓN DE LA ESTRATEGIA CORPORATIVA A LA SOLUCIÓN TECNOLÓGICA

3.4.1 Componentes de la solución tecnológica colaborativa

Los componentes desarrollados en la solución de esta tesis doctoral se muestran en la Figura 3-25. Como se puede observar en dicha figura, la arquitectura propuesta en esta tesis doctoral está formada por dos capas:

- **PIBOK-PB:** Process Improvement Based On Knowledge – Patterns Based. Esta capa está formada por tres módulos que van a gestionar la mejora de procesos a través de *Patrones de Producto* y un repositorio de activos de procesos (PIBOK-PB-PAL).

- **Collaborative-Platform:** Esta capa proporciona las funcionalidades de los entornos colaborativos.

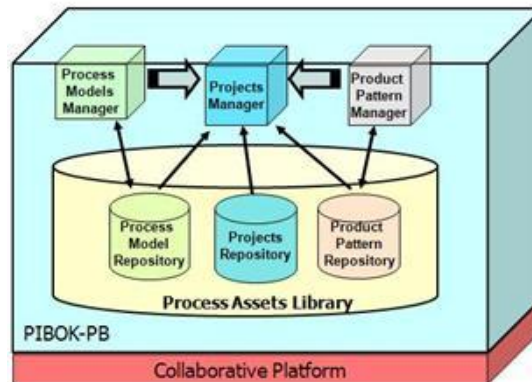


Figura 3-25 Componentes propios desarrollados

A continuación se describen los componentes propios desarrollados.

- **PIBOK-PB:** Process Improvement Based On Knowledge – Patterns Based. Esta capa está formada por tres módulos que van a gestionar: las mejores prácticas de la Ingeniería del Software, los patrones de producto y los proyectos software que se van a desarrollar; y un repositorio de activos de procesos (PIBOK-PB-PAL).

Módulos Gestores:

- Gestor de Modelos de Procesos Software: este componente se encarga de gestionar los modelos y técnicas de procesos software educidos de las organizaciones y de las mejores prácticas de la Ingeniería del Software. Este gestor sirve para seleccionar un modelo o técnica de proceso en función de parámetros de la organización y del proyecto que se va a desarrollar.

El rol encargado de gestionar los modelos y técnicas de procesos software es el jefe de proyecto.

Si los modelos o técnicas proporcionados por el sistema no se ajustan a las necesidades del proyecto, el jefe de proyecto puede adaptar dichos modelos y técnicas a la metodología y procesos de su organización con las técnicas y/o actividades que considere más oportunas, creando un modelo de proceso específico para su organización.

- Gestor de Patrones de Producto: este componente proporciona el conocimiento de los expertos sobre cómo se realiza cada una de las actividades del modelo de proceso creado en el componente Gestor de Modelos de Procesos Software. Para ello, el

sistema buscará e instanciará los *Patrones de Producto* para cada una de las actividades del proceso software.

El rol encargado de la búsqueda e instanciación de los *Patrones de Producto* es el jefe de proyecto.

La búsqueda de los *Patrones de Producto* se realiza con una regla que tiene que cumplir tres premisas: la actividad se realiza en un contexto determinado; cumple unas determinadas restricciones; y, coincide con el problema que se quiere resolver. El consecuente de la inferencia es: identificar el/los *Patrones de Producto* que tienen el conocimiento que permite al rol correspondiente realizar esa actividad.

If you find yourself in this **context**
(and) with this **problem**
(and) entailing these **forces**
then
map a product pattern in your project
(and) look for more product patterns

La instanciación de los *Patrones de Producto* es la elección de qué *Patrón de Producto* seleccionar si en la búsqueda se hubiera encontrado más de un *Patrón de Producto* para realizar una actividad. La elección la realiza el jefe de proyecto con la información que le proporciona cada uno de los *Patrones de Producto*.

- Gestor de Proyectos: este componente sirve para gestionar el proyecto. Para ello, se genera una guía electrónica de ejecución del proyecto y una guía electrónica de proceso.

La guía electrónica de ejecución del proyecto genera un diagrama Gantt con la planificación del proyecto, especificando por cada actividad del modelo los roles asociados a la tarea y el tiempo planificado para realizar dicha tarea. Además, se proporciona un workflow del proyecto, es decir, una lista de las tareas que se tienen que realizar, así como una breve descripción de las mismas. Por cada una de las actividades del workflow se generará una guía electrónica de proceso.

La guía electrónica de proceso, proporciona a cada actividad del modelo de proceso elegido en el proyecto, los *Patrones de Producto* instanciados en el módulo Gestor de

Patrones de Producto. El *Patrón de Producto* proporciona la información para poder realizar la actividad:

- qué entradas necesita para realizar la actividad (campo Entries),
- qué pasos tiene que realizar para la realización de la actividad (campo Solution)
- qué productos de salida va a obtener (campo Exit)
- con qué roles va a interactuar en dicha actividad (campo Roles)
- lecciones aprendidas de la realización de esta actividad con este patrón (campo Lessons Learned)
- plantilla que facilita la realización de la actividad (campo Templates)

Repositorio de Activos de Proceso (PIBOK-PB-PAL): el repositorio de Activos de Procesos está compuesto por cinco repositorios conceptuales y tres repositorios físicos. En la Figura 3-25 se muestra la relación entre los tres repositorios físicos y los tres módulos gestores. A continuación se describe el repositorio de activos de procesos (PIBOK-PB-PAL) representada mediante la wiki:

- Repositorio de Modelos de Procesos Estándar: este repositorio almacena los modelos de procesos y las actividades de las mejores prácticas de la Ingeniería del Software. Este módulo interactúa con un proceso que se realiza antes de la implantación de procesos en una organización que se denomina, “Educción de las Mejores Prácticas en la Ingeniería del Software”. Este proceso se encarga de extraer las mejores prácticas de estándares, libros, documentos técnicos y los almacena en este repositorio. El repositorio interactúa con dos módulos gestores y dos repositorios de datos:
 - Gestor de Modelos de Procesos Software: el repositorio proporciona los modelos y actividades almacenados en el repositorio que cumplan con los requisitos de las características de la organización y del proyecto. Es decir, proporciona todos aquellos modelos de procesos y, metodologías definidas como mejores prácticas en la Ingeniería del Software que se adapten al tipo de organización y proyecto que se está llevando a cabo.
 - Gestor de Proyectos: si el jefe de proyecto está de acuerdo en utilizar un modelo de proceso estándar, este repositorio proporciona el modelo seleccionado junto con sus actividades y toda la información relacionada.

- Repositorio de Modelos de Procesos Específicos: si el jefe de proyecto no está de acuerdo en utilizar un modelo de proceso o metodología estándar, este repositorio proporciona los modelos, metodologías y actividades estándar por si se quiere basar en ellas para la creación de su propio modelo de proceso.
- Repositorio de Proyectos: el “repositorio de Modelos de Procesos Estándar” proporciona al “repositorio de Proyectos” los datos relacionados con el modelo de proceso o metodología seleccionada por el jefe de proyecto.
- Repositorio de Modelos de Procesos Específicos: este repositorio almacena toda la información relacionada con los modelos de procesos propios definidos por las organizaciones (actividades, roles, etc). El repositorio interactúa con dos módulos y un repositorio de datos:
 - Gestor de Modelos de Procesos Software: este gestor permite que se creen las organizaciones sus propios modelos de proceso y metodologías que son almacenados en este repositorio.
 - Gestor de Proyectos: proporciona el modelo creado por el jefe de proyecto.
 - Repositorio de Proyectos: el repositorio de Modelos de Procesos Específicos proporciona los datos relacionados con el modelo de proceso creado por el jefe de proyecto para la realización del proyecto que está en curso.

Ambos repositorios conceptuales se representa mediante un mismo repositorio físico, ya que almacenan los mismos datos: los modelos de proceso, sus actividades, la información relativa a las actividades y, las reglas y parámetros mediante las cuales se infieren los modelos que mejor se adecuan a las características de las organizaciones y proyectos. El modelo entidad-relación de este repositorio se muestra en la Figura 3-26.

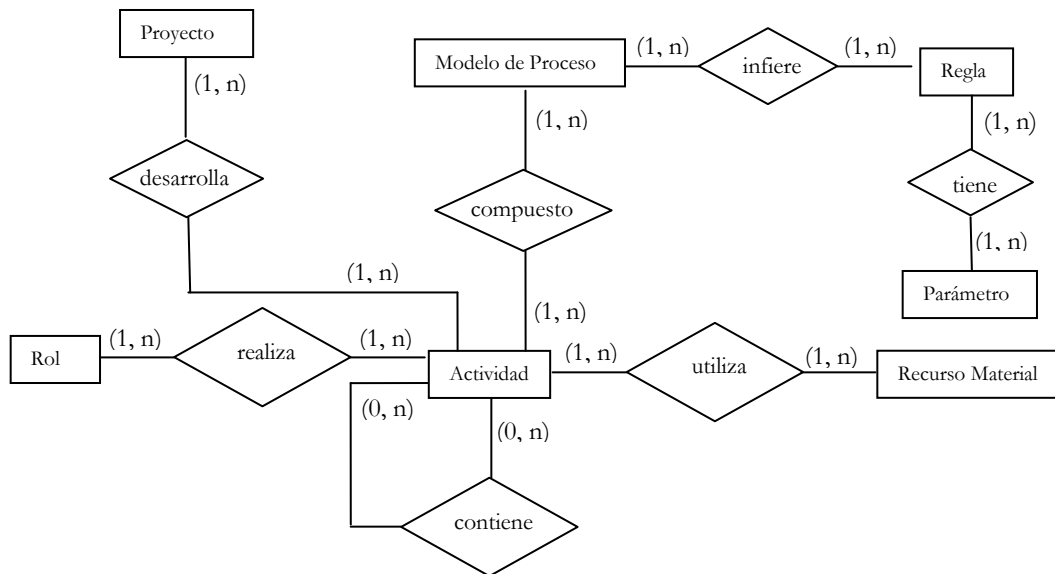


Figura 3-26 Modelo Entidad-Relación - Modelos de Procesos

- Repositorio de Patrones de Producto Estándar: en este repositorio se encuentran almacenados los patrones de producto educidos de las mejores prácticas de la Ingeniería del Software.

El repositorio interactúa con dos módulos y un repositorio de datos:

- Gestor de Patrones de Producto: este repositorio le proporciona los *patrones de producto* donde el módulo gestor pueda realizar el proceso de búsqueda de los *patrones de producto* para cada una de las actividades del modelo de proceso.
 - Gestor de Proyectos: en dicho módulo se lleva a cabo el proceso de instanciación de los *patrones de producto*, en el que se define para cada una de las actividades del modelo de proceso el/los *patrones de producto* que se van a ejecutar. Dicha información de los *patrones de producto* la obtiene de este repositorio.
 - Repositorio Proyectos: el repositorio de Patrones de Producto Estándar le proporciona al repositorio de Proyectos los *patrones de producto* instanciados para cada una de las actividades del proyecto en curso.
- Repositorio de Patrones de Producto Específicos: en este repositorio se encuentran almacenados los patrones de producto que se creen a partir de la educación del conocimiento de cómo se realizan cada una de las actividades en la organización.

El repositorio interactúa con dos módulos y un repositorio de datos:

- Gestor de Patrones de Producto: en el módulo gestor se realiza el proceso de búsqueda de *patrones de producto* para cada una de las actividades del modelo de proceso. Esta búsqueda, además de realizarse en el Repositorio de Patrones de Producto Estándar, también se realiza en el Repositorio de Patrones de Producto Específicos.
- Gestor de Proyectos: el repositorio le proporciona los *patrones de producto* que se van a instanciar en cada una de las actividades del proyecto para la ejecución de dicho proyecto.
- Repositorio Proyectos: el repositorio de Patrones de Producto Específicos le proporciona los *patrones de producto* instanciados para cada una de las actividades del proyecto en curso.

Los repositorios, Repositorio de Patrones de Producto Estándar y Repositorio de Patrones de Producto Específicos, almacenan toda la información relacionada con los patrones de producto. Ambos repositorios conceptuales se va a representar mediante un repositorio físico denominado Repositorio de Patrones de Producto, ver Figura 3-27. Este repositorio físico va a contener la siguiente información: el nombre del *patrón de producto*; el contexto inicial y final detallando las precondiciones del sistema para aplicar el patrón y una vez aplicado; las restricciones de la organización y del proyecto que son necesarios para poder llevar a cabo ese *patrón de producto*; los productos de entrada necesarios para poder aplicar el patrón y los productos de salida que se van a obtener una vez desarrollado dicho patrón; la solución del patrón, en el que se detalla el diagrama de actividades con todos los pasos que tiene que realizar cada uno de los roles para poder obtener los productos de salida. Se proporcionará también las plantillas en el caso de que hubiera. Las lecciones aprendidas; este campo del patrón se rellenará una vez realizado el patrón ya que se van a obtener dichas lecciones aprendidas una vez aplicado los patrones de producto a un proyecto determinado. Los recursos de información que se proporciona al usuario como un campo adicional, para en el caso de tener alguna duda en la ejecución de la solución, se tenga material adicional donde buscar y poder resolver esas dudas. El nivel de capacidad o madurez se le facilita al usuario para que sepa qué niveles de los modelos de procesos de referencias se obtiene una vez desarrollado ese *patrón de producto* en un proyecto. Además, en el modelo entidad-relación,

Figura 3-27, se puede observar cómo se guardan los patrones de producto relacionados, es decir, con qué otros patrones de producto se relaciona por diversos criterios, por ejemplo, pertenecer a la misma fase del ciclo de vida del desarrollo software, patrones necesarios para poder realizar éste, etc.

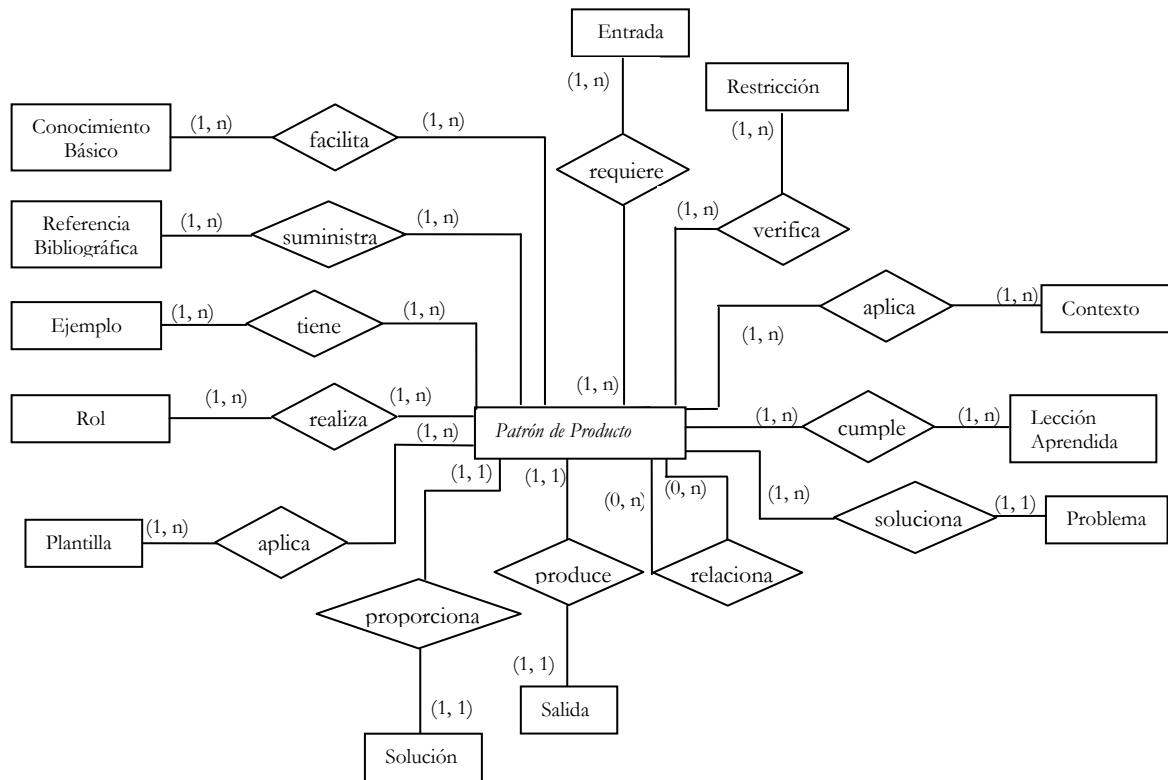


Figura 3-27 Modelo Entidad-Relación - Repositorio de Patrones de Producto

- Repositorio de Proyectos: este repositorio contiene la información de todos los proyectos realizados y en curso de una organización.

Los datos que se almacenan en este repositorio corresponden a tres grupos de datos que se pueden categorizar en:

- datos relacionados con el modelo de proceso seleccionado para realizar un proyecto. Se almacenarán los datos de la organización, sus proyectos, y toda la información relativa a los proyectos: modelos de procesos y/o metodología aplicada al mismo, actividades desarrolladas, recursos y costes.
- datos de los *patrones de producto* instanciados de cada una de las actividades de dicho modelo de proceso. Se almacenarán los *patrones de producto* instanciados para cada una de las actividades, es decir, qué *patrones de producto* se han

utilizado en cada actividad y toda la información relacionada con dichos patrones pero con los datos de la actividad.

- datos correspondientes de la ejecución de cada una de esas actividades. Se almacenarán todos los datos correspondientes de la ejecución de cada una de las actividades con cada *patrón de producto*, la guía electrónica de ejecución del proyecto y la guía de proceso de las actividades del proyecto.

El repositorio interactúa con el módulo:

- Gestor de Proyectos: es el módulo que realiza la ejecución del proyecto. El repositorio le proporciona a este módulo: el modelo de proceso con las actividades, los *patrones de producto* correspondientes a cada actividad y la información requerida por la guía electrónica de ejecución del proyecto y la guía electrónica de proceso.

El modelo entidad-relación del repositorio de Proyectos se muestra en la Figura 3-28.

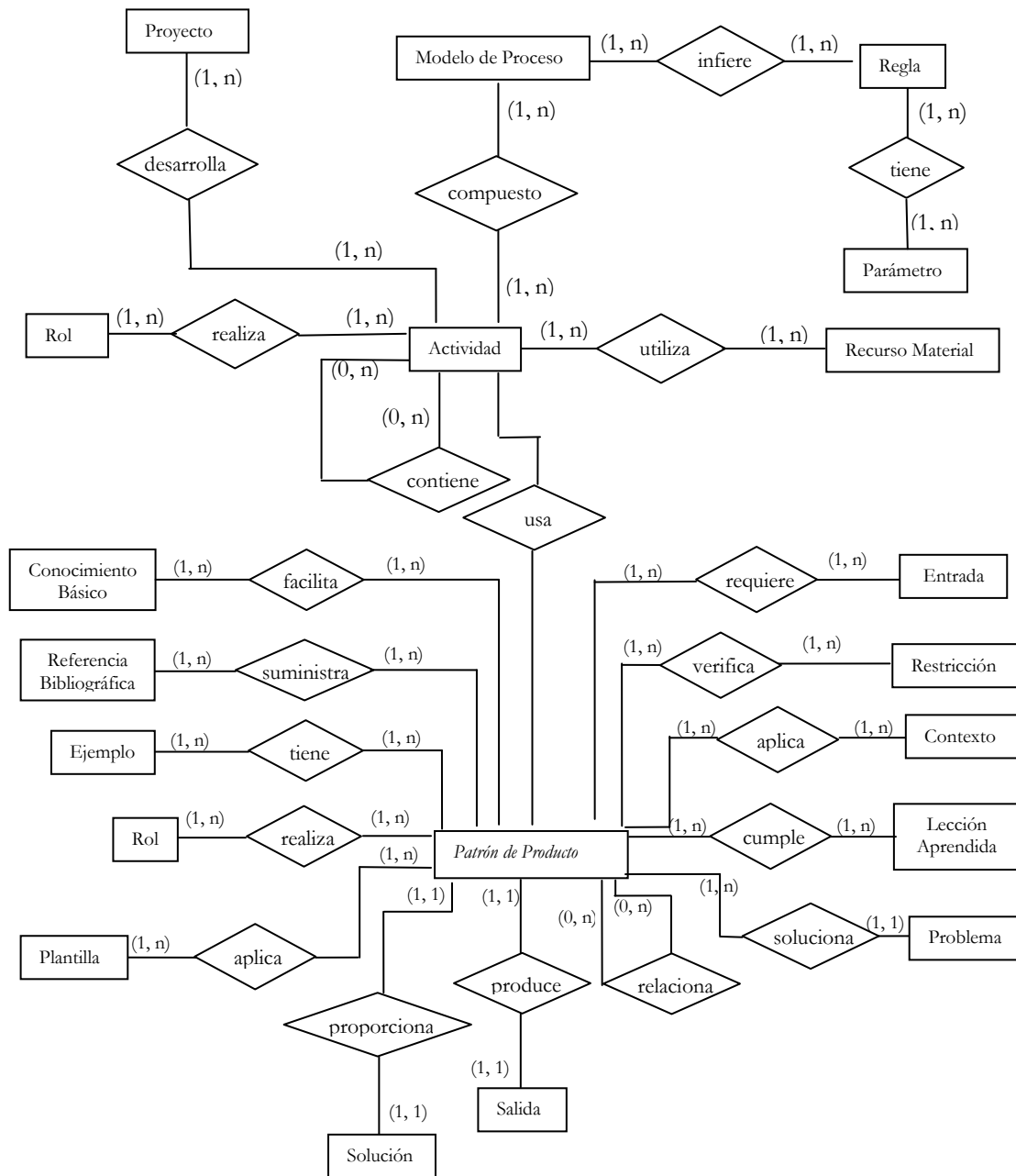


Figura 3-28 Modelo Entidad-Relación - Repositorio de Proyectos

- **COLLABORATIVE PLATFORM:** Esta capa proporciona las funcionalidades de los entornos colaborativos. La plataforma colaborativa proporciona las funcionalidades y características colaborativas a la solución propuesta. Dichas características son:

- Proveer de una buena comunicación a los diferentes roles de las organizaciones.
- Coordinar el trabajo entre los miembros del equipo en tiempo y espacio diferente.

- Integrar información entre los diferentes sistemas, usando opciones de desarrollo flexibles y herramientas de gestión, todo ello en una única solución.
- Añadir funcionalidad a través de desarrollos de componentes que sean integrables por dicha solución
- Proporcionar un interfaz usable y familiar.
- Ser independiente del almacenaje de los datos de configuración, datos personales y datos del portal colaborativo que se vaya a generar.
- Permitir realizar búsquedas avanzadas

Una característica importante en la solución propuesta es la búsqueda de información. La solución propuesta se basa en artefactos de conocimiento reutilizables, patrones de producto, almacenados en repositorio de datos, anteriormente explicados. Para que dichos artefactos sean reutilizables de manera eficiente, se debe proporcionar un motor de búsqueda que busque tanto en datos estructurados, como es el repositorio de activos de proceso, como en datos no estructurados como son los ficheros anexados a los patrones de producto. Es por ello, que después de un estudio exhaustivo de los entornos colaborativos que proporcionan además de las características antes mencionadas, un motor de búsqueda eficiente, se ha elegido la plataforma colaborativa Microsoft Office Sharepoint Services, cuyo motor de búsqueda es Microsoft Enterprise Search.

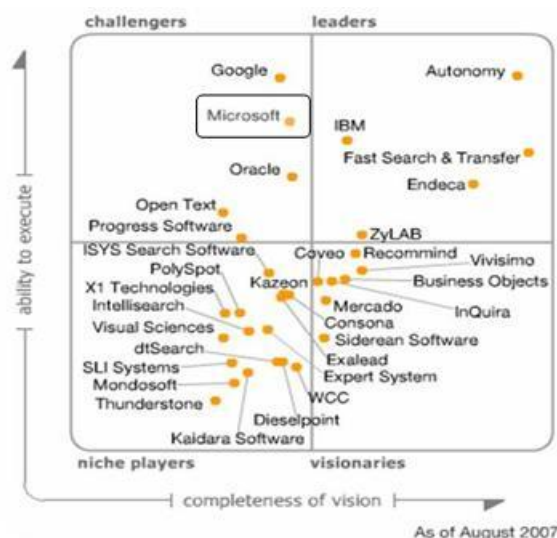


Figura 3-29 Motores de búsqueda

En la Figura 3-29, se muestra como el motor de búsqueda de Microsoft, se ha situado entre los motores de búsqueda más importantes como son Google, Oracle e IBM. Una de las características importante de Microsoft Enterprise Search, es que proporciona resultados relevantes de datos estructurados y no estructurados. El interfaz para realizar la búsqueda es simple y los resultados de las búsquedas son dinámicos. Algunas de las características de dicho motor de búsqueda son:

- Búsqueda por palabras clave
- Búsqueda dentro de los resultados devueltos
- Interfaces estándares y avanzadas
- Resalta lo más significativo
- Wildcard search: caracteres que sustituyen otros caracteres en las búsquedas, son búsquedas con comodín, por ejemplo: “?”, “*”.
- Definir la búsqueda por sitios web o por repositorio
- Categorizar los resultados por repositorio
- Ordenar resultados por campos seleccionados por el usuario
- Mostrar resultados de URL, autor, tamaño de archivo, fecha de creación, fecha modificada, y otros meta-datos
- Algoritmos avanzados para mejorar la relevancia
- Operadores de búsqueda lógicos
- Colapsar duplicados
- Indexar cientos de tipos de archivos
- Búsqueda explícita por mayúscula/minúscula
- Personalización de la búsqueda UI para el desktop y el servidor
- Informes de las búsquedas y optimización del administrador
- Almacenar una búsqueda para el futuro (Alertas, RSS)
- Alertas de correo con enlaces al conjunto de resultados actualizados
- Resultados devueltos por errores tipográficos (“Querías decir...”)
- Búsqueda a través de meta-datos

3.4.2 Adaptación de la estrategia corporativa genérica a la solución tecnológica propuesta.

En esta tesis doctoral se ha definido una estrategia corporativa para el uso del marco

metodológico definido, que combina la especificación de un conjunto de reglas colaborativas con mecanismos de reutilización, teniendo en cuenta los diferentes tipos de equipos de trabajo que actualmente se están implantando en las organizaciones. Esta estrategia es independiente de los procesos software, del tipo de organización o característica del proyecto.

La estrategia se ha diseñado para los roles del jefe de proyecto y para el equipo de desarrollo, donde está incluido los roles de analista y diseñador.

La estrategia corporativa adaptada para el jefe de proyecto puede comenzar de dos maneras (Figura 3-30):

1. Si el jefe de proyecto desea que el sistema le ayude a elegir el modelo de proceso software que mejor se adapte a las necesidades de la organización y de la empresa, ejecutará el gestor de Modelos de Proceso y el gestor de Patrones de Producto para que el sistema le proporcione los procesos y los patrones de producto que mejor se adapte a su proyecto. Una vez que el jefe de proyecto tenga las actividades del modelo de proceso y los patrones de producto de aquellas tareas que él desee, el sistema generará un diagrama Gantt con la precedencia en la ejecución de las actividades del proyecto. El jefe de proyecto tendrá que insertar en el diagrama Gantt la información de duración, recursos y costes que requiera el proyecto.
2. Si el jefe de proyecto no desea que el sistema le ayude a elegir el modelo de proceso, el jefe de proyecto podrá planificar su proyecto software mediante un diagrama Gantt. Cuando realice dicha planificación, podrá buscar e instanciar los patrones de producto de las actividades del proyecto, gracias al catálogo de patrones de producto representado mediante la wiki.

A partir de este paso, el jefe de proyecto exportará los datos a una plataforma colaborativa, para que los miembros del equipo de desarrollo puedan realizar las actividades del proyecto de forma colaborativa, si así lo requiere la actividad.



Figura 3-31 Adaptación de la Estrategia Corporativa a la solución tecnológica propuesta

A continuación, se realiza la descripción de los pasos de la estrategia adaptada a la solución tecnológica propuesta en esta tesis doctoral junto a un caso práctico que fue desarrollado por la empresa Sias en la validación de esta tesis doctoral en el proyecto software denominado “School Management System”.

- La estrategia corporativa adaptada para el jefe de proyecto es la siguiente:

1.- Buscar y Seleccionar Modelo de Proceso. El jefe de proyecto realizará la búsqueda de los modelos de proceso y metodologías que mejor se adapten a las características de la organización y del proyecto. El sistema mostrará los resultados de la búsqueda y el jefe de proyecto en función de su experiencia y de la información que le muestra en los resultados de dicha búsqueda seleccionará el modelo de proceso o metodología para desarrollar el proyecto. Una vez seleccionado el modelo de proceso o la metodología, el jefe de proyecto podrá adaptar el árbol de procesos proporcionado a las necesidades de su proyecto, añadiendo, modificando o eliminando actividades.

“School Management System”, SIAS company

El jefe de proyecto proporcionó las siguientes características:

- de la organización: tamaño de la organización menos de 50 empleados, no tiene equipos de desarrollo distribuidos geográficamente,*
- del proyecto: proyecto de gestión; paradigma orientado a objetos; equipos de desarrollo 3 personas; experiencia del equipo de desarrollo 5 años; sin experiencia en metodologías y modelos de procesos.*

El sistema en función de estas características, seleccionó la metodología Craig Larman y eXtreme Programming como las dos metodologías que más se adecuaban a dichas características. El jefe de proyecto, en función de la información proporcionada por el sistema de cada metodología seleccionó la metodología Craig Larman para el desarrollo del proyecto. Una vez seleccionada, el jefe de proyecto pudo personalizar las actividades y procesos de dicha metodología a las necesidades de su proyecto. En este caso, el jefe de proyecto añadió las actividades de gestión de proyectos software, estimación, planificación y seguimiento al árbol de procesos de la metodología.

2.- Buscar e Instanciar Patrones de Producto. El jefe de proyecto podrá buscar e instanciar los patrones de producto de los procesos y actividades del árbol de procesos de la metodología o modelo de proceso seleccionado.

“School Management System”, SIAS company

El jefe de proyecto hizo clic en el botón de búsqueda de patrones de producto y se realizó la búsqueda de los patrones de producto del árbol de procesos seleccionado y modificado por el jefe de proyecto. Para una actividad, Diagrama de Casos de Uso, el sistema encontró varios patrones de producto que se podrían utilizar para la realización de la misma. El jefe de proyecto seleccionó un patrón de producto en función a la información que le proporcionaba el sistema, en concreto, el campo solution que le proporcionaba los pasos que tenía que realizar para obtener dicho producto y los campos templates y examples.

3.- Planificar el proyecto. El jefe de proyecto planificará el proyecto, definiendo las tareas a desarrollar, el tiempo de desarrollo de cada actividad así como asignará los roles que desarrollarán cada actividad.

A este paso, el jefe de proyecto puede llegar por dos caminos:

- Una vez buscado y seleccionado el modelo de proceso o metodología y los patrones de producto que ayuden a la realización de la misma. En este caso, en esta actividad el jefe de proyecto simplemente planificará el tiempo de desarrollo de las actividades del árbol de proceso y asignará los roles a cada una de ellas.
- Sin haber buscado y seleccionado el modelo de proceso o metodología y los patrones de producto. En este caso, el sistema mediante la ayuda de un software de Gestión de proyecto, por ejemplo Project 2003, planificará el proyecto con una técnica de planificación, por ejemplo el Diagrama Gantt. Para realizar dicha planificación se le proporcionará al jefe de proyecto un esquema de qué información relevante de la organización y del proyecto tiene que incluir en dicha planificación. Una vez realizada dicha planificación, el jefe de proyecto podrá realizar la búsqueda e instanciación de los patrones de producto de dichas actividades. Para ello, se ha desarrollado un plug-in en Visual Basic 2006, que añade la funcionalidad de buscar e instanciar los patrones de producto para obtener los productos software de las actividades. Para realizar dicha búsqueda el sistema necesita información del contexto y restricciones (forces) de la organización, proyecto y actividad de la que se quiere buscar los patrones de producto. El sistema proporcionará los patrones de

producto que se adecuen a dichas características y el jefe de proyecto instanciará uno de ellos para la realización de la actividad. Esta búsqueda se realiza en el repositorio de conocimiento de activos de procesos y proyectos. Y el resultado se muestra en la planificación del proyecto, diagrama Gantt, al que se le añade una columna con la dirección url del *patrón de producto* instanciado para cada una de las actividades.

“School Management System”, SIAS company

En este proyecto, al haber seleccionado, mediante los gestores de Modelos de Procesos y Patrones de Producto, la metodología Craig Larman y los patrones de producto de dichas actividades, el jefe de proyecto únicamente planificó el proyecto, definiendo para dichas actividades, el tiempo y los roles asignados. Para ello se utilizó la herramienta de gestión de proyecto “Microsoft Project 2007”, y la técnica de planificación de proyecto “Diagrama Gantt”.

4.- Importar la planificación del proyecto a la herramienta colaborativa. El jefe de proyecto importará la planificación del proyecto a la herramienta colaborativa donde el proyecto es desarrollado. Con estos datos, el equipo de desarrollo podrá trabajar colaborativamente en las tareas y actividades donde la colaboración es requerida.

“School Management System”, SIAS company

Microsoft Office Sharepoint Portal Server (MOSS) fue la plataforma colaborativa. Para importar la planificación del proyecto, se desarrolló un webpart en C# en Visual Studio 2005. Este webpart importa los datos de la planificación del proyecto a una lista personalizada con vista del tipo diagrama Gantt. De esta forma, todo el mundo involucrado en el proyecto puede ver la planificación del mismo.

5.- Modificar la planificación del proyecto. El jefe de proyecto podrá modificar la planificación del proyecto en la herramienta colaborativa. En el caso que alguna modificación se realice en la planificación, se la herramienta colaborativa enviará una alerta para notificar al equipo de desarrollo dichos cambios.

“School Management System”, SIAS company

El jefe de proyecto planificó el proyecto dos veces debido a que la actividad “Requirement specification” en la fase de análisis y la actividad “Class Diagram” en la fase de diseño se retrasaron. Aunque la planificación del proyecto se modificó dos veces, ambas versiones de la planificación fueron almacenadas en el portal web colaborativo. De esta forma, el jefe de proyecto pudo comprobar la desviación en el tiempo y coste de la planificación inicial del proyecto. En ambas modificaciones, el jefe de proyecto notificó mediante alertas los cambios en la planificación. Algunos miembros del equipo pidieron una reunión virtual para aclarar algunas dudas de la nueva planificación. Además, los comentarios que el jefe de proyecto realice en la reunión virtual se publicaron en el tablón de anuncios de Preguntas Frecuentes (FAQ).

6.- Participar en la ejecución del proyecto. El jefe de proyecto participará en la ejecución del proyecto: accederá y participará a las reuniones virtuales, responderá los comentarios y dudas en los foros de discusión, hará comentarios en los tableros de anuncios, comprobará el estado de las actividades, se comunicará con el equipo de desarrollo, compartirá su conocimiento, etc. Debido a que el jefe de proyecto puede conectarse al portal web colaborativo en cualquier momento podrá participar en la ejecución del proyecto de forma rápido y más efectiva permitiendo compartir el conocimiento entre el equipo de desarrollo, tanto conocimiento explícito (activos del proyecto) como el conocimiento tácito (know-how de las actividades y procesos del proyecto).

“School Management System”, SIAS company

El jefe de proyecto participó en todo momento en la ejecución del proyecto. En la planificación de este proyecto, el equipo de desarrollo tenía hitos donde los productos de salida eran entregables de las actividades. El jefe de proyecto, revisó dichos entregables y notificó al equipo de desarrollo de los cambios que tenían que hacer en dichos productos. Para ello, el jefe de proyecto almacenó los informes de los cambios en una biblioteca de documentos “changes”, y notificó al equipo de desarrollo de los cambios para su rework. Además, el jefe de proyecto aclaró las dudas a través de los foros de discusiones y participó en reuniones virtuales para gestionar esas actividades.

En este sentido, el jefe de proyecto nos hizo llegar su satisfacción ya que esta solución le permitía realizar la gestión y ejecución del proyecto en el mismo portal web colaborativo, mejorando la comunicación con el equipo de desarrollo reduciendo el rework y el tiempo para el desarrollo de los productos software.

7.- Seguimiento del proyecto. El jefe de proyecto accederá a las actividades y activos software para comprobar el estado de dichas actividades y poder realizar el seguimiento del proyecto. En el caso de haber retrasos en las actividades, el jefe de proyecto podrá comprobar las razones que lo causa e intentar resolverlas o en el peor de los casos tendrá que replanificar el proyecto.

“School Management System”, SIAS company

El jefe de proyecto realice el seguimiento del proyecto a través del portal web colaborativo. El accedió al espacio de trabajo compartido y comprobó el estado de las actividades (no comenzada, comenzada, en progreso, terminada). En este proyecto, hubo dos retrasos debido a que el cliente retrasó la reunión de captura de requisitos y posteriormente porque un diseñador se puso enfermo. Esos retrasos se comunicaron vía email y a través del tablón de anuncio al jefe de proyecto quien decidió en ambos casos volver a planificar el proyecto teniendo en cuenta ambas circunstancias.

8.- Retroalimentar los activos de proceso y proyecto. El jefe de proyecto podrá modificar el repositorio de conocimiento con el conocimiento explícito y tácito del desarrollo del proyecto, de esa forma se enriquece el repositorio de conocimiento con la experiencia de los expertos para su posterior reutilización. Únicamente el jefe de proyecto podrá modificar la wiki, de esa forma, se garantiza que los activos publicados han sido verificados y validados por él. La reutilización de este conocimiento tendrá un impacto positivo en la productividad de las organizaciones y reducirá el tiempo y coste del proyecto software ya que el equipo de desarrollo podrá obtener el conocimiento necesario para desarrollar las actividades del proyecto con el conocimiento del repositorio. Esto elimina la necesidad de buscar el conocimiento en libros y recursos electrónicos externos, con su ahorro en tiempo y mejora en la productividad de los proyectos software.

“School Management System”, SIAS company

Los datos del proyecto almacenados en las bibliotecas de documentos del portal web colaborativo se actualizaron en el repositorio de la wiki. En la wiki, se creó en su estructura un proyecto nuevo con todo su conocimiento, conocimiento explícito a través de los productos obtenidos en el desarrollo del mismo, y el conocimiento tácito que se almacenó en el campo de Lecciones aprendidas. Esta retroalimentación fue valorada positivamente ya que el conocimiento almacenado del proyecto desarrollado pudo utilizarla en otros proyectos software.

- La estrategia corporativa adaptada para el equipo de desarrollo es la siguiente:

1.- Visualizar la planificación del proyecto. Cada miembro del equipo podrá acceder a la planificación del proyecto, a las actividades asignadas a cada uno de ellos así como podrán ver la asignación de tiempo y roles para el desarrollo de dichas actividades.

“School Management System”, SIAS company

Todos los miembros del equipo accedieron al portal web colaborativo mediante un login y una password. Una vez dentro de dicho portal, los miembros del equipo pudieron ver la planificación total del proyecto e información general del proyecto a través de tableros de anuncios. Esto fue valorado positivamente por el equipo de desarrollo ya que les proporcionaba una visión global del proyecto.

2.- Acceder al área de Trabajo Compartida. Cada miembro del equipo de desarrollo podrá acceder a sus actividades en un área de trabajo compartido asignado para el desarrollo de dicha actividades. En dicho área de trabajo, los miembros del equipo podrán utilizar todas las funcionalidades colaborativas para desarrollar la actividad.

2.1.- Añadir comentarios y dudas en el foro de discusión. En el área de trabajo compartido el equipo podrá añadir comentarios y dudas que podrán compartir con el resto de los miembros del equipo asignados a la misma actividad.

2.2.- Programar Reuniones Virtuales. El equipo de trabajo podrá programar reuniones virtuales donde se tomarán decisiones sobre la actividad a desarrollar, así como podrán discutir las dudas y comentarios expresados en los foros de discusión.

“School Management System”, SIAS company

Cada miembro del equipo accedió a sus áreas de trabajo compartido para realizar y desarrollar sus actividades.

3.- Verificar los productos de entrada para desarrollar las actividades. Los miembros del equipo tendrán que comprobar si la actividad se podrá desarrollar por orden de precedencia, y si todos los productos de entrada están disponibles para poder ejecutar la actividad. Si los productos de entrada no están disponibles, el rol encargado de la actividad notificará que dicha actividad está en estado “en Espera” esperando los productos de entrada que necesita para desarrollarse.

“School Management System”, SIAS company

El equipo de desarrollo comprobó que todos los productos de entrada estaban disponibles. Por ejemplo, para la actividad “Use Case Diagram”, el producto de entrada es la especificación de requisitos. Por lo que si dicha especificación no se encuentra disponible, se notificará que dicha actividad se encuentra en estado “en espera” hasta que dicho producto de entrada se proporcione.

4.- Desarrollar las actividades. El equipo desarrollará la actividad colaborativamente utilizando las funcionalidades que proporciona la herramienta colaborativa. Esta solución colaborativa proporciona el conocimiento de los expertos a través de los *patrones de producto*, que se proporciona en el momento del desarrollo de la actividad a través del repositorio de conocimiento.

“School Management System”, SIAS company

Cada miembro del equipo desarrollo sus actividades asignadas. Antes de empezar con sus actividades, el equipo de desarrollo tuvo una reunión virtual para coordinar el trabajo que tenían que realizar en dicha actividad. En este sentido, el

patrón de producto asociado a cada actividad fue utilizado como una guía para entender y desarrollar los productos de la actividad esperados. Para ello, ellos recuperaron los productos de entrada de una biblioteca de documentos “Entradas” y desarrollaban los productos de salida también en bibliotecas de documentos “Salidas” utilizando específicas herramientas en cada actividad. Por ejemplo, Microsoft Word fue utilizado para la captura de requisitos. Los miembros asignados a una actividad trabajaron colaborativamente on-line y offline para desarrollar el producto software. En esta actividad, especificación de requisitos, se utilizó para desarrollarla un foro de discusión y dos reuniones virtuales con el cliente. El estado de dicha actividad se modificaba cuando correspondía y si algún problema ocurría se notificó al jefe de proyecto. Por los comentarios de los miembros del equipo, las funcionalidades más utilizadas en este proyecto fueron los foros de discusión y las características colaborativas del editor de documento (control de versiones y aprobación y rechazo del documento).

5.- Acceder a los activos de procesos y proyectos. El equipo de desarrollo puede en todo momento recuperar los activos de procesos y proyectos del repositorio de conocimiento. Dicho repositorio proporciona el conocimiento tácito de las actividades, ejemplos de proyectos desarrollados y plantillas que ayudan al desarrollo de la actividad.

“School Management System”, SIAS company

El equipo de desarrollo recuperó ejemplos de previos activos de procesos y proyectos de la wiki para desarrollar sus actividades. La información más valorada por el equipo de desarrollo fueron: solución, plantillas y ejemplos. Mientras desarrollaban sus actividades, ellos accedían a los patrones de producto al principio con más frecuencia ya que de utilizar los mismos pasos una y otra vez, a medida que accedían al mismo patrón se les hacía fácil de aprender. De media, en este proyecto los patrones de producto fueron accedidos de tres a cinco veces.

6.- Actualizar estado del proyecto. Una vez desarrollada cada actividad, el estado de la actividad y el tanto por ciento de realización de la actividad se modifica para que el jefe de

proyecto valide la actividad y la apruebe. Cuando el jefe de proyecto apruebe la actividad, los datos de la actividad son públicos al resto de los miembros del equipo.

“School Management System”, SIAS company

El equipo de desarrollo modificó el estado de las actividades una vez que estas fueron desarrolladas para que el jefe de proyecto aprobara y pusieran públicos los datos de la misma. Es así como una actividad se da por terminada.

3.4.3 Arquitecturas de Despliegue

La arquitectura descrita puede ser desplegada en diferentes escenarios que se detallan a continuación:

- Si una organización quiere desplegar la arquitectura antes descrita y no tiene los suficientes recursos económicos para comprar la licencia de Microsoft Office Sharepoint Server (MOSS), puede optar por instalar en sus organizaciones Microsoft Windows SharePoint Services 3.0 (WSS 3.0), que es gratis para las organizaciones que usan sistemas operativos Microsoft Windows Server 2003, y que proporciona los servicios básicos de Sharepoint. La PAL, donde se almacenan los repositorios de las organizaciones, se almacenarán en los servidores del grupo Software Engineering Lab (SEL-UC3M) de la Universidad Carlos III de Madrid, donde esta tesis doctoral se está realizando. Esta arquitectura de despliegue se muestra en la Figura 3-32.

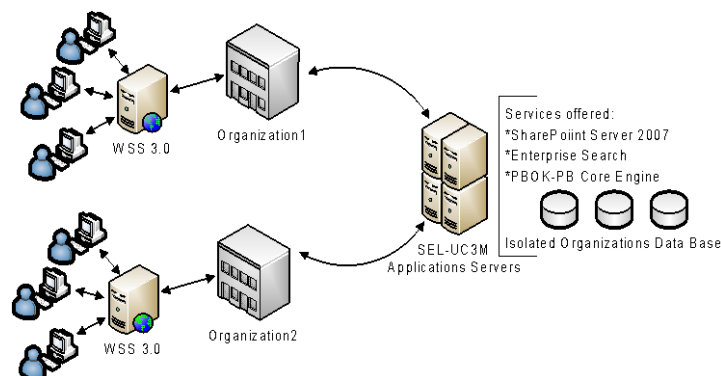


Figura 3-32 Opción 1 de Modelo de Arquitectura de Despliegue.

- Si una organización desea tener la solución colaborativa completa en sus oficinas y no tiene problemas a la hora de invertir en licencias, la organización tendría que instalarse Microsoft Office SharePoint Server 2007 (MOSS). En este caso, la PAL estaría instalada en la organización teniendo total control de ella. Esta opción incluye además Microsoft Enterprise Search. El modelo de la arquitectura de despliegue se puede ver en la Figura 3-33.

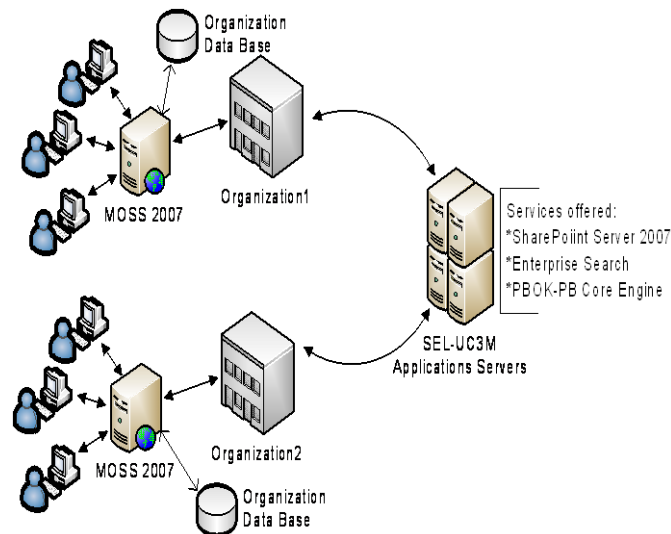


Figura 3-33 Opción 2 de Modelo de Arquitectura de Despliegue.

La arquitectura de PIBOK-PB está diseñada para permitir a las organizaciones compartir datos entre ellos así como permitir la comunicación usando mensajes XML.

Además, la integración de nuevas características de cada organización es posible gracias a la extensión de las capacidades que nos permite la plataforma colaborativa.

4. VALIDACIÓN

4. VALIDACIÓN	144
4.1 Introducción.....	149
4.2 Planificación de la Validación Experimental.....	150
4.3 Ejecución de la Validación Experimental	164
4.4 Análisis de los datos obtenidos en la Validación Experimental	171

La validación experimental se ha realizado de forma controlada y se ha documentado siguiendo el estudio realizado por (Jedlitschka et al, 2008) para que sea un proceso repetible. En este capítulo se detallan las siguientes secciones de la validación experimental:

- Introducción.
- Planificación de la Validación Experimental.
- Ejecución de la Validación Experimental.
- Análisis de los datos obtenidos en la Validación Experimental.

4.1 INTRODUCCIÓN

El objetivo de la validación experimental es comprobar si el uso del Marco Metodológico propuesto en esta tesis doctoral, mejora la eficiencia de uso de los procesos software, con el fin último de facilitar la difusión, uso y reutilización de activos de procesos y proyectos, utilizando como elemento encapsulador de conocimiento, el artefacto definido en la presente tesis, *patrón de producto*, y utilizando la estrategia corporativa en una plataforma colaborativa.

A continuación se describen los objetivos e hipótesis de la tesis doctoral.

Objetivos de la Investigación:

- **Objetivo 1:** Definir un artefacto que permita representar la experiencia y el conocimiento de los ingenieros software expertos sobre cómo hacer un producto software, basándose en la reutilización de las mejores prácticas de la ingeniería del software y del conocimiento tácito de los ingenieros expertos para obtener un producto software específico, con el fin de incidir en la resolución de la adquisición, representación, uso y, reutilización del conocimiento.
- **Objetivo 2:** Definir un marco metodológico que proporcione una estrategia corporativa que describa los procesos para realizar la difusión, uso y reutilización de los activos de proceso en una organización, con el fin de incidir en la resolución de la accesibilidad y transferencia del conocimiento.

- **Objetivo 3:** Mejorar la eficiencia de uso de los procesos software reduciendo el tiempo de desarrollo de los productos software (Hornbæk, 2006), con el fin de incidir en la resolución de la mejora de la usabilidad de procesos software.
- **Objetivo 4:** Mejorar parámetros que inciden en la calidad de los productos software a desarrollar, con el fin de incidir en la resolución de la mejora de la usabilidad de procesos software.

Hipótesis de la Investigación:

- **Hipótesis H1.1:** Es posible mejorar la eficiencia de uso de los procesos software y la calidad de los productos obtenidos utilizando el concepto de *patrón de producto* como elemento de reutilización de procesos software y encapsulador del conocimiento. Esta hipótesis cubre los objetivos 1, 3 y 4 de esta tesis doctoral.
- **Hipótesis H1.2:** Es posible mejorar la productividad (eficiencia de uso) de los proyectos software, utilizando el marco metodológico definido en esta tesis doctoral. Esta hipótesis cubre los objetivos 2, 3 y 4 de esta tesis doctoral.

4.2 PLANIFICACIÓN DE LA VALIDACIÓN EXPERIMENTAL

El método utilizado en la validación experimental es el método empírico: se han definido, planificado y realizado experimentos; se han recogido datos a través de cuestionarios y hojas de seguimiento; posteriormente se han analizado e interpretado los datos mediante técnicas estadísticas; y por último se han extraído conclusiones

4.2.1 Fases de la validación experimental

Con el objetivo de poder validar independientemente los elementos definidos en esta tesis doctoral (*patrón de producto* y marco metodológico) y poder observar cómo afectan la utilización de dichos elementos en el desarrollo de proyectos software, la validación experimental se ha realizado en tres fases independientes.

Fase I: Realización de los proyectos software sin el uso de los Patrones de Producto ni del Marco Metodológico.

El objetivo de la Fase I es el desarrollo de diez proyectos software sin la utilización de los *patrones de producto* ni del marco metodológico, únicamente con el conocimiento y la experiencia del jefe de proyecto y su equipo de trabajo.

Las características de la Fase I de la validación experimental son las siguientes:

- **Respecto a los Proyectos.** Se desarrollarán diez proyectos software de similares características, proyectos de gestión, con un tiempo estimado de realización de seis meses de duración.
- **Respecto al Equipo de Trabajo.** En esta fase participarán treinta ingenieros software en equipos de desarrollo de tres ingenieros software por proyecto. El perfil de todos los ingenieros software será de dos a cuatro años de experiencia. Los participantes involucrados en el desarrollo de los proyectos no tendrán conocimiento previo de los proyectos software a desarrollar.
- **Respecto a la Metodología.** Los proyectos software se desarrollarán centrándose en los procesos de las fases de análisis y diseño de la metodología propuesta por Craig Larman y el proceso de Gestión de proyecto.
- **Respecto a la Documentación.** A los equipos de trabajo únicamente se les proporcionará la planificación del proyecto, no se les entregará ningún tipo de documentación adicional al proyecto.
- **Respecto a la Formación.** Los equipos de desarrollo no recibirán formación en esta fase de validación, los proyectos software se desarrollarán únicamente con el conocimiento que dichos equipos de desarrollo tengan hasta el momento.

Durante la realización de esta fase, los ingenieros software involucrados en los proyectos software rellenarán la “Hoja de Seguimiento 1” (Tabla 4-1) y el “Cuestionario 1” (Tabla 4-2). La “Hoja de Seguimiento 1” se les proporcionará a los desarrolladores de los procesos de análisis, diseño y gestión sin el uso de los *patrones de producto*. En esta hoja de seguimiento se les hace preguntas relativas al número de horas que han tardado en realizar las

actividades desarrolladas, el grado de conocimiento para realizar dicha actividad, si han necesitado colaborar con otros miembros del equipo y las vías que han utilizado para ello.

Hoja de seguimiento 1 – Fase I Validación Experimental		Valoración
1	Proyecto	
2	Actividad	
3	Rol	
4	¿Ha utilizado el patrón de producto?	
5	Fecha comienzo	
6	Número de horas de estudio de cómo se realiza la actividad	
	Número de horas que ha tardado en realizar la actividad	
7	Prerrequisitos de la Actividad (Productos necesarios para realizar esta actividad)	
8	Documentos o producto/s de salida	
9	¿Qué grado de conocimiento tiene de la actividad (1-10)?	
	¿Has utilizado algún documento bibliográfico para realizar esta actividad?	
	¿Cuál/es?	
10	¿Has necesitado colaborar con algún miembro del equipo?	
	¿Con qué otros roles ha colaborado?	
	Número de horas que has tardado en ponerte en contacto con el/los rol/es?	
	Número de horas de interacción con el rol	
	¿Cómo se ha realizado dicha colaboración (email, reuniones, herramienta, etc)?	
	¿Por qué necesitaba la interacción? (Compartir documento, prerrequisito, etc)	
11	Grado de dificultad del proyecto (1-10)	
12	La estimación de la planificación ha sido correcta (1-10)	
13	Comentario sobre la realización de la actividad	

Tabla 4-1 Hoja de Seguimiento 1 – Fase I Validación Experimental

En el “Cuestionario 1”, el equipo de desarrollo principalmente tiene que valorar el grado de conocimiento que poseen de la actividad a realizar y si valorarían el tener conocimiento tácito de proyectos previos.

Cuestionario 1 - Fase I Validación Experimental		Valoración
1	Valore qué grado de conocimiento tiene para la realización de sus actividades. 1: muy poco 2: poco; 3: normal 4: mucho 5:experto	
	Comentario	
2	Considera que tener información/conocimiento de expertos le facilitaría la realización de sus actividades. 1: muy poco 2: poco; 3: normal 4: mucho 5: indispensable	
	Comentario	
3	Cuando tiene dudas a la hora de realizar alguna actividad del proyecto, busco información/conocimiento de expertos en la materia. Comente por orden de preferencia la fuente de búsqueda: 1: 0-20%; 2: 20%-40%; 3: 40%-60%; 4: 60%-80%; 5: 80%-100%.	
	Manuales	
	Libros	
	Internet	
	Otros recursos	
	Comentario	
4	Los manuales que proporciona la empresa son accesibles. 1: nada 2: muy poco; 3: algo 4: bastante 5: mucho	
	Comentario	
5	Ayudaría a la realización de las actividades poder acceder a datos, información y conocimiento de proyectos previos. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
	Comentario	

Tabla 4-2 Cuestionario 1 – Fase I Validación Experimental

Fase II: Validación de los *patrones de producto* definidos en el Catálogo de Patrones de Producto realizado en esta tesis doctoral.

El objetivo de la Fase II es la validación de los *patrones de producto* que abarcan las técnicas de análisis, diseño y gestión de un proyecto software, almacenados en el catálogo de patrones y accesible a través de la wiki desarrollada en esta tesis doctoral.

El propósito de esta fase es:

1. Realizar los mismos proyectos software que en la fase I utilizando los *patrones de producto* definidos y almacenados en la wiki, con equipos de desarrollo distintos a los equipos que realizaron los proyectos software sin *patrones de producto* (Fase I).

2. Comprobar que la utilización de los *patrones de producto* facilita el acceso, uso y reutilización del conocimiento de las buenas prácticas en los procesos software; mejora la eficiencia de uso de dichos procesos software, la calidad de los productos obtenidos y por tanto el tiempo de rehacer los productos software (**Hipótesis H1.1**).

Características de la Fase II de la validación experimental:

- **Respecto a los Proyectos.** Se desarrollarán los mismos diez proyectos software que en la Fase I.
- **Respecto al Equipo de Trabajo.** Los ingenieros software que han formado parte del equipo de desarrollo en un proyecto realizado sin la utilización de los *patrones de producto* (Fase I), no formarán parte del equipo de desarrollo del mismo proyecto software realizado con los *patrones de producto* (Fase II). De esta forma se asegura que los ingenieros software no tienen conocimiento previo adquirido de los proyectos software. En esta fase de la validación experimental participarán 30 Ingenieros software, con el mismo perfil y experiencia que los ingenieros software que participen en la Fase I.
- **Respecto a la Metodología.** Se desarrollarán los mismos procesos de las fases de análisis y diseño propuesta por Craig Larman, y el proceso de Gestión de proyecto, que en la Fase I con la característica que en la Fase II se realizan dichos procesos utilizando los *patrones de producto* definidos en el catálogo de *patrones de producto* proporcionados a través de la wiki.
- **Respecto a la Documentación.** En la Fase II se suministra a los equipos de trabajo además de la planificación del proyecto, un manual explicativo de los *patrones de producto* y el catálogo de *patrones de producto*.
- **Respecto a la Formación.** Los equipos de desarrollo recibirán sesiones de formación en el que se les ha explicado: qué es un *patrón de producto*; qué formato y campos tiene; cómo se aplican los *patrones de producto* en el desarrollo software; cómo acceder a la wiki donde está el catálogo de *patrones de producto*; y, cuál es la estructura y el contenido de la wiki. La formación se ha realizado en dos formatos, dependiendo de la ubicación física de los equipos de trabajo que participan en cada proyecto:
 - Formato presencial: una sesión de tres horas de duración.
 - Formato virtual: dos sesiones de dos horas cada una.

Durante la realización de esta fase, los ingenieros software involucrados en los proyectos rellenarán la “Hoja de Seguimiento 2” y el “Cuestionario 2”, los cuales han permitido la recogida de datos para analizar y validar si el desarrollo de los procesos software utilizando el concepto de *Patrón de Producto* como elemento de reutilización de procesos software y encapsulador del conocimiento, mejora la eficiencia de uso de los procesos software, la calidad de los productos obtenidos y por tanto el tiempo de rehacer los productos software (**Hipótesis H1.1**).

En la hoja de seguimiento 2 (Tabla 4-3) se pregunta al equipo de desarrollo sobre el tiempo que han tardado en estudiar y desarrollar la actividad con la ayuda del *patrón de producto*, preguntas relativas a si valoran la utilización del *patrón de producto*, cómo de fácil ha sido utilizar algunos de los campos del patrón y sobre la comunicación entre los miembros del equipo.

Hoja de seguimiento 2 – Fase II Validación Experimental		Valoración
1	Proyecto	
2	Actividad	
3	Rol	
4	¿Ha utilizado el <i>patrón de producto</i> ?	
5	Fecha comienzo	
6	Número de horas de estudio de la solución del <i>patrón de producto</i>	
	Número de horas que ha tardado en realizar la actividad	
7	Prerrequisitos de la Actividad (Productos necesarios para realizar esta actividad)	
	¿Has necesitado algún producto que no estuviera especificado en la entrada del <i>patrón de producto</i> ? ¿Cuál?	
	Comentario	
8	Documentos o producto/s de salida	
	¿Has obtenido algún producto que no estuviera especificado en la salida del <i>patrón de producto</i> ? ¿Cuál?	
	Comentario	
9	¿Qué grado de conocimiento tiene de la actividad (1-10)?	
	¿Con la solución que te proporciona el <i>patrón de producto</i> ha podido realizar la actividad?	
	¿Has utilizado las referencias bibliográficas que proporciona el <i>patrón de producto</i> ?	
	¿Ha necesitado realizar alguna otra búsqueda de información que no le proporcionara las referenciadas bibliográficas que no proporcionara el <i>patrón de producto</i> ? ¿Cuál/es?	
	Comentario	
10	¿Has necesitado colaborar con los miembros del equipo que le menciona el <i>patrón de producto</i> ?	
	¿Has necesitado colaborar con otros miembros del equipo?	
	Número de horas que has tardado en ponerte en contacto con el/los rol/es?	
	Número de horas de interacción con el rol	
	¿Cómo se ha realizado dicha colaboración (email, reuniones, herramienta, etc)?	
	¿Por qué necesitaba la interacción? (Compartir documento, prerrequisito, etc)	
11	¿Has utilizado la plantilla del <i>patrón de producto</i> ?	
	¿Facilita la realización de la actividad la plantilla del <i>patrón de producto</i> ?	
	¿Realizaría alguna modificación en la plantilla del <i>patrón de producto</i> ?	
12	Comentario sobre la realización de la actividad	

Tabla 4-3 Hoja de Seguimiento 2 – Fase II de Validación

En el “Cuestionario 2”, el equipo de desarrollo principalmente tiene que valorar los *patrones de producto* (Tabla 4-4).

Cuestionario 2- Fase II Validación Experimental (Patrones de Producto)		Valoración
1	Los <i>patrones de producto</i> proporcionan el conocimiento para desarrollar las actividades de un proyecto software. 1: muy poco; 2: poco; 3: normal; 4: bastante 5: mucho.	
2	He requerido conocimiento/información que no proporcionaba el <i>patrón de producto</i> . 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
3	Valore los siguientes campos: 1: muy poco valorado; 2: poco; 3: normal; 4: bastante 5: mucho.	
	Solution	
	Template	
	Lessons Learned	
	Information Resource	
4	Considera que sería más usable tener on-line, via web, los patrones de producto. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
5	Considera útil las horas de formación dedicadas al training de patrones de producto. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
6	Le facilitó el <i>patrón de producto</i> reutilizar conocimiento de otros proyectos. 1: nada; 2:poco; 3:normal; 4:bastante; 5: mucho	
7	Conteste si usted realizó el rol de jefe de Proyecto: La elección del <i>patrón de producto</i> me resultó difícil 1: nada; 2:poco; 3:normal; 4: bastante difícil 5: muy difícil	
8	Conteste si usted realizó el rol de jefe de Proyecto: A la hora de realizar la planificación en un diagrama Gantt, le facilitó el campo “solution” el cálculo del tiempo estimado que se tarda en desarrollar una actividad con el <i>patrón de producto</i> . 1: nada; 2:poco; 3:normal; 4: bastante 5: mucho	
9	Conteste si usted realizó el rol de jefe de Proyecto: Valore si el campo "entries" le sirvió para establecer las precedencias entre los patrones de producto y por tanto entre actividades. 1: nada; 2:poco; 3:normal; 4: bastante 5: mucho	

Tabla 4-4 Cuestionario 2– Fase II Validación Experimental

Fase III: Validación del Marco Metodológico a través del despliegue de la solución propuesta en esta tesis doctoral.

El objetivo de esta fase es validar que es posible mejorar la eficiencia de uso de los procesos software, utilizando el marco metodológico propuesto en esta tesis doctoral. Para ello se ha utilizado el elemento que permite encapsular y reutilizar el conocimiento, *patrón de producto*, y la estrategia corporativa, soportado por tecnología colaborativa.

El propósito de esta fase es:

1. Puesta en práctica del Marco Metodológico descrito en esta tesis doctoral a través de una plataforma colaborativa.
2. Comprobar que la utilización del marco metodológico que incluye la utilización de los *patrones de producto* y la estrategia corporativa, a través de la plataforma colaborativa, facilita la difusión, uso y reutilización de los activos de procesos y proyectos en las organizaciones mejorando la productividad (eficiencia de uso) de los proyectos software **(Hipótesis H1.2)**.

Las características de la Fase III de la validación son las siguientes:

- **Respecto a los Proyectos.** Se desarrollarán los mismos diez proyectos software que en la Fase I y la Fase II utilizando el marco metodológico descrito en esta tesis doctoral.
- **Respecto al Equipo de Trabajo.** Al igual que en la fase II, los ingenieros software que van a formar parte del equipo de desarrollo en esta fase serán diferentes a los ingenieros software de las fases anteriores, es decir, aquellos ingenieros software que formaron parte del equipo de desarrollo en un proyecto sin la utilización de los *patrones de producto* (Fase I) o con la utilización de los *patrones de producto* (Fase II), no formarán parte del equipo de trabajo en la realización del mismo proyecto con *el marco metodológico* (Fase III). De esta forma se asegura que los ingenieros software no tienen conocimiento previo adquirido de los proyectos software que van a desarrollar. En esta fase de validación participarán treinta ingenieros software, con el mismo perfil y experiencia que los ingenieros software que han participado en la Fase I y la Fase II.
- **Respecto a la Metodología.** Se desarrollarán los proyectos software siguiendo la metodología propuesta por Craig Larman, centrándose en los procesos y tareas de las

fases de análisis y diseño, y los procesos de Gestión de proyecto, al igual que en la Fase I y la Fase II.

- **Respecto a la Documentación.** En la Fase III se facilitará un manual explicativo de los *patrones de producto*, la *estrategia corporativa* del despliegue de la solución, el catálogo de *patrones de producto* y un manual de usuario de la plataforma web colaborativa.
- **Respecto a la Formación.** A los equipos de desarrollo se les formará sobre: qué es un *patrón de producto*; qué formato y campos tiene; cómo se aplican los *patrones de producto* en el desarrollo software; cómo acceder a la wiki donde está el catálogo de *patrones de producto*; cuál es la estructura y el contenido de la wiki; la *estrategia corporativa* y explicación de la plataforma colaborativa que van a usar. La formación se realizará en dos formatos, dependiendo de la ubicación física de los equipos de trabajo que participen en cada proyecto:
 - Formato presencial: una sesión de cuatro horas de duración.
 - Formato virtual: dos sesiones de tres horas cada una.

Durante la realización de esta fase, los ingenieros software involucrados en los proyectos rellenarán la hoja de seguimiento 3 (Tabla 4-5) y el cuestionario (Tabla 4-6), los cuales permitirán la recogida de datos para analizar y validar que el desarrollo de los proyectos software con el marco metodológico definido en esta tesis doctoral mejora la productividad (eficiencia de uso) de los proyectos software (**Hipótesis H1.2**).

Hoja de seguimiento 3 – Fase III Validación Experimental		Valoración
1	Proyecto	
2	Actividad	
3	Rol	
4	¿Ha utilizado el <i>patrón de producto</i> ?	
5	Fecha comienzo	
6	Número de horas de estudio de la solución del <i>patrón de producto</i>	
	Número de horas que ha tardado en realizar la actividad	
7	Prerrequisitos de la Actividad (Productos necesarios para realizar esta actividad)	
	¿Has necesitado algún producto que no estuviera especificado en la entrada del <i>patrón de producto</i> ? ¿Cuál?	
	Comentario	
8	Documentos o producto/s de salida	
	¿Has obtenido algún producto que no estuviera especificado en la salida del <i>patrón de producto</i> ? ¿Cuál?	
	Comentario	
9	¿Qué grado de conocimiento tiene de la actividad (1-10)?	
	¿Con la solución que te proporciona el <i>patrón de producto</i> ha podido realizar la actividad?	
	¿Has utilizado las referencias bibliográficas que proporciona el <i>patrón de producto</i> ?	
	¿Ha necesitado realizar alguna otra búsqueda de información que no le proporcionara las referenciadas bibliográficas que no proporcionara el <i>patrón de producto</i> ? ¿Cuál/es?	
	Comentario	
10	¿Has necesitado colaborar con los miembros del equipo que le menciona el <i>patrón de producto</i> ?	
	¿Has necesitado colaborar con otros miembros del equipo?	
	Número de horas que has tardado en ponerte en contacto con el/los rol/es?	
	Número de horas de interacción con el rol	
	¿Cómo se ha realizado dicha colaboración (email, reuniones, herramienta, etc)?	
	¿Por qué necesitaba la interacción? (Compartir documento, prerrequisito, etc)	
11	¿Has utilizado la plantilla del <i>patrón de producto</i> ?	
	¿Facilita la realización de la actividad la plantilla del <i>patrón de producto</i> ?	
	¿Realizaría alguna modificación en la plantilla del <i>patrón de producto</i> ?	
12	¿Ha seguido la estrategia corporativa?	
13	¿Los <i>patrones de producto</i> y la estrategia corporativa le han facilitado el entendimiento del proceso?	
14	¿Los <i>patrones de producto</i> y la estrategia corporativa le han facilitado la transferencia y compartición del conocimiento en la organización?	
15	Comentario sobre la realización de la actividad	

Tabla 4-5 Hoja de Seguimiento 3 – Fase III de Validación

En el “Cuestionario 3”, el equipo de desarrollo tiene que valorar los *patrones de producto* y la estrategia corporativa (Tabla 4-6).

Cuestionario 3- Fase III Validación Experimental (Patrones de Producto)		Valoración
1	Los <i>patrones de producto</i> proporcionan el conocimiento para desarrollar las actividades de un proyecto software. 1: muy poco; 2: poco; 3: normal; 4: bastante; 5: mucho.	
2	He requerido conocimiento/información que no proporcionaba el <i>patrón de producto</i> . 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
3	Valore los siguientes campos: 1: muy poco valorado; 2: poco; 3: normal; 4: bastante 5: mucho.	
	Solution	
	Template	
	Lessons Learned	
	Information Resource	
4	Considera que sería más usable tener on-line, via web, los patrones de producto. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
5	Considera útil las horas de formación dedicadas al training de patrones de producto. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
6	Le facilitó el <i>patrón de producto</i> reutilizar conocimiento de otros proyectos. 1: nada; 2:poco; 3:normal; 4:bastante; 5: mucho	
7	Conteste si usted realizó el rol de jefe de Proyecto: La elección del <i>patrón de producto</i> me resultó difícil 1: nada; 2:poco; 3:normal; 4: bastante difícil 5: muy difícil	
8	Conteste si usted realizó el rol de jefe de Proyecto: A la hora de realizar la planificación en un diagrama Gantt, le facilitó el campo “solution” el cálculo del tiempo estimado que se tarda en desarrollar una actividad con el <i>patrón de producto</i> . 1: nada; 2:poco; 3:normal; 4: bastante 5: mucho	
9	Conteste si usted realizó el rol de jefe de Proyecto: Valore si el campo "entries" le sirvió para establecer las precedencias entre los patrones de producto y por tanto entre actividades. 1: nada; 2:poco; 3:normal; 4: bastante 5: mucho	
10	Considera que los <i>patrones de producto</i> facilitan el entendimiento del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
11	Considera que los patrones de producto facilitan el aprendizaje del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
12	Considera que la estrategia corporativa facilita el entendimiento del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
13	Considera que la estrategia corporativa facilita el aprendizaje del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	
14	Considera que la estrategia corporativa junto con la herramienta colaborativa facilita la transferencia del conocimiento. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	

Tabla 4-6 Cuestionario 3– Fase III Validación Experimental

4.2.2 Variables objeto de estudio

Para poder concluir si el uso del Marco Metodológico definido en esta tesis doctoral, que está compuesto de la definición de los *patrones de producto*, la estrategia corporativa y su aplicación mediante una herramienta colaborativa, afecta a los procesos software mejorando la eficiencia de uso de los procesos involucrados en el desarrollo de proyectos software y optimizando ciertos parámetros de calidad de los productos desarrollados, se recogerán datos sobre los proyectos desarrollados en las diferentes fases de validación, y a partir de dichos datos se observarán las diferentes variables que se desea medir.

Las variables que se van a medir son, variables cuantitativas continuas (variables que adquieren cualquier valor dentro de un intervalo especificado de valores), y variables cualitativas ordinales (variables que toman distintos valores ordenados siguiendo una escala establecida).

Las variables cuantitativas continuas se recogerán de las hojas de seguimiento que han cumplimentado los ingenieros software en las tres fases de validación. Estas variables son:

- Para medir la **Eficiencia de Uso de los Procesos Software**:
 - A. **Tiempo de desarrollo.** se pretende comprobar :
 - en la Fase II: si el hecho de tener los *patrones de producto*, que aportan al equipo de desarrollo el conocimiento y la experiencia de proyectos previos a través de una guía de pasos que les proporciona cómo realizar las tareas del proyecto, lecciones aprendidas, plantillas, etc; incrementa o reduce el tiempo de desarrollo del proyecto.
 - en la Fase III: si el marco metodológico que aporta los patrones de producto y una estrategia corporativa en una herramienta colaborativa; incrementa o reduce el tiempo de desarrollo del proyecto.
- Para medir la **Calidad de los Productos Software**: además de comprobar si el tiempo de desarrollo software se incrementa o reduce, se va a comprobar cómo es la calidad de los productos software obtenidos en cada fase de validación. Para medir la calidad de los productos software se observarán las siguientes variables:

- B. Volatilidad:** Se medirá la trazabilidad de los requisitos de los proyectos software comprobando si todos los requisitos de la actividad especificación de requisitos están representados en el diagrama de casos de uso.
- C. El número de clases y métodos en el diagrama de clases:** Se medirán cuántas clases y métodos no se han identificado en el diagrama de clase.
- D. El Tiempo de Rework:** El tiempo de rework es una variable importante en todos los proyectos software ya que es el tiempo empleado en rehacer los productos software una y otra vez hasta que pasan el nivel de calidad exigido. Se medirá el **número de revisiones realizadas en los productos**, esta variable nos permitirá observar si el hecho de tener el conocimiento tácito de cómo se realizan las actividades, minimiza o incrementa el número de veces que es necesario repetir una tarea.

Las variables cualitativas ordinarias se han recogido a través de los cuestionarios cumplimentados por los ingenieros software en las diferentes fases de validación. Estas variables son:

- E. Grado de satisfacción del equipo de trabajo utilizando los patrones de producto y el marco metodológico propuesto:** se pretende comparar el grado de aceptación y de satisfacción que han tenido, realizando los proyectos sin utilizar los patrones de producto y el marco metodológico, y siguiéndolos.
- F. Comunicación entre miembros del equipo de trabajo:** se pretende comprobar si, gracias al marco metodológico que se propone en la presente tesis doctoral, se mejora la comunicación entre los miembros del equipo de desarrollo.

4.2.3 Proyectos y participantes

En las tres fases de la validación, se realizarán los mismos diez proyectos software. Los proyectos software tendrán las mismas características: proyectos software de gestión cuyo tiempo estimado de desarrollo sea de 6 meses.

Para tener una muestra amplia y abarcar diferentes contextos en el desarrollo de los proyectos software, se realizarán proyectos software en el ámbito académico y en el ámbito empresarial. En ambos contextos, el perfil de los participantes será similar, Ingenieros Técnicos en Informática de Gestión e Ingeniero Informático.

4.3 EJECUCIÓN DE LA VALIDACIÓN EXPERIMENTAL

En este apartado se describe la ejecución y análisis de cada una de las fases de validación.

4.3.1 Contexto

Se han desarrollado los mismos proyectos en las tres fases de validación, a continuación se describen brevemente:

- **Proyecto 1:** un sistema web que gestione las reservas y los pedidos de una cadena de restaurantes.
- **Proyecto 2:** un sistema web que gestione la seguridad de un parque de atracciones.
- **Proyecto 3:** un sistema web que gestione un club deportivo. El sistema permitirá el control de actividades culturales y deportivas.
- **Proyecto 4:** un sistema web que gestione las noticias y servicios de un periódico.
- **Proyecto 5:** un sistema web que gestione los procedimientos de una entidad pública y que permita a los usuarios consultar sus procedimientos.
- **Proyecto 6:** un sistema web que gestione los historiales médicos y las consultas de un hospital.
- **Proyecto 7:** un sistema web que gestione la matriculación de estudiantes en un colegio.
- **Proyecto 8:** un sistema web que gestione las representaciones y eventos de un teatro.
- **Proyecto 9:** un sistema web que gestione las reservas de una cadena hotelera.
- **Proyecto 10:** un sistema web que gestione los procedimientos de una clínica veterinaria.

El ámbito de cada proyecto ha sido:

- **Ámbito académico:** se han desarrollado 6 proyectos software (proyectos 1 al proyecto 6). Los equipos de desarrollo involucrados en estos 6 proyectos fueron

alumnos titulados en Ingeniería Técnica en Informática de Gestión e Ingeniería Informática en la Universidad Carlos III de Madrid. Dichos ingenieros tenían experiencia de dos a cuatro años en el desarrollo de proyectos software. Estos equipos de trabajo recibieron las sesiones de formación de tipo presencial.

- **Ámbito empresarial:** se han desarrollado 4 proyectos software en el ámbito empresarial. Las empresas en las que se han desarrollado los proyectos son dos pymes mexicanas, “SIAS” y “Alonso Sw”, dedicadas a la consultoría de proyectos software. En la empresa “SIAS” se han realizado los proyectos 7 y 8; y, en la empresa “Alonso Sw” se han realizado los proyectos 9 y 10. En el caso de los equipos de trabajo mexicanos, son titulados Ingenieros Informáticos con experiencia profesional, 5 años, en el desarrollo de proyectos Web, pero sin amplio conocimiento en el uso de metodologías ni modelos de procesos en proyectos software.

4.3.2 Ejecución de las fases de validación

Fase I: Realización de los proyectos software sin el uso de los Patrones de Producto ni del Marco Metodológico.

En la Fase I se han desarrollado diez proyectos software sin la utilización de los *patrones de producto* ni del marco metodológico, únicamente con el conocimiento y la experiencia del jefe de proyecto y su equipo de trabajo. Los proyectos software se han desarrollado (Figura 4-1):

- En el ámbito académico, seis proyectos software (Proyecto 1 al Proyecto 6), entre Mayo de 2006 y Febrero de 2007.
- En el ámbito empresarial, cuatro proyectos (Proyecto 7 al Proyecto 10) entre Septiembre 2007 y Febrero del 2008.

adecuaban a cada actividad del proyecto, basándose en el problema, restricciones y contexto de la actividad. De esta forma, cada rol del proyecto sabe en cada momento qué tiene que hacer y cómo lo tiene que realizar.

Los *patrones de producto* utilizados son:

- En la Fase de Análisis:
 - *Patrón de Producto* Especificación de Requisitos.
 - *Patrón de Producto* Diagrama de Casos de Uso.
 - *Patrón de Producto* Casos de Uso en Formato Alto Nivel.
 - *Patrón de Producto* Casos de Uso en Formato Expandido.

- En la Fase de Diseño:
 - *Patrón de Producto* Diagrama de Clases.
 - *Patrón de Producto* Diagrama de Secuencia.
 - *Patrón de Producto* Diagrama de Colaboración.

- En la Fase de Gestión:
 - *Patrón de Producto* Estimación Puntos de Función.
 - *Patrón de Producto* Estimación COCOMO II.
 - *Patrón de Producto* Planificación Diagrama Gantt.

Los proyectos software se realizaron:

- En el ámbito académico, los seis proyectos software (Proyecto 1 al Proyecto 6) entre Noviembre de 2006 y Junio de 2007, en los cuales participaron 18 ingenieros software. La formación impartida a los ingenieros software fue una sesión presencial de tres horas de duración.

- En el ámbito empresarial, se desarrollaron los cuatro proyectos en paralelo entre Marzo 2008 y Junio del 2008. La formación que recibieron los ingenieros software que participaron en los proyectos, fue de dos sesiones virtuales de dos horas de duración.

En la Figura 4-2 se muestra la planificación de la Fase I y la Fase II para que se tenga una visión general de la realización de las dos fases.

2006												2007												2008					
Abr	May	Jun	Jul	Ag	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ag	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun			
Proyecto 1- SE1, SE2, SE3							Proyecto 1- SE4, SE5, SE6																						
Proyecto 2- SE4, SE5, SE6							Proyecto 2- SE7, SE8, SE9																						
Proyecto 3- SE7, SE8, SE9							Proyecto 3- SE1, SE2, SE3																						
							Proyecto 4- SE10, SE11, SE12					Proyecto 4- SE13, SE14, SE15					Proyecto 7- SE37, SE38, SE39						Proyecto 7- SE40, SE41, SE42						
							Proyecto 5- SE13, SE14, SE15					Proyecto 5- SE16, SE17, SE18					Proyecto 8- SE40, SE41, SE42						Proyecto 8- SE43, SE44, SE45						
							Proyecto 6- SE16, SE17, SE18					Proyecto 6- SE10, SE11, SE12					Proyecto 9- SE43, SE44, SE45						Proyecto 9- SE46, SE47, SE48						
																	Proyecto 10- SE46, SE47, SE48						Proyecto 10- SE37, SE38, SE39						
Fase I																													
Fase II																													

Figura 4-2 Planificación Fase I y Fase II

La Fase II se ha realizado en dos sub-fases:

- Fase II- Sub-Fase I: se realizaron los proyectos 1, 2 y 3 con el uso de los *patrones de producto* definidos en su versión inicial. Después de realizar los proyectos, los equipos de trabajo rellenaron la Hoja de Seguimiento 2 y el Cuestionario 2, se recogieron los datos de las variables a medir para su posterior comparación con la Fase I, y se extrajeron los comentarios y sugerencias referentes a los campos de los *patrones de producto*.
- Fase II-Sub-Fase II: siguiendo con las sugerencias y comentarios recogidos de la hoja de seguimiento y el cuestionario realizado por los ingenieros software que participaron en la Sub-Fase I, un nuevo campo (Conocimiento Básico) se añadió a la representación de los *patrones de producto*. La nueva versión de los *patrones de producto* se aplicó al resto de proyectos de la Fase II (Proyectos 4 al Proyecto 10). Los ingenieros software que participaron en dichos proyectos rellenaron la Hoja de Seguimiento 2 y el Cuestionario 2, para la recogida de datos y su posterior análisis y comparación con la Fase I.

En la Tabla 4-8 se muestran los participantes de la Fase I y la Fase II para resaltar que los ingenieros software que participaron en un proyecto en la Fase II no fueron los mismos ingenieros software que desarrollaron ese mismo proyecto en la Fase I. De esa forma se puede

asegurar que los ingenieros software no tienen el conocimiento previo de los requisitos del proyecto y por tanto no afecta al resultado de la validación experimental.

PROYECTOS	FASE I (Sin Patrón de Producto – Sin Marco Metodológico)	FASE II (Con Patrón de Producto)	
		Sub-Fase I Versión Inicial del Patrón de Producto	Sub-Fase II Versión Final del Patrón de Producto
Proyecto 1	SE1, SE2, SE3	SE4, SE5, SE6	
Proyecto 2	SE4, SE5, SE6	SE7, SE8, SE9	
Proyecto 3	SE7, SE8, SE9	SE1, SE2, SE3	
Proyecto 4	SE10, SE11, SE12		SE13, SE14, SE15
Proyecto 5	SE13, SE14, SE15		SE16, SE17, SE18
Proyecto 6	SE16, SE17, SE18		SE10, SE11, SE12
Proyecto 7	SE37, SE38, SE39		SE40, SE41, SE42
Proyecto 8	SE40, SE41, SE42		SE43, SE44, SE45
Proyecto 9	SE43, SE44, SE45		SE46, SE47, SE48
Proyecto 10	SE46, SE47, SE48		SE37, SE38, SE39

SE: Ingeniero software

Tabla 4-8 Participantes de la Fase I y Fase II

Fase III: Validación del Marco Metodológico a través del despliegue de la solución propuesta en esta tesis doctoral

En esta fase se han realizado los mismos diez proyectos software que en la Fase I y la Fase II con la versión final de los *patrones de producto* implementado mediante una wiki (<http://productpatterns.sel.inf.uc3m.es/>) y la estrategia corporativa implantada en una plataforma colaborativa para el desarrollo de los proyectos software.

Los *patrones de producto* utilizados han sido los mismos que en la Fase II, son:

- En la Fase de Análisis:
 - *Patrón de Producto* Especificación de Requisitos
 - *Patrón de Producto* Diagrama de Casos de Uso.
 - *Patrón de Producto* Casos de Uso en Formato Alto Nivel
 - *Patrón de Producto* Casos de Uso en Formato Expandido
- En la Fase de Diseño:
 - *Patrón de Producto* Diagrama de Clases
 - *Patrón de Producto* Diagrama de Secuencia

- *Patrón de Producto* Diagrama de Colaboración
- En la Fase de Gestión:
 - *Patrón de Producto* Estimación Puntos de Función
 - *Patrón de Producto* Estimación COCOMO II
 - *Patrón de Producto* Planificación Diagrama Gantt

Los proyectos software se realizaron:

- En el ámbito académico, los seis proyectos software (Proyecto 1 al Proyecto 6) entre Septiembre de 2007 y Abril de 2008, en los cuales participaron 18 ingenieros software. La formación impartida a los ingenieros software fue una sesión presencial de cuatro horas de duración.
- En el ámbito empresarial, se desarrollaron los cuatro proyectos en paralelo entre Septiembre 2008 y Diciembre del 2008. La formación que recibieron los ingenieros software fueron dos sesiones virtuales de tres horas de duración.

A continuación se muestra en la Figura 4-3 la planificación de la tres fase de validación.

2006												2007												2008											
Abr	May	Jun	Jul	Ag	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ag	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul	Ag	Sep	Oct	Nov	Dic			
Proyecto 1- SE1, SE2, SE3						Proyecto 1- SE4, SE5, SE6						Proyecto 1- SE19, SE20, SE21						Proyecto 4- SE28, SE29, SE30																	
Proyecto 2- SE4, SE5, SE6						Proyecto 2- SE7, SE8, SE9						Proyecto 2- SE22, SE23, SE24						Proyecto 5- SE31, SE32, SE33																	
Proyecto 3- SE7, SE8, SE9						Proyecto 3- SE1, SE2, SE3						Proyecto 3- SE25, SE26, SE27						Proyecto 6- SE34, SE35, SE36																	
Proyecto 4- SE10, SE11, SE12						Proyecto 4- SE13, SE14, SE15						Proyecto 7- SE37, SE38, SE39						Proyecto 7- SE40, SE41, SE42																	
Proyecto 5- SE13, SE14, SE15						Proyecto 5- SE16, SE17, SE18						Proyecto 8- SE40, SE41, SE42						Proyecto 8- SE43, SE44, SE45																	
Proyecto 6- SE16, SE17, SE18						Proyecto 6- SE10, SE11, SE12						Proyecto 9- SE43, SE44, SE45						Proyecto 9- SE46, SE47, SE48																	
Proyecto 10- SE46, SE47, SE48						Proyecto 10- SE37, SE38, SE39						Proyecto 10- SE49, SE50, SE51						Proyecto 10- SE52, SE53, SE54																	
Proyecto 10- SE49, SE50, SE51						Proyecto 10- SE52, SE53, SE54						Proyecto 10- SE55, SE56, SE57						Proyecto 10- SE58, SE59, SE60																	

Figura 4-3 Planificación Fases de validación

En la Tabla 4-9 se muestran los participantes de las tres fases. Es importante reiterar que los ingenieros software que han participado en la Fase III no son los mismos ingenieros software que realizaron los mismos proyectos en la Fase I y la Fase II. De esa forma se asevera que los ingenieros software además de no tener el conocimiento previo de los requisitos del proyecto no tienen conocimiento sobre los *patrones de producto*.

PROYECTOS	FASE I (Sin Patrón de Producto – Sin Marco Metodológico)	FASE II (Con Patrón de Producto)		FASE III (Con Patrón de Producto y Marco Metodológico)
		Sub-Fase I Versión Inicial del Patrón de Producto	Sub-Fase II Versión Final del Patrón de Producto	
Proyecto 1	SE1, SE2, SE3	SE4, SE5, SE6		SE19, SE20, SE21
Proyecto 2	SE4, SE5, SE6	SE7, SE8, SE9		SE22, SE23, SE24
Proyecto 3	SE7, SE8, SE9	SE1, SE2, SE3		SE25, SE26, SE27
Proyecto 4	SE10, SE11, SE12		SE13, SE14, SE15	SE28, SE29, SE30
Proyecto 5	SE13, SE14, SE15		SE16, SE17, SE18	SE31, SE32, SE33
Proyecto 6	SE16, SE17, SE18		SE10, SE11, SE12	SE34, SE35, SE36
Proyecto 7	SE37, SE38, SE39		SE40, SE41, SE42	SE49, SE50, SE51
Proyecto 8	SE40, SE41, SE42		SE43, SE44, SE45	SE52, SE53, SE54
Proyecto 9	SE43, SE44, SE45		SE46, SE47, SE48	SE55, SE56, SE57
Proyecto 10	SE46, SE47, SE48		SE37, SE38, SE39	SE58, SE59, SE60

SE: Ingeniero software

Tabla 4-9 Participantes Fases de validación

4.4 ANÁLISIS DE LOS DATOS OBTENIDOS EN LA VALIDACIÓN EXPERIMENTAL

4.4.1 Análisis cuantitativo

De los datos recogidos a través de las hojas de seguimiento se han analizado las variables cuantitativas definidas en la validación experimental. Para realizar dicho análisis se ha utilizado el procedimiento “contraste de hipótesis”, que servirá para aceptar o rechazar una hipótesis que se emita acerca de un parámetro de la validación experimental.

Se han definido en cada una de las fases de la validación muestras independientes ya que la observación de una de ellas no condiciona en nada a la observación de las otras.

Además, en cada una de las variables se han definido las hipótesis nulas (H_0) correspondientes sobre lo que se desea decidir: H_0 = los valores de las muestras son iguales; y, se ha realizado el test T-Student para comprobar mediante la distribución t de Student si la hipótesis nula es cierta. Se ha aplicado el test t-Student porque se asume que las muestras siguen una distribución de probabilidad normal (describe el rango de valores de la variable aleatoria así como la probabilidad de que el valor de la variable aleatoria esté dentro de un subconjunto de dicho rango) pero el tamaño muestral es pequeño como para que el estadístico en el que está basada la inferencia esté normalmente distribuido, utilizándose una estimación de la desviación típica (medida cuadrática que informa de la media de distancias que tienen los

datos respecto de su media aritmética), expresada en las mismas unidades que la variable en lugar del valor real.

Fase II: Validación de los *patrones de producto* definidos en el Catálogo de Patrones de Producto realizado en esta tesis doctoral.

Se han definido dos muestras independientes para todas las variables a analizar:

- Muestra 1: representa los proyectos que se han desarrollado sin *patrones de producto* (**Fase I**).
- Muestra 2: representa los proyectos que se han desarrollado con *patrones de producto* implementados en la wiki (<http://productpatterns.sel.inf.uc3m.es>) (**Fase II**).

A continuación se describe el análisis cuantitativo realizado en la Fase II de validación de cada una de las variables.

A. Eficiencia de Uso de los Procesos Software (Objetivo 1). Se han recogido los valores del tiempo que se ha tardado en desarrollar cada producto software. Para una mayor claridad, se analizará la eficiencia de uso de los procesos software por fases del ciclo de vida (fase de análisis, fase de diseño y fase de gestión).

A.1. Fase de Análisis: los modelos realizados en la fase de análisis son: la Especificación de Requisitos, el Diagrama de Casos de Uso, Casos de Uso alto nivel, Casos de Uso expandidos.

Las hipótesis nulas definidas para comprobar la eficiencia de uso de las actividades de la fase de análisis son:

- H0-ER: el tiempo de desarrollo para realizar la especificación de requisitos de ambas muestras es igual.
- H0-UC: el tiempo de desarrollo para realizar el diagrama de casos de uso en ambas muestras es igual.
- H0-UCH: el tiempo de desarrollo para realizar los casos de uso de alto nivel en ambas muestras es igual.
- H0-UCE: el tiempo de desarrollo para realizar los casos de uso expandidos en ambas muestras es igual.

Las variables definidas son:

- Var 1: tiempo (en hora) de desarrollo en realizar la especificación de requisitos en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 2: tiempo (en hora) de desarrollo en realizar la especificación de requisitos en un proyecto donde los patrones de producto si fueron usados (muestra 2).
- Var 3: tiempo (en hora) de desarrollo en realizar el diagrama de casos de uso en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 4: tiempo (en hora) de desarrollo en realizar el diagrama de casos de uso en un proyecto donde los patrones de producto si fueron usados (muestra 2).
- Var 5: tiempo (en hora) de desarrollo en realizar el caso de uso de alto nivel en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 6: tiempo (en hora) de desarrollo en realizar el caso de uso de alto nivel en un proyecto donde los patrones de producto si fueron usados (muestra 2).
- Var 7: tiempo (en hora) de desarrollo en realizar el caso de uso expandido en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 8: tiempo (en hora) de desarrollo en realizar el caso de uso expandido en un proyecto donde los patrones de producto si fueron usados (muestra 2).

En la Tabla 4-10 se muestra un resumen de los resultados estadísticos y en la Figura 4-4 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Especificación de Requisitos		Diagrama de Casos de Uso		Casos de Uso Alto Nivel		Casos de Uso Expandidos	
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8
Recuento	10	10	10	10	10	10	10	10
Media	8.5	10.8	10.6	7.22	4.9	2.6	4.3	2.5
Moda	8	11	10	6.5	4	2	4	2
Varianza	1.61	1.73	0.93	0.77	0,76	0,48	0,45	0.27
Desviación Std	1.26	1.31	0.96	0.88	0,87	0,69	0.67	0.52
Mínimo	7	8	9	6.3	4	2	3	2
Máximo	11	13	12	9	6	4	5	3
Rango	4	5	3	2.7	2	2	2	1
Std. skewness	1.05	-0.82	0.14	1.21	0,28	1,01	-0.56	0.0
Std. kurtosis	0.16	1.31	-0.40	0.09	-1.11	-0.09	-0.18	-1.65
T student	-3,98		8,17		6,49		6.65	
P - Value	0,001		0,000		0,000		0,000	

Tabla 4-10 Resumen estadístico: Eficiencia de Uso – Fase de Análisis

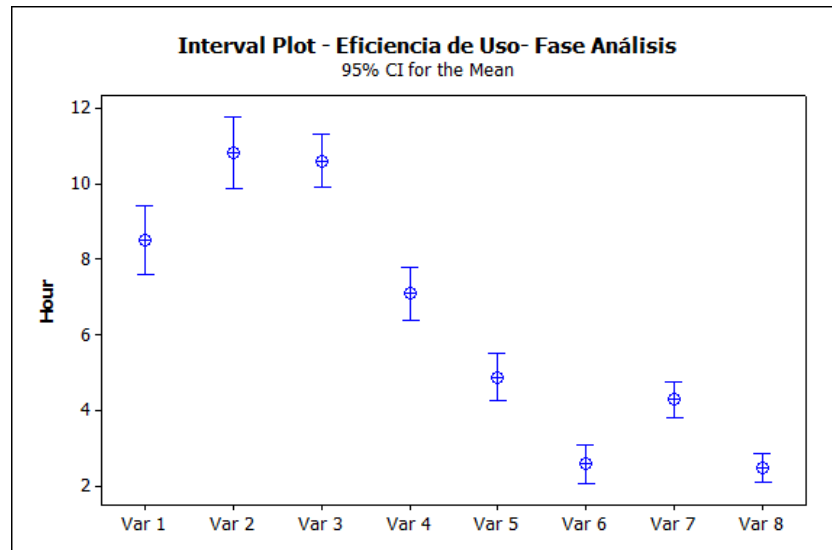


Figura 4-4 Interval Plot – Eficiencia de Uso - Fase Análisis

Con valores p-value de 0,001 para la especificación de requisitos; 0.000 para el diagrama de casos de uso; los casos de uso de alto nivel y los casos de uso expandidos todos ellos menores de 0,05; se puede rechazar las hipótesis H0-ER, H0-UC, H0-UCH y H0-UCE respectivamente; por lo que se afirma que el tiempo de desarrollo en realizar los cuatro modelos de análisis en ambas muestras es diferente. Como se puede observar en la Figura 4-4, en todas las actividades excepto en la actividad especificación de requisitos, el tiempo de realizar la actividad con *patrones de producto* es menor que si se realiza la actividad sin patrones de producto. La excepción, en la actividad especificación de requisitos se puede afirmar que ocurre por el tiempo de aprendizaje en utilizar los *patrones de producto*.

A.2. Fase de Diseño: los modelos realizados son el Diagrama de Clases, Diagrama de Secuencia y el Diagrama de Colaboración.

Las hipótesis nulas definidas para comprobar la eficiencia de uso de las actividades de la fase de diseño son:

- H0-CLD: el tiempo de desarrollo para realizar el diagrama de clases de ambas muestras es igual.
- H0-SD: el tiempo de desarrollo para realizar el diagrama de secuencia en ambas muestras es igual.
- H0-CD: el tiempo de desarrollo para realizar el diagrama de colaboración en ambas muestras es igual.

Las variables definidas son:

- Var 1: tiempo (en hora) de desarrollo en realizar el diagrama de clases en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 2: tiempo (en hora) de desarrollo en realizar el diagrama de clases en un proyecto donde los patrones de producto si fueron usados (muestra 2).
- Var 3: tiempo (en hora) de desarrollo en realizar el diagrama de secuencia en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 4: tiempo (en hora) de desarrollo en realizar el diagrama de secuencia en un proyecto donde los patrones de producto si fueron usados (muestra 2).
- Var 5: tiempo (en hora) de desarrollo en realizar el diagrama de colaboración en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 6: tiempo (en hora) de desarrollo en realizar el diagrama de colaboración en un proyecto donde los patrones de producto si fueron usados (muestra 2).

En la Tabla 4-11 se muestra un resumen de los resultados estadísticos y en la Figura 4-5 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Diagrama de Clases		Diagrama de Secuencia		Diagrama de Colaboración	
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6
Recuento	10	10	10	10	10	10
Media	16,9	20.3	13.3	10.6	12	9.2
Moda	19	18	14	11	12	9
Varianza	3.65	6.67	1.56	1.6	0.66	0.84
Desviación Std	1.91	2.58	1.25	1.26	0.81	0.91
Mínimo	14	17	11	9	11	8
Máximo	19	24	15	12	13	11
Rango	5	7	4	3	2	3
Std. skewness	-0.23	0.21	-0.91	-0.36	0.0	0.77
Std. kurtosis	-1.12	-1.19	-0.31	-1.07	-0.89	0.25
T student	-3.34		4,80		7,20	
P - Value	0,004		0,000		0,000	

Tabla 4-11 Resumen estadístico: Eficiencia de Uso – Fase de Diseño

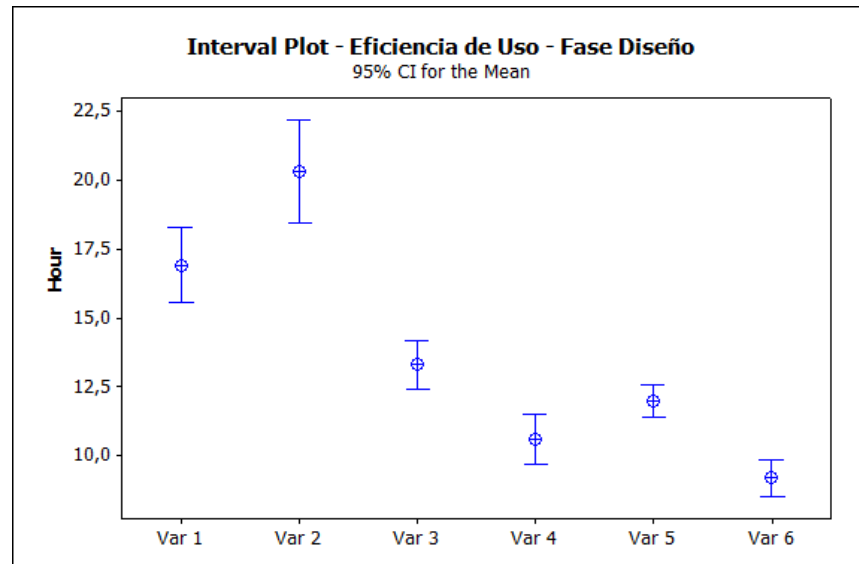


Figura 4-5 Interval Plot – Eficiencia de Uso - Fase Diseño

Con valores p-value de 0,004 para el diagrama de clases; 0.000 para el diagrama de secuencia y el diagrama de colaboración, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-CLD, H0-SD y H0-CD respectivamente; por lo que se afirma que el tiempo de desarrollo en realizar los tres modelos de diseño en ambas muestras es diferente.

A.3. Fase de Gestión: los modelos realizados son la Estimación utilizando los Puntos de Función de Albretch, Estimación utilizando Cocomo II y la Planificación realizando un Diagrama Gantt.

Las hipótesis nulas definidas para comprobar la eficiencia de uso de las actividades de la fase de gestión son:

- H0-AE: el tiempo de desarrollo para realizar la estimación utilizando los puntos de función Albretch de ambas muestras es igual.
- H0-CE: el tiempo de desarrollo para realizar la estimación utilizando Cocomo II en ambas muestras es igual.
- H0-GP: el tiempo de desarrollo para realizar la planificación utilizando el diagrama Gantt en ambas muestras es igual.

Las variables definidas son:

- Var 1: tiempo (en hora) de desarrollo en realizar la estimación usando los puntos de función en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 2: tiempo (en hora) de desarrollo en realizar la estimación usando los puntos de función en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 3: tiempo (en hora) de desarrollo en realizar la estimación usando Cocomo II en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 4: tiempo (en hora) de desarrollo en realizar la estimación usando Cocomo II en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 5: tiempo (en hora) de desarrollo en realizar la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 6: tiempo (en hora) de desarrollo en realizar la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto si fueron usados (muestra 2)

En la Tabla 4-12 se muestra un resumen de los resultados estadísticos y en la Figura 4-6 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Estimación Puntos de Función		Estimación Cocomo II		Planificación Diagrama Gantt	
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6
Recuento	10	10	10	10	10	10
Media	9.1	6.9	9.2	6.6	7.2	4.7
Moda	9	6	9	6	8	5
Varianza	0.98	0.98	1.06	0.48	1.06	0.45
Desviación Std	0.99	0.99	1.03	0.69	1.03	0.67
Mínimo	8	6	8	6	5	4
Máximo	11	8	11	8	8	6
Rango	3	2	3	2	3	2
Std. skewness	0.78	0.31	0.35	1,01	-1.61	0.55
Std. kurtosis	-1.10	-1.48	-0.57	-0.09	0.61	-0.18
T student	4,95		6.59		6,41	
P - Value	0.000		0.000		0.000	

Tabla 4-12 Resumen estadístico: Eficiencia de Uso – Fase de Gestión

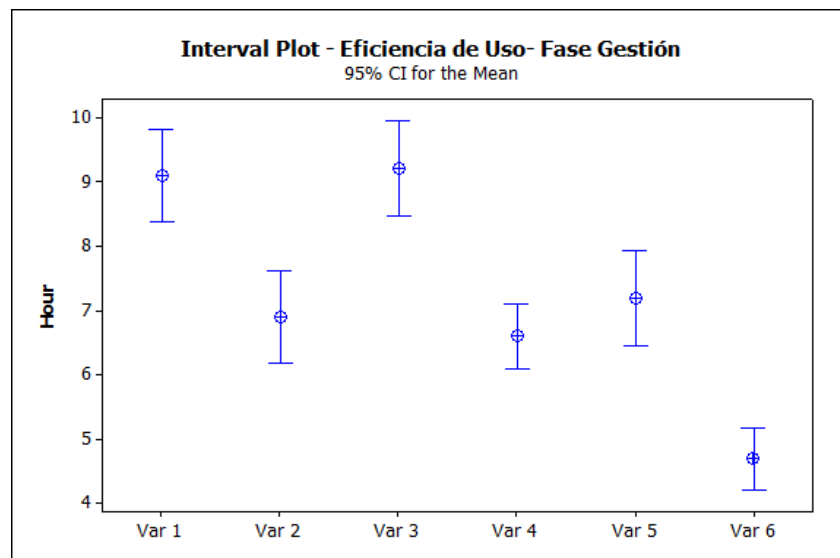


Figura 4-6 Interval Plot – Eficiencia de Uso - Fase Gestión

Con valores p-value de 0,000 para la estimación mediante puntos de función; la estimación mediante COCOMO II y la planificación mediante Diagrama Gantt, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-AE, H0-CE y H0-GP respectivamente; por lo que se afirma que el tiempo de desarrollo en realizar las tres actividades de gestión en ambas muestras es diferente. Como se puede observar en la Figura 4-6, podemos afirmar que el tiempo de desarrollo en las tres actividades donde los *patrones de producto* fueron utilizados es menor que en los proyectos donde no se utilizaron los *patrones de producto*.

B. Volatilidad de los Requisitos. Es muy importante determinar el grado de volatilidad de los requisitos, este parámetro se va a medir mediante el número de requisitos que son identificados en la especificación de requisitos pero no son satisfecho en el diagrama de casos de uso. Para obtener esta información, la descripción del caso de uso tiene un campo donde se indica qué requisitos es satisfecho en ese caso de uso.

La hipótesis nula definida en esta variable es:

- H0-MR: el número de requisitos perdidos en el diagrama de casos de uso en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de requisitos perdidos en el diagrama de casos de uso en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 2: número de requisitos perdidos en el diagrama de casos de uso en un proyecto donde los patrones de producto si fueron usados (muestra 2).

En la Tabla 4-13 se muestra un resumen de los resultados estadísticos y en la Figura 4-7 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Número de requisitos perdidos	
	Var 1	Var 2
Recuento	10	10
Media	8	5
Moda	9	5
Varianza	1.11	1.77
Desviación Std	1.05	1.33
Mínimo	6	3
Máximo	9	7
Rango	3	4
Std. skewness	-0.91	-0.45
Std. kurtosis	-0.29	-0.48
T student	5,58	
P - Value	0,000	

Tabla 4-13 Resumen estadístico: Volatilidad de Requisitos

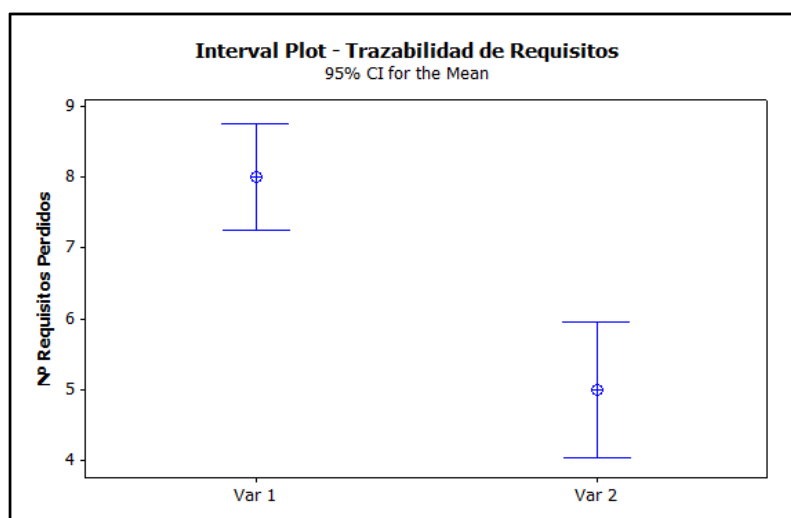


Figura 4-7 Interval Plot – Volatilidad de Requisitos

Con p-value de 0,000 menor de 0,05; significa que se puede rechazar las hipótesis H0-MR; y por tanto se puede afirmar que el número de requisitos perdidos en el diagrama de casos de uso en ambas muestras es diferente. Como se puede observar en la Figura 4-7, podemos afirmar que el número de requisitos perdidos en el diagrama de casos de uso donde los patrones de producto fueron utilizados es menor que en los proyectos donde no se utilizaron los patrones de producto.

C. Clases y métodos no identificados. Otro parámetro que se analizó fue el número de clases y métodos no identificados en la primera versión de los diagramas de clase de cada proyecto.

C.1. Número de Clases no identificadas: se recogió el número de clases no identificadas en la primera versión de los diagramas de clase desarrollados en los 10 proyectos software.

La hipótesis nula definida en esta variable es:

- H0-CNI: el número de clases no identificadas en la primera versión del diagrama de clases en ambas muestras es el mismo.

Las variables definidas son:

- Var 1: número de clases del diagrama de clases en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 2: número de clases del diagrama de clases en un proyecto donde los patrones de producto si fueron usados (muestra 2).

En la Tabla 4-14 se muestra un resumen de los resultados estadísticos y en la Figura 4-8 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Número de clases No Identificadas	
	Var 1	Var 2
Recuento	10	10
Media	6	3.1
Moda	6	4
Varianza	2.66	1.43
Desviación Std	1.63	1.19
Mínimo	4	1
Máximo	9	5
Rango	5	4
Std. skewness	0.74	-0.31
Std. kurtosis	-0.18	-0.23
T student	4,53	
P - Value	0,000	

Tabla 4-14 Resumen estadístico: Número de Clases No Identificadas

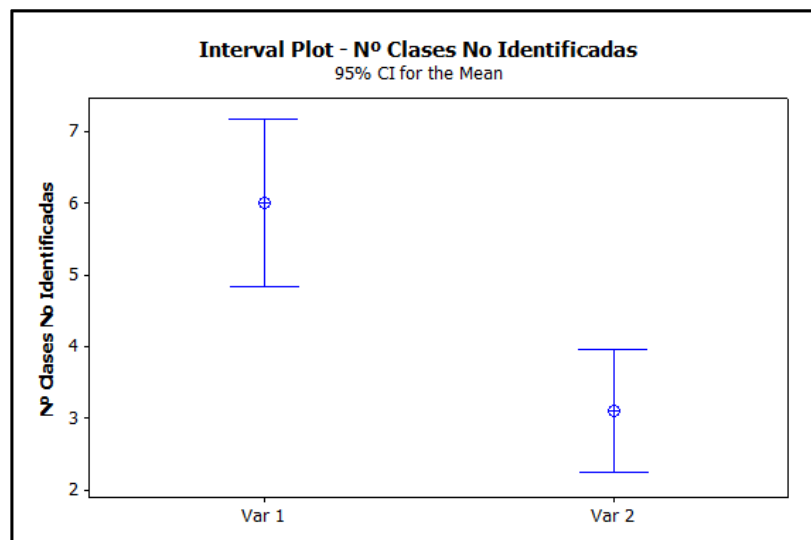


Figura 4-8 Interval Plot – Número de Clases No Identificadas

Con p-value de 0,000 menor de 0,05 se puede rechazar las hipótesis H_0 -CNI; por tanto se puede afirmar que el número de clases no identificadas en ambas muestras es diferente. Como se puede observar en la Figura 6-8, podemos afirmar que el número de clases no identificadas en el diagrama de clases donde los patrones de producto fueron utilizados es menor que en los proyectos donde no se utilizaron los *patrones de producto*.

C.2. Número de Métodos no identificados: se recogió el número de métodos no identificados en la primera versión de los diagramas de clase desarrollados en los 10 proyectos software.

La hipótesis nula definida en esta variable es:

- H0-MNI: el número de métodos no identificados en la primera versión del diagrama de clases en ambas muestras es el mismo.

Las variables definidas son:

- Var 1: número de métodos del diagrama de clases en un proyecto donde los patrones de producto no fueron usados (muestra 1).
- Var 2: número de métodos del diagrama de clases en un proyecto donde los patrones de producto si fueron usados (muestra 2).

En la Tabla 4-15 se muestra un resumen de los resultados estadísticos y en la Figura 4-9 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Número de Métodos No Identificados	
	Var 1	Var 2
Recuento	10	10
Media	11.2	5.5
Moda	12	5
Varianza	9.51	1.61
Desviación Std	3.08	1.26
Mínimo	6	4
Máximo	16	8
Rango	10	4
Std. skewness	-0.08	1.05
Std. kurtosis	-0.29	0.16
T student	5,40	
P - Value	0,000	

Tabla 4-15 Resumen estadístico: Número de Métodos No Identificados

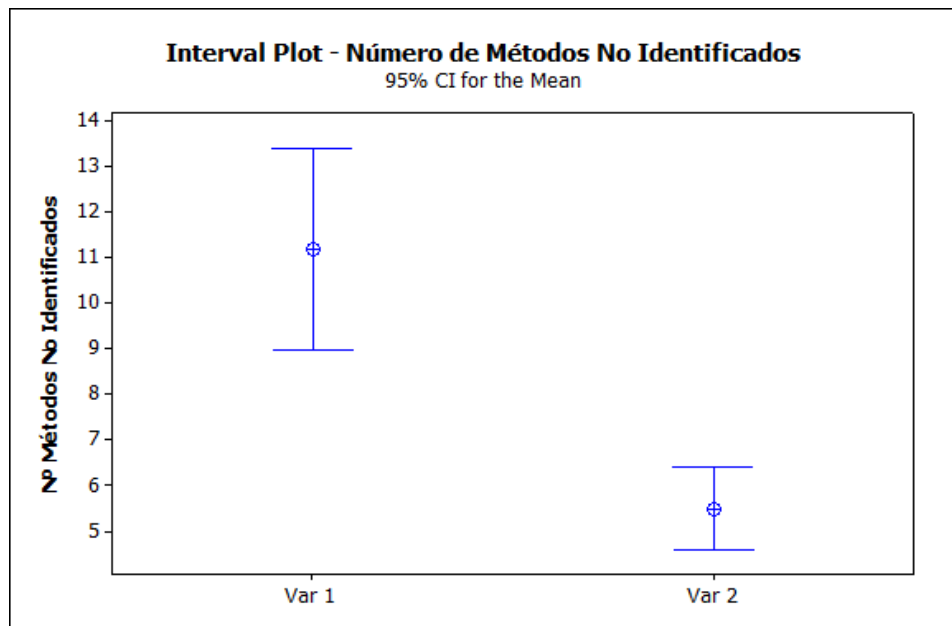


Figura 4-9 Interval Plot – Número de Métodos No Identificados

Con p-value de 0,000 menor de 0,05 se puede rechazar las hipótesis H_0 -MNI; por tanto se puede afirmar que el número de métodos no identificados en ambas muestras es diferente. Como se puede observar en la Figura 4-9, podemos afirmar que el número de métodos no identificados en el diagrama de clases donde los *patrones de producto* fueron utilizados es menor que en los proyectos donde no se utilizaron los *patrones de producto*.

D. Revisiones de los modelos software seleccionados hasta que alcanzan la calidad requerida (Objetivo 3). Otro aspecto importante es el número de revisiones de los productos hasta que alcancen el control de calidad establecido en cada modelo.

D.1. Fase de Análisis: los modelos realizados son la Especificación de Requisitos, el Diagrama de Casos de Uso, Casos de Uso alto nivel, Casos de Uso expandidos.

Las hipótesis nulas definidas son:

- H_0 -ERR: el número de revisiones realizadas en la especificación de requisitos de ambas muestras es igual.
- H_0 -UCR: el número de revisiones realizadas en el diagrama de casos de uso en ambas muestras es igual.
- H_0 -UCHR: el número de revisiones realizadas en los casos de uso de alto nivel en ambas muestras es igual.

- H0-UCER: el número de revisiones realizadas en los casos de uso expandidos en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de revisiones realizadas en la especificación de requisitos en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 2: número de revisiones realizadas en la especificación de requisitos en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 3: número de revisiones realizadas del diagrama de casos de uso en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 4: número de revisiones realizadas del diagrama de casos de uso en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 5: número de revisiones realizadas del caso de uso de alto nivel en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 6: número de revisiones realizadas del caso de uso de alto nivel en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 7: número de revisiones realizadas del caso de uso expandido en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 8: número de revisiones realizadas del caso de uso expandido en un proyecto donde los patrones de producto si fueron usados (muestra 2)

En la Tabla 4-16 se muestra un resumen de los resultados estadísticos y en la Figura 4-10 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Especificación de Requisitos		Diagrama de Casos de Uso		Casos de Uso Alto Nivel		Casos de Uso Expandidos	
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8
Recuento	10	10	10	10	10	10	10	10
Media	2.8	1.6	3.5	1.8	2.4	1.2	2.4	1.0
Moda	4	2	4	2	3	2	3	1
Varianza	1.06	0.26	1.16	0.44	0.93	0.62	0.93	0.66
Desviación Std	1.03	0.51	1.08	0.66	0.96	0.78	0.96	0.81
Mínimo	1	1	2	1	1	0	1	0
Máximo	4	2	5	3	4	2	4	2
Rango	3	1	3	2	3	2	3	2
Std. skewness	-0.35	0.62	0.0	0.0	-0.14	-0.52	-0.14	-0.0
Std. kurtosis	-0.57	-1.46	-0.66	0.05	-0.40	-0.69	-0.40	-0.89
T student	3.29		3.74		3.04		3.50	
P - Value	0.006		0,002		0.007		0.003	

Tabla 4-16 Resumen estadístico: Número de Revisiones – Fase de Análisis

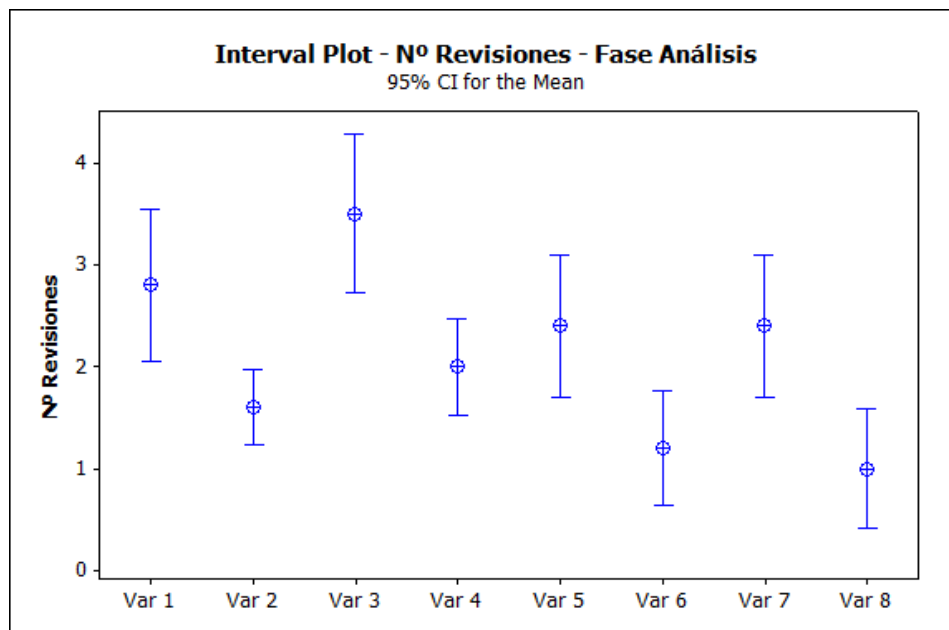


Figura 4-10 Interval Plot – Número de Revisiones – Fase Análisis

Con valores p-value de 0,006 para la especificación de requisitos; 0.002 para el diagrama de casos de uso; 0.007 para los casos de uso de alto nivel; 0.003 para los casos de uso expandidos todos ellos menores de 0,05; se puede rechazar las hipótesis H_0 -ERR, H_0 -UCR, H_0 -UCHR y H_0 -UCER respectivamente; por lo que se puede afirmar que el número de revisiones realizadas en los cuatro modelos de análisis en ambas muestras son diferentes. Como se puede observar en la Figura 4-10, se puede afirmar que el número de revisiones realizados en los proyectos donde los *patrones*

de producto fueron utilizados es menor que en los proyectos donde no se utilizaron los *patrones de producto*.

D.2. Fase de Diseño: los modelos realizados son el Diagrama de Clases, Diagrama de Secuencia y el Diagrama de Colaboración.

Las hipótesis nulas definidas son:

- H0-CLDR: número de revisiones realizadas en el diagrama de clases de ambas muestras es igual.
- H0-SDR: número de revisiones realizadas en el diagrama de secuencia en ambas muestras es igual.
- H0-CDR: número de revisiones realizadas en el diagrama de colaboración en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de revisiones realizadas en el diagrama de clases en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 2: número de revisiones realizadas en el diagrama de clases en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 3: número de revisiones realizadas en el diagrama de secuencia en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 4: número de revisiones realizadas en el diagrama de secuencia en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 5: número de revisiones realizadas en el diagrama de colaboración en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 6: número de revisiones realizadas en el diagrama de colaboración en un proyecto donde los patrones de producto si fueron usados (muestra 2)

En la Tabla 4-17 se muestra un resumen de los resultados estadísticos y en la Figura 4-11 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Diagrama de Clases		Diagrama de Secuencia		Diagrama de Colaboración	
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6
Recuento	10	10	10	10	10	10
Media	4.3	2.1	3.7	1.9	2	1.1
Moda	4	2	4	2	2	1
Varianza	0.9	0.98	1.34	0.54	0.66	0.1
Desviación Std	0.94	0.99	1.15	0.73	0.81	0.31
Mínimo	3	1	2	1	1	1
Máximo	6	4	5	3	3	2
Rango	3	3	3	2	2	1
Std. skewness	0.30	0.78	-0.44	0.21	0.0	4.08
Std. kurtosis	-0.22	-0.10	-0.79	-0.47	-0.89	6.45
T student	5.06		4.14		3.25	
P - Value	0.000		0.001		0.008	

Tabla 4-17 Resumen estadístico: Número de Revisiones – Fase de Diseño

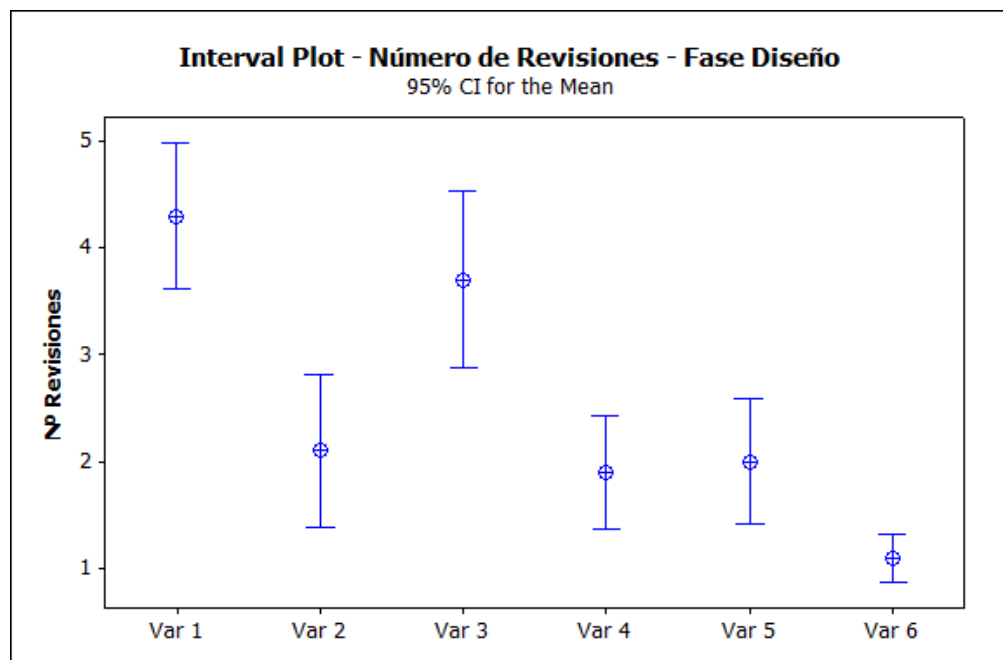


Figura 4-11 Interval Plot – Número de Revisiones – Fase Diseño

Con valores p-value de 0,000 para el diagrama de clases; 0.001 para el diagrama de secuencia y 0.008 para el diagrama de colaboración, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-CLDR, H0-SDR y H0-CDR respectivamente; por lo que se puede afirmar que el número de revisiones realizadas a los tres modelos de diseño en ambas muestras es diferente. Como se puede observar en la Figura 4-11, se puede afirmar que el número de revisiones realizados en los proyectos donde los patrones de producto fueron utilizados es menor que en los proyectos donde no se utilizaron los patrones de producto.

D.3. Fase de Gestión: los modelos realizados son la Estimación utilizando los Puntos de Función de Albretch, Estimación utilizando Cocomo II y la Planificación realizando un Diagrama Gantt.

Las hipótesis nulas definidas son:

- H0-AER: número de revisiones realizadas en la estimación utilizando los puntos de función Albretch de ambas muestras es igual.
- H0-CER: número de revisiones realizadas en la estimación utilizando Cocomo II en ambas muestras es igual.
- H0-GPR: número de revisiones realizadas en la planificación utilizando el diagrama Gantt en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de revisiones realizadas en la estimación usando los puntos de función en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 2: número de revisiones realizadas en la estimación usando los puntos de función en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 3: número de revisiones realizadas en la estimación usando Cocomo II en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 4: número de revisiones realizadas en la estimación usando Cocomo II en un proyecto donde los patrones de producto si fueron usados (muestra 2)
- Var 5: número de revisiones realizadas en la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto no fueron usados (muestra 1)
- Var 6: número de revisiones realizadas en la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto si fueron usados (muestra 2).

En la Tabla 4-18 se muestra un resumen de los resultados estadísticos y en la Figura 4-12 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Estimación Puntos de Función		Estimación Cocomo II		Planificación Diagrama Gantt	
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6
Recuento	10	10	10	10	10	10
Media	2.8	1.3	2.7	1.4	2.8	1.5
Moda	3	1	3	1	3	1
Varianza	0.62	0.23	0.67	0.26	0.17	0.27
Deviación Std	0.78	0.48	0.82	0.51	0.42	0.52
Mínimo	2	1	1	1	2	1
Máximo	4	2	4	2	3	2
Rango	2	1	3	1	1	1
Std. skewness	0.52	1.33631	-1.04	0.625	-2.29	0.0
Std. kurtosis	-0.69	-0.79	0.79	-1.46	0.90	-1.65
T student	5.13		4.23		6.09	
P - Value	0.000		0.001		0.000	

Tabla 4-18 Resumen estadístico: Número de Revisiones – Fase de Gestión

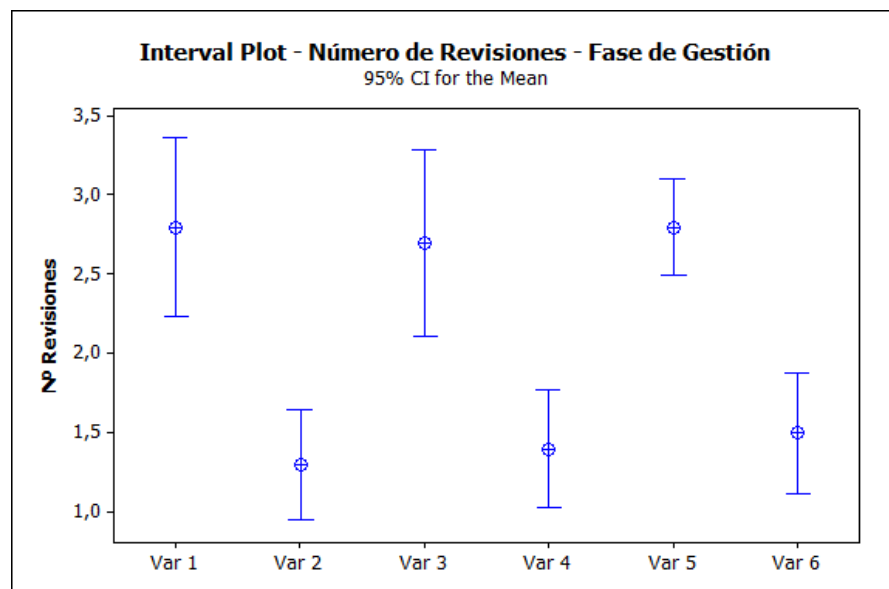


Figura 4-12 Interval Plot – Número de Revisiones – Fase de Gestión

Con valores p-value de 0,000 para la estimación en puntos de función; 0.001 para la estimación Cocomo II y 0.000 para el planificación mediante el diagrama Gantt, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-AER, H0-CER y H0-GPR respectivamente; por lo que se puede afirmar que el número de revisiones realizadas a los tres modelos de gestión en ambas muestras es diferente. Como se puede observar en la Figura 4-12, se puede afirmar que el número de revisiones realizados en los proyectos donde los *patrones de producto* fueron utilizados es menor que en los proyectos donde no se utilizaron los *patrones de producto*.

Fase III: Validación del Marco Metodológico a través del despliegue de la solución propuesta en esta tesis doctoral

Se han definido tres muestras independientes debido a que la observación de cada una de ellas no condiciona en nada a la observación de las otras:

- Muestra 1: representa los proyectos que se han desarrollado sin *patrones de producto* (**Fase I**).
- Muestra 2: representa los proyectos que se han desarrollado con *patrones de producto* implementados en la wiki (<http://productpatterns.sel.inf.uc3m.es>) (**Fase II**).
- Muestra 3: representa los proyectos que se han desarrollado con *patrones de producto* y el marco metodológico (**Fase III**).

A continuación se describe el análisis cuantitativo realizado en la Fase III de validación de cada una de las variables.

A. Eficiencia de Uso de los Procesos Software (Objetivo 3). Se han recogido los valores del tiempo que se ha tardado en desarrollar cada producto software. Para una mayor claridad, se analizará al igual que en la Fase II, la eficiencia de uso de los procesos software por fases del ciclo de vida (fase de análisis, fase de diseño y fase de gestión).

A.1. Fase de Análisis: los modelos realizados son la Especificación de Requisitos, el Diagrama de Casos de Uso, Casos de Uso alto nivel, Casos de Uso expandidos.

Las hipótesis nulas definidas para comprobar la eficiencia de uso de las actividades de la fase de análisis son:

- H0-ER: el tiempo de desarrollo para realizar la especificación de requisitos de ambas muestras es igual.
- H0-UC: el tiempo de desarrollo para realizar el diagrama de casos de uso en ambas muestras es igual.
- H0-UCH: el tiempo de desarrollo para realizar los casos de uso de alto nivel en ambas muestras es igual.
- H0-UCE: el tiempo de desarrollo para realizar los casos de uso expandidos en ambas muestras es igual.

Las variables definidas son:

- Var 1: tiempo (en hora) de desarrollo en realizar la especificación de requisitos en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: tiempo (en hora) de desarrollo en realizar la especificación de requisitos en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: tiempo (en hora) de desarrollo en realizar la especificación de requisitos en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 4: tiempo (en hora) de desarrollo en realizar el diagrama de casos de uso en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 5: tiempo (en hora) de desarrollo en realizar el diagrama de casos de uso en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 6: tiempo (en hora) de desarrollo en realizar el diagrama de casos de uso en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 7: tiempo (en hora) de desarrollo en realizar los casos de uso de alto nivel en un proyecto donde los patrones de producto ni el marco metodológico no fueron usados (muestra 1).
- Var 8: tiempo (en hora) de desarrollo en realizar los casos de uso de alto nivel en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 9: tiempo (en hora) de desarrollo en realizar los casos de uso de alto nivel en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 10: tiempo (en hora) de desarrollo en realizar los casos de uso expandido en un proyecto donde los patrones de producto ni el marco metodológico no fueron usados (muestra 1).
- Var 11: tiempo (en hora) de desarrollo en realizar los casos de uso expandido en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).

- Var 12: tiempo (en hora) de desarrollo en realizar los casos de uso expandido en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-19 se muestra un resumen de los resultados estadísticos y en la Figura 4-13 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Especificación de Requisitos			Diagrama de Casos de Uso			Casos de Uso Alto Nivel			Casos de Uso Expandidos		
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9	Var 10	Var 11	Var 12
Recuento	10	10	10	10	10	10	10	10	10	10	10	10
Media	8.5	10.8	7	10.6	7.1	5.2	4.9	2.6	1.15	4.3	2.5	1.1
Moda	8	11	7	10	7	5	4	2	1	4	2	1
Varianza	1.61	1.73	0.66	0.93	0.93	0.4	0.76	0.48	0.11	0.45	0.27	0.1
Desviación Std	1.26	1.31	0.81	0.96	0.96	0.63	0.87	0.69	0.33	0.67	0.52	0.31
Mínimo	7	8	6	9	6	4	4	2	1	3	2	1
Máximo	11	13	8	12	9	6	6	4	2	5	3	2
Rango	4	5	2	3	3	2	2	2	1	2	1	1
Std. skewness	1.05	-0.82	0.0	0.14	1.03	-0.17	0.28	1.00	2.93	-0.55	0.0	4.08
Std. kurtosis	0.16	1.31	-0.89	-0.41	0.03	0.11	-1.11	-0.09	3.07	-0.18	-1.65	6.45
T student	7.76			5.20			5.91			7.20		
P - Value	0.000			0.000			0.000			0.000		

Tabla 4-19 Resumen estadístico: Eficiencia de Uso – Fase de Análisis

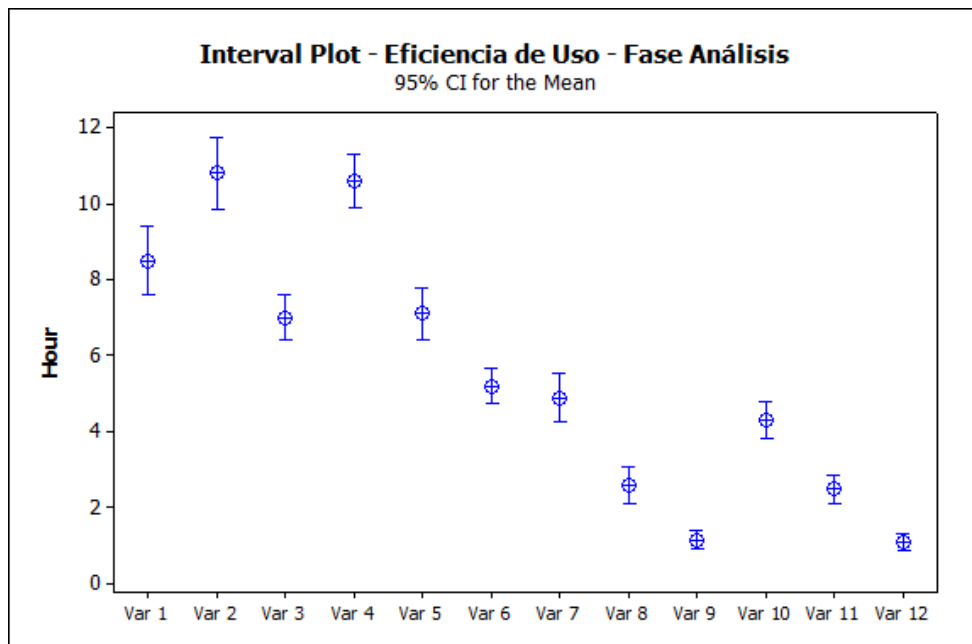


Figura 4-13 Interval Plot – Eficiencia de Uso – Fase de Análisis

Con valores p-value de 0,000 para la especificación de requisitos, el diagrama de casos de uso, los casos de uso de alto nivel y los casos de uso expandidos, todos ellos menores de 0,05; se puede rechazar las hipótesis H_0 -ER, H_0 -UC, H_0 -UCH y H_0 -UCE respectivamente; por lo que se afirma que el tiempo de desarrollo en realizar los cuatro actividades de análisis en las tres muestras es diferente. Como se puede observar en la Figura 4-13, podemos afirmar que el tiempo de desarrollo en las actividades donde los patrones de producto y el marco metodológico fueron utilizados es menor que en los proyectos donde se utilizaron únicamente los *patrones de producto* y en los proyectos donde ni los patrones de producto ni el marco metodológico fueron utilizados.

A.2. Fase de Diseño: los modelos realizados son el Diagrama de Clases, Diagrama de Secuencia y el Diagrama de Colaboración.

Las hipótesis nulas definidas para comprobar la eficiencia de uso de las actividades de la fase de diseño son:

- H_0 -CLD: el tiempo de desarrollo para realizar el diagrama de clases de ambas muestras es igual.
- H_0 -SD: el tiempo de desarrollo para realizar el diagrama de secuencia en ambas muestras es igual.

- H0-CD: el tiempo de desarrollo para realizar el diagrama de colaboración en ambas muestras es igual.

Las variables definidas son:

- Var 1: tiempo (en hora) de desarrollo en el diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: tiempo (en hora) de desarrollo en el diagrama de clases en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: tiempo (en hora) de desarrollo en el diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 4: tiempo (en hora) de desarrollo del diagrama de secuencia en un proyecto donde los patrones de producto ni el marco metodológico no fueron usados (muestra 1).
- Var 5: tiempo (en hora) de desarrollo del diagrama de secuencia en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 6: tiempo (en hora) de desarrollo del diagrama de secuencia en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 7: tiempo (en hora) de desarrollo del diagrama de colaboración en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 8: tiempo (en hora) de desarrollo del diagrama de colaboración en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 9: tiempo (en hora) de desarrollo del diagrama de colaboración en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-20 se muestra un resumen de los resultados estadísticos y en la Figura 4-14 se muestra la gráfica “interval plot” que proporciona los intervalos de

confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Diagrama de Clases			Diagrama de Secuencia			Diagrama de Colaboración		
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9
Recuento	10	10	10	10	10	10	10	10	10
Media	16.9	20.3	13.7	13.3	10.6	7.35	12	9.2	7.4
Moda	19	18	13	14	11	8	12	9	7
Varianza	3.65	6.67	2.45	1.56	1.6	1.22	0.66	0.84	0.71
Desviación Std	1.91	2.58	1.56	1.25	1.26	1.11	0.81	0.91	0.84
Mínimo	14	17	11	11	9	5	11	8	6
Máximo	19	24	16	15	12	8.5	13	11	9
Rango	5	7	5	4	3	3.5	2	3	3
Std. skewness	-0.23	0.21	-0.31	-0.91	-0.36	-1.67	0.0	0.77	0.51
Std. kurtosis	-1.12	-1.19	-0.49	-0.31	-1.07	0.62	-0.89	0.25	0.23
T student	6,91			6.11			4.56		
P - Value	0.000			0.000			0.000		

Tabla 4-20 Resumen estadístico: Eficiencia de Uso – Fase de Diseño

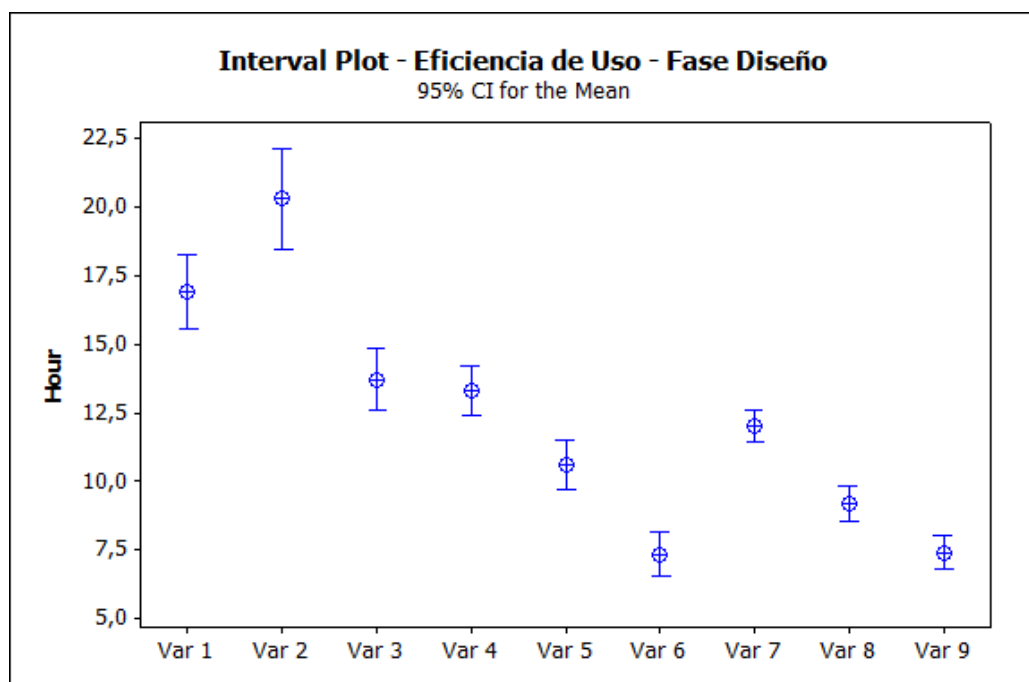


Figura 4-14 Interval Plot: Eficiencia de Uso – Fase de Diseño

Con valores p-value de 0,000 para el diagrama de clases, el diagrama de secuencia y el diagrama de colaboración, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-CLD, H0-SD y H0-CD respectivamente; por lo que se

afirma que el tiempo de desarrollo en realizar los tres modelos de diseño en las tres muestras es diferente. Como se puede observar en la Figura 4-14, podemos afirmar que el tiempo de desarrollo en los proyectos donde los *patrones producto* y el marco metodológico fueron utilizados es menor que en los proyectos donde no se utilizaron.

A.3. Fase de Gestión: los modelos realizados son la Estimación utilizando los Puntos de Función de Albretch, Estimación utilizando Cocomo II y la Planificación realizando un Diagrama Gantt.

Las hipótesis nulas definidas para comprobar la eficiencia de uso de las actividades de la fase de gestión son:

- H0-AE: el tiempo de desarrollo para realizar la estimación utilizando los puntos de función Albretch de ambas muestras es igual.
- H0-CE: el tiempo de desarrollo para realizar la estimación utilizando Cocomo II en ambas muestras es igual.
- H0-GP: el tiempo de desarrollo para realizar la planificación utilizando el diagrama Gantt en ambas muestras es igual.

Las variables definidas son:

- Var 1: tiempo (en hora) de desarrollo en realizar la estimación usando los puntos de función en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: tiempo (en hora) de desarrollo en realizar la estimación usando los puntos de función en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: tiempo (en hora) de desarrollo en realizar la estimación usando los puntos de función en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 4: tiempo (en hora) de desarrollo en realizar la estimación usando Cocomo II en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1)
- Var 5: tiempo (en hora) de desarrollo en realizar la estimación usando Cocomo II en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).

- Var 6: tiempo (en hora) de desarrollo en realizar la estimación usando Cocomo II en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 7: tiempo (en hora) de desarrollo en realizar la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 8: tiempo (en hora) de desarrollo en realizar la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 9: tiempo (en hora) de desarrollo en realizar la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-21 se muestra un resumen de los resultados estadísticos y en la Figura 4-15 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Estimación Puntos de Función			Estimación Cocomo II			Planificación Diagrama Gantt		
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9
Count	10	10	10	10	10	10	10	10	10
Average	9.1	6.9	5.2	9.2	6.6	5.2	7.2	4.7	3.4
Mode	9	6	6	9	6	5	8	5	4
Variance	0.98	0.989	1.06	1.06	0.48	0.84	1.06	0.45	0.4889
Std Deviation	0.01	0.99	1.03	1.03	0.69	0.91	1.03	0.67	0.69
Minimum	8	6	4	8	6	4	5	4	2
Maximum	11	8	7	11	8	7	8	6	4
Range	3	2	3	3	2	3	3	2	2
Std. skewness	0.78	0.31	0.35	0.35	1.01	0.77	-1.61	0.55	-1.01
Std. kurtosis	-0.11	-1.48	-0.57	-0.57	-0.09	0.25	0.61	-0.18	-0.09
T student	3.75			3.83			4.23		
P - Value	0.002			0.001			0.001		

Tabla 4-21 Resumen estadístico: Eficiencia de Uso – Fase de Gestión

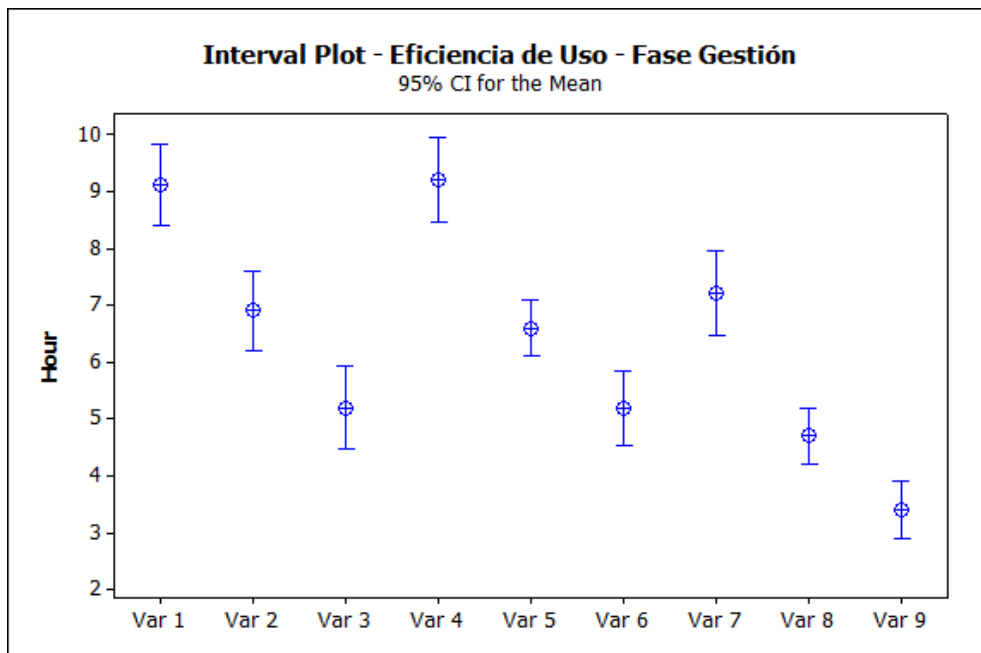


Figura 4-15 Interval Plot – Eficiencia de Uso – Fase de Gestión

Con valores p-value de 0,002 para la estimación mediante puntos de función; 0.001 para la estimación mediante COCOMO II y la planificación mediante Diagrama Gantt, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-AE, H0-CE y H0-GP respectivamente; por lo que se afirma que el tiempo de desarrollo en realizar los tres modelos de gestión en las tres muestras es diferente. Como se puede observar en la Figura 4-15, podemos afirmar que el tiempo de desarrollo en las actividades donde los *patrones de producto* y el marco metodológico fueron utilizados es menor que en el resto.

B. Volatilidad de los Requisitos. Es muy importante determinar el grado de volatilidad de los requisitos, este parámetro se va a medir mediante el número de requisitos que son identificados en la especificación de requisitos pero no son satisfecho en el diagrama de casos de uso. Para obtener esta información, la descripción del caso de uso tiene un campo donde se indica qué requisitos es satisfecho en ese caso de uso.

La hipótesis nula definida en esta variable es:

- H0-MR: el número de requisitos perdidos en el diagrama de casos de uso en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de requisitos perdidos en el diagrama de casos de uso en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: número de requisitos perdidos en el diagrama de casos de uso en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: número de requisitos perdidos en el diagrama de casos de uso en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-22 se muestra un resumen de los resultados estadísticos y en la Figura 4-16 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Número de requisitos perdidos		
	Var 1	Var 2	Var 3
Recuento	10	10	10
Media	8	5	3.3
Moda	9	5	3
Varianza	1.11	1.77	0.67
Desviación Std	1.05	1.33	0.82
Mínimo	6	3	2
Máximo	9	7	5
Rango	3	4	3
Std. skewness	-0.91	-0.45	1.04
Std. kurtosis	-0.29	-0.48	0.79
T student	3.43		
P – Value	0.004		

Tabla 4-22 Resumen estadístico: Volatilidad de Requisitos

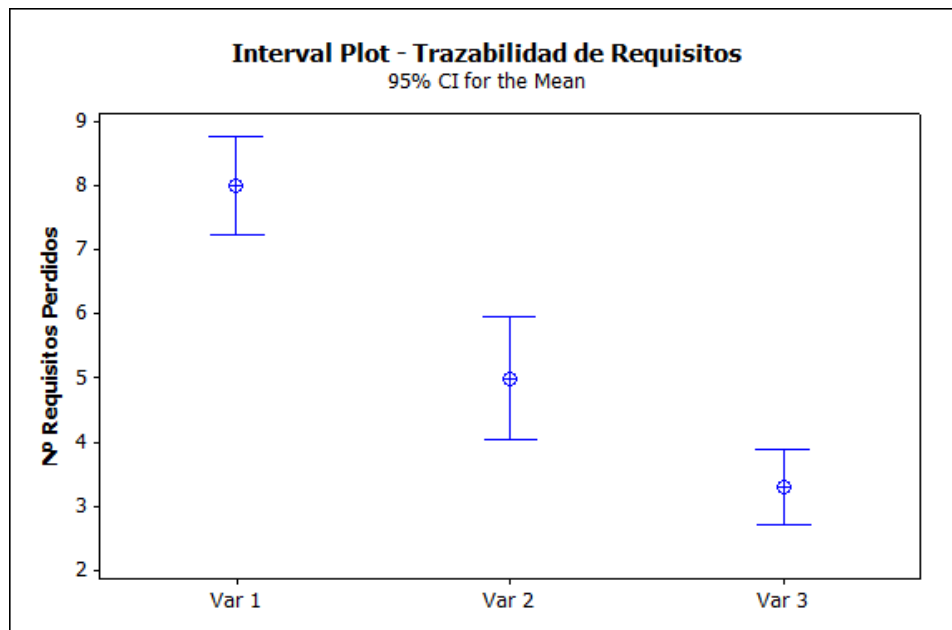


Figura 4-16 Interval Plot – Volatilidad de Requisitos

Con p-value de 0,004 menor de 0,05; significa que se puede rechazar las hipótesis H0-MR; y por tanto se puede afirmar que el número de requisitos perdidos en el diagrama de casos de uso en las tres muestras es diferente. Como se puede observar en la Figura 4-16, podemos afirmar que el número de requisitos perdidos en el diagrama de casos de uso donde los patrones de producto y el marco metodológico fueron utilizados.

C. Clases y métodos no identificados. Otro parámetro que se analizó fue el número de clases y métodos no identificados en la primera versión de los diagramas de clase de cada proyecto.

C.1. Número de Clases no identificadas: se recogió el número de clases no identificadas en la primera versión de los diagramas de clase desarrollados en los 10 proyectos software.

La hipótesis nula definida en esta variable es:

- H0-CNI: el número de clases no identificadas en la primera versión del diagrama de clases en ambas muestras es el mismo.

Las variables definidas son:

- Var 1: número de clases del diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).

- Var 2: número de clases del diagrama de clases en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: número de clases del diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-23 se muestra un resumen de los resultados estadísticos y en la Figura 4-17 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Número de Clases No Identificadas		
	Var 1	Var 2	Var 3
Recuento	10	10	10
Media	6.0	3.1	1.7
Moda	6	4	2
Varianza	2.66	1.43	0.45
Desviación Std	1.63	1.19	0.67
Mínimo	4.0	1.0	1.0
Máximo	9.0	5.0	3.0
Rango	5.0	4.0	2.0
Std. skewness	0.74	-0.31	0.55
Std. kurtosis	-0.18	-0.23	-0.18
T student	3.22		
P – Value	0.006		

Tabla 4-23 Resumen estadístico: Número de Clases No Identificadas

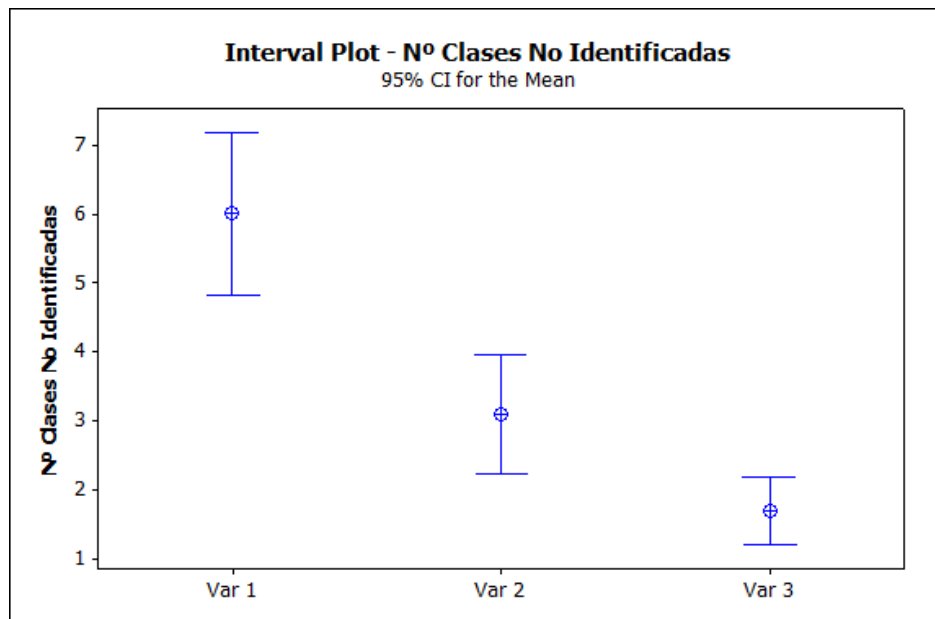


Figura 4-17 Interval Plot – Número de Clases No Identificados

Con p-value de 0,006 menor de 0,05 se puede rechazar las hipótesis H_0 -CNI; por tanto se puede afirmar que el número de clases no identificadas en las tres muestras es diferente. Como se puede observar en la Figura 4-17, podemos afirmar que el número de clases no identificadas en el diagrama de clases donde los *patrones de producto* y el marco metodológico fueron utilizados es menor.

C.2. Número de Métodos no identificados: se recogió el número de métodos no identificados en la primera versión de los diagramas de clase desarrollados en los 10 proyectos software.

La hipótesis nula definida en esta variable es:

- H_0 -MNI: el número de métodos no identificados en la primera versión del diagrama de clases en ambas muestras es el mismo.

Las variables definidas son:

- Var 1: número de métodos del diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: número de métodos del diagrama de clases en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: número de métodos del diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico fueron usados (muestra 3).

En la Tabla 4-24 se muestra un resumen de los resultados estadísticos y en la Figura 4-17 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Número de Métodos No Identificados		
	Var 1	Var 2	Var 3
Recuento	10	10	10
Media	11.2	5.5	3.6
Moda	9	5	3
Varianza	9.51	1.61	1.37
Desviación Std	3.08	1.26	1.17
Mínimo	6	4	2
Máximo	16	8	6
Rango	10	4	4
Std. skewness	-0.08	1.05	0.61
Std. kurtosis	-0.29	0.16	0.66
T student	3.48		
P – Value	0.003		

Tabla 4-24 Resumen estadístico: Número de Métodos No Identificados

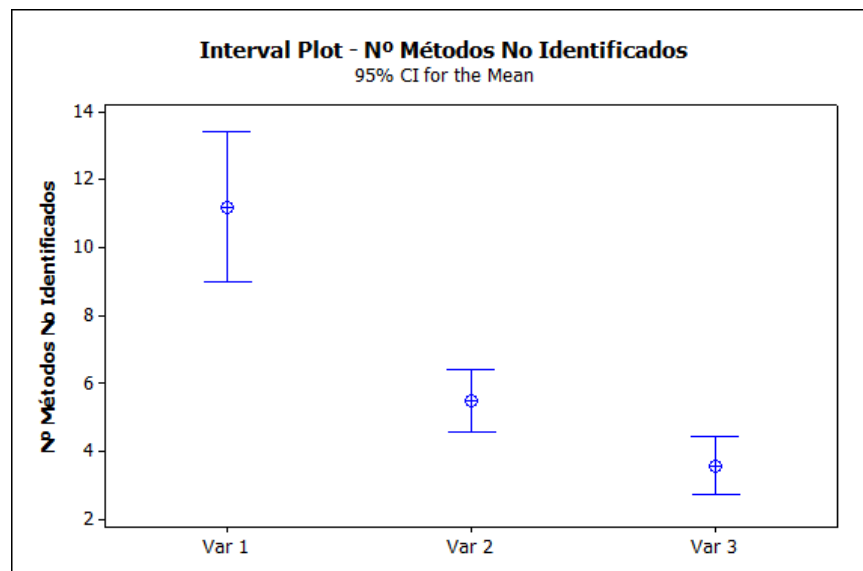


Figura 4-17 Interval Plot – Número de Métodos No Identificados

Con p-value de 0,003 menor de 0,05 se puede rechazar las hipótesis H0-MNI; por tanto se puede afirmar que el número de métodos no identificados en las tres muestras es diferente. Como se puede observar en la Figura 4-17, podemos afirmar

que el número de métodos no identificados en el diagrama de clases donde los *patrones de producto* y el marco metodológico fueron utilizados es menor.

D. Revisiones de los modelos software seleccionados hasta que alcanzan la calidad requerida (Objetivo 3). Otro aspecto importante es el número de revisiones de los productos hasta que alcancen el control de calidad establecido en cada modelo.

D.1. Fase de Análisis: los modelos realizados son la Especificación de Requisitos, el Diagrama de Casos de Uso, Casos de Uso alto nivel, Casos de Uso expandidos.

Las hipótesis nulas definidas son:

- H0-ERR: el número de revisiones realizadas en la especificación de requisitos de ambas muestras es igual.
- H0-UCR: el número de revisiones realizadas en el diagrama de casos de uso en ambas muestras es igual.
- H0-UCHR: el número de revisiones realizadas en los casos de uso de alto nivel en ambas muestras es igual.
- H0-UCER: el número de revisiones realizadas en los casos de uso expandidos en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de revisiones realizadas en la especificación de requisitos en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: número de revisiones realizadas en la especificación de requisitos en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: número de revisiones realizadas en la especificación de requisitos en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 4: número de revisiones realizadas del diagrama de casos de uso en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).

- Var 5: número de revisiones realizadas del diagrama de casos de uso en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 6: número de revisiones realizadas del diagrama de casos de uso en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 7: número de revisiones realizadas del caso de uso de alto nivel en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 8: número de revisiones realizadas del caso de uso de alto nivel en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 9: número de revisiones realizadas del caso de uso de alto nivel en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 10: número de revisiones realizadas del caso de uso expandido en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 11: número de revisiones realizadas del caso de uso expandido en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 12: número de revisiones realizadas del caso de uso expandido en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-25 se muestra un resumen de los resultados estadísticos y en la Figura 4-23 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Especificación de Requisitos			Diagrama de Casos de Uso			Casos de Uso Alto Nivel			Casos de Uso Expandidos		
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9	Var 10	Var 11	Var 12
Recuento	10	10	10	10	10	10	10	10	10	10	10	10
Media	2,8	1.6	0.7	3,5	2	1	2.4	1.2	0.3	2.4	1.0	0.2
Moda	3	2	1	3	2	1	2	1	1	1	0	0
Varianza	1.06	0.26	0.23	1.16	0.44	0.44	0.93	0.62	0.23	0.93	0.66	0.17
Desviación Std	1.03	0.51	0.48	1.08	0.66	0.66	0.96	0.78	4,48	0.96	0.81	0.42
Mínimo	1	1	0.0	2	1	0.0	1	0	0	1	0	0
Máximo	4	2	1	5	3	2	4	2	1	4	2	1
Rango	3	1	1	3	2	2	3	2	1	3	2	1
Std. skewness	-0.35	-0.62	-1.33	0.0	0.0	0.0	-0.14	-0.52	1.33	-0.14	0.0	2.29
Std. kurtosis	-0.57	-1.3	-0.79	-0.66	0.05	0.05	-0.40	-0.69	-0.79	-0.41	-0.89	0.91
T student	4.02			3.35			3.08			2.75		
P – Value	0.001			0.004			0.008			0.0016		

Tabla 4-25 Resumen estadístico: Número de Revisiones – Fase de Análisis

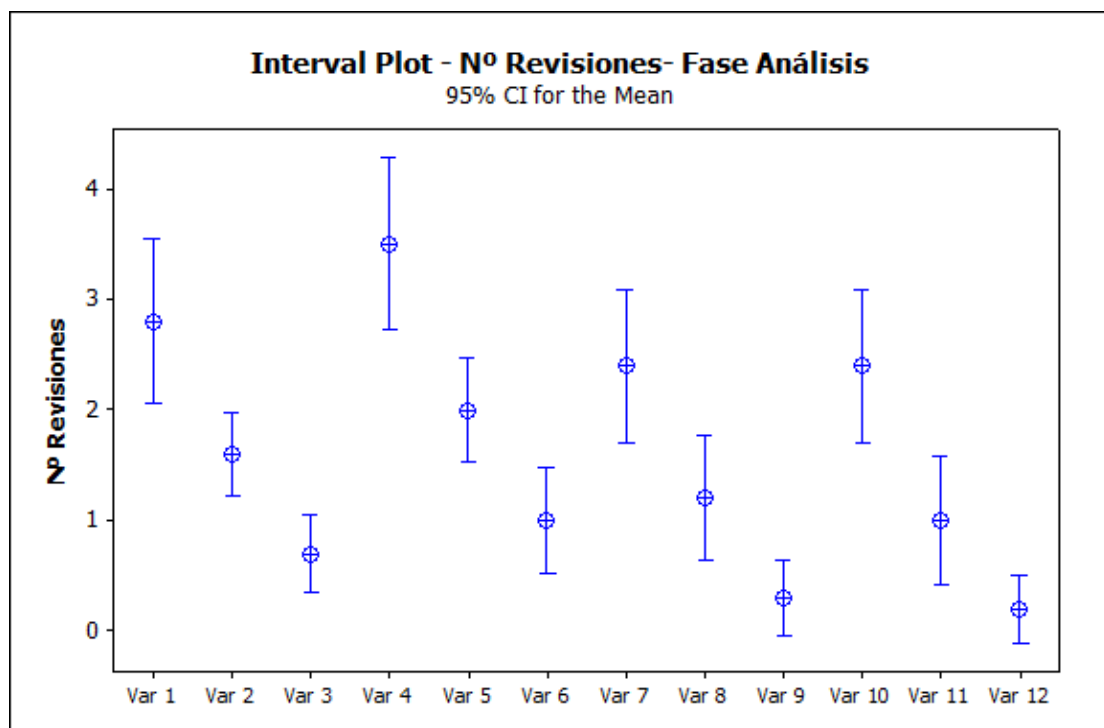


Figura 4-18 Interval Plot – Número de Revisiones- Fase Análisis

Con valores p-value de 0,001 para la especificación de requisitos; 0.004 para el diagrama de casos de uso; 0.008 para los casos de uso de alto nivel; 0.0016 para los casos de uso expandidos todos ellos menores de 0,05; se puede rechazar las hipótesis H_0 -ERR, H_0 -UCR, H_0 -UCHR y H_0 -UCER respectivamente; por lo que se puede

afirmar que el número de revisiones realizadas en las cuatro actividades de análisis en las tres muestras son diferentes. Como se puede observar en la Figura 4-18, se puede afirmar que el número de revisiones realizados en los proyectos donde los patrones de producto y el marco metodológico fueron utilizados es menor que en el resto de los proyectos.

D.2. Fase de Diseño: los modelos realizados son el Diagrama de Clases, Diagrama de Secuencia y el Diagrama de Colaboración.

Las hipótesis nulas definidas son:

- H0-CLDR: número de revisiones realizadas en el diagrama de clases de ambas muestras es igual.
- H0-SDR: número de revisiones realizadas en el diagrama de secuencia en ambas muestras es igual.
- H0-CDR: número de revisiones realizadas en el diagrama de colaboración en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de revisiones realizadas en el diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: número de revisiones realizadas en el diagrama de clases en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: número de revisiones realizadas en el diagrama de clases en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 4: número de revisiones realizadas en el diagrama de secuencia en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 5: número de revisiones realizadas en el diagrama de secuencia en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).

- Var 6: número de revisiones realizadas en el diagrama de secuencia en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 7: número de revisiones realizadas en el diagrama de colaboración en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 8: número de revisiones realizadas en el diagrama de colaboración en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 9: número de revisiones realizadas en el diagrama de colaboración en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-26 se muestra un resumen de los resultados estadísticos y en la Figura 4-19 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Diagrama de Clases			Diagrama de Secuencia			Diagrama de Colaboración		
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9
Recuento	10	10	10	10	10	10	10	10	10
Media	4.3	2.1	0.8	3.7	1.9	1.1	2.0	1.1	0.6
Moda	4	2	1	4	2	1	2	1	1
Varianza	0.9	0.98	0.4	1.34	0.54	0.1	0.66	0.1	0.26
Desviación Std	0.94	0.99	0.63	1.15	0.73	0.31	0.81	0.31	0.51
Mínimo	3	1	0.0	2	1	1	1	1	0.0
Máximo	6	4	2	5	3	2	3	2	1
Rango	3	3	2	3	2	1	2	1	1
Std. skewness	0.31	0.78	0.17	-0.44	0.21	4.08	0.0	4.08	-0.62
Std. kurtosis	-0.22	-0.11	0.11	-0.79	-0.47	6.45	-0.89	6.45	-1.46
T student	3,49			3,15			2,61		
P – Value	0.003			0.008			0,021		

Tabla 4-26 Resumen estadístico: Número de Revisiones – Fase de Diseño

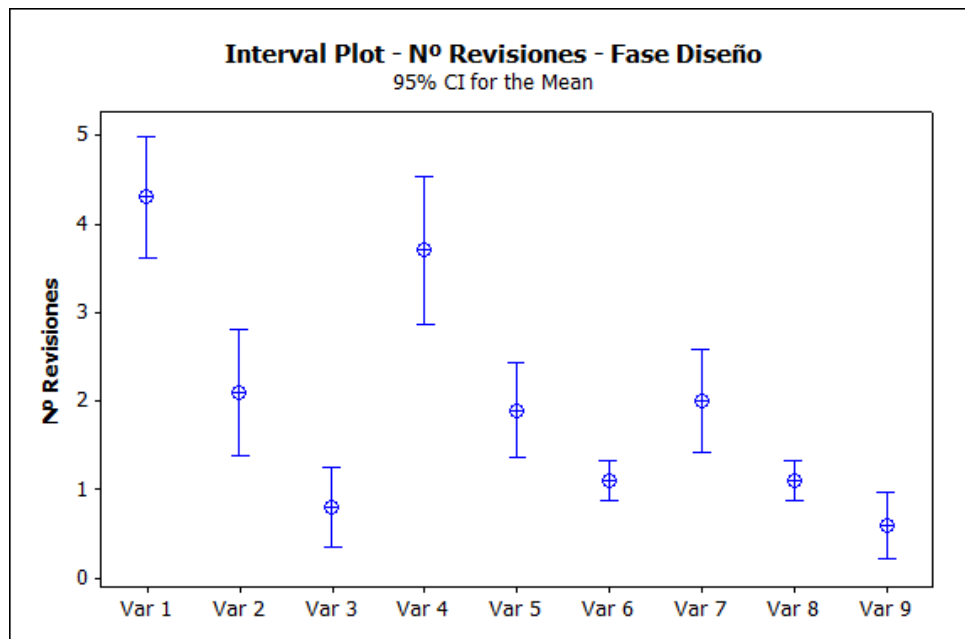


Figura 4-19 Interval Plot – Número de Revisiones- Fase de Diseño

Con valores p-value de 0,003 para el diagrama de clases; 0.008 para el diagrama de secuencia y 0.021 para el diagrama de colaboración, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-CLDR, H0-SDR y H0-CDR respectivamente; por lo que se puede afirmar que el número de revisiones realizadas a los tres modelos de diseño en las tres muestras son diferentes. Como se puede observar en la Figura 4-19, se puede afirmar que el número de revisiones realizadas en los proyectos donde los patrones de *producto* y el marco metodológico fueron utilizados es menor que en el resto de proyectos.

D.3. Fase de Gestión: los modelos realizados son la Estimación utilizando los Puntos de Función de Albretch, Estimación utilizando Cocomo II y la Planificación realizando un Diagrama Gantt.

Las hipótesis nulas definidas son:

- H0-AER: número de revisiones realizadas en la estimación utilizando los puntos de función Albretch de ambas muestras es igual.
- H0-CER: número de revisiones realizadas en la estimación utilizando Cocomo II en ambas muestras es igual.
- H0-GPR: número de revisiones realizadas en la planificación utilizando el diagrama Gantt en ambas muestras es igual.

Las variables definidas son:

- Var 1: número de revisiones realizadas en la estimación usando los puntos de función en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 2: número de revisiones realizadas en la estimación usando los puntos de función en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 3: número de revisiones realizadas en la estimación usando los puntos de función en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 4: número de revisiones realizadas en la estimación usando Cocomo II en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1)
- Var 5: número de revisiones realizadas en la estimación usando Cocomo II en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 6: número de revisiones realizadas en la estimación usando Cocomo II en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).
- Var 7: número de revisiones realizadas en la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto y el marco metodológico no fueron usados (muestra 1).
- Var 8: número de revisiones realizadas en la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto si fueron usados pero no fue utilizado el marco metodológico (muestra 2).
- Var 9: número de revisiones realizadas en la planificación mediante un diagrama Gantt en un proyecto donde los patrones de producto y el marco metodológico si fueron usados (muestra 3).

En la Tabla 4-27 se muestra un resumen de los resultados estadísticos y en la Figura 4-20 se muestra la gráfica “interval plot” que proporciona los intervalos de confianza para la media de la muestra, en la que se puede observar la mediana de cada variable definida.

	Estimación Puntos de Función			Estimación Cocomo II			Planificación Diagrama Gantt		
	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8	Var 9
Recuento	10	10	10	10	10	10	10	10	10
Media	2.8	1.3	0.5	2.7	1.4	0.7	2.8	1.5	0.8
Moda	3	1	1	3	1	1	3	1	1
Varianza	0.62	0.23	0.27	0.67	0.26	0.45	0.17	0.27	0.4
Desviación Std	0.78	0.48	0.52	0.82	0.51	0.67	0.42	0.52	0.63
Mínimo	2	1	0.0	1	1	0.0	2	1	0
Máximo	4	2	1	4	2	2	3	2	2
Rango	2	1	1	3	1	2	1	1	2
Std. skewness	0.52	1.33	0.0	-1.04	0.62	0.55	-2.29	0.0	-1.65
Std. kurtosis	-0.69	-0.79	-1.65	0.79	-1.46	-0.18	0.90	0.17	0.11
T student	3,54			2.60			2.69		
P – Value	0.003			0.019			0.016		

Tabla 4-27 Resumen estadístico: Número de Revisiones – Fase de Gestión

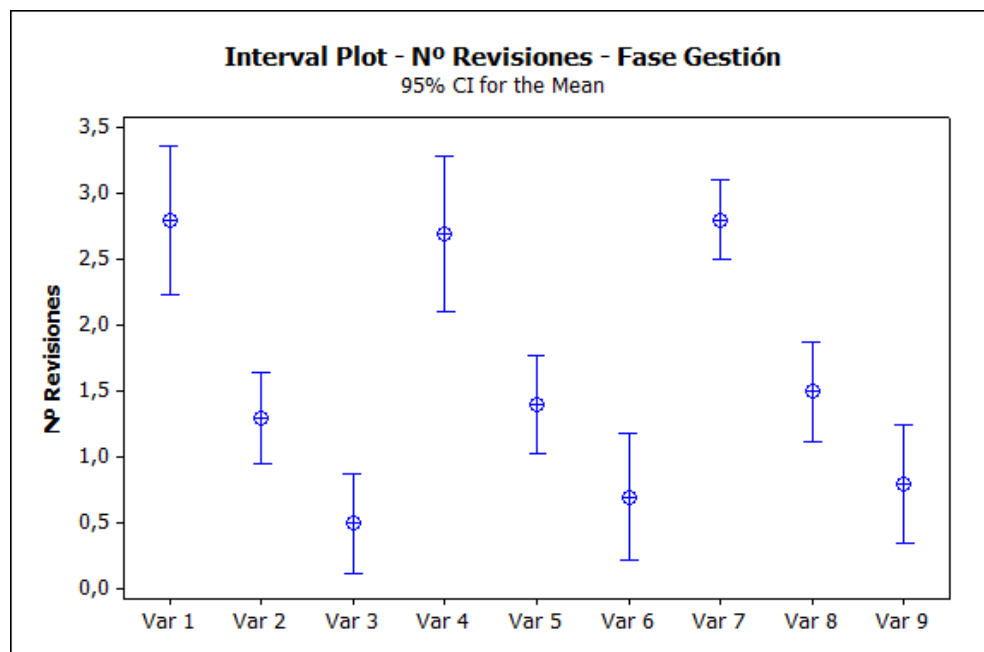


Figura 4-20 Interval Plot – Número de Revisiones- Fase de Diseño

Con valores p-value de 0,003 para la estimación por puntos de función; 0.019 para la estimación con Cocomo II y 0.016 para la planificación diagrama Gantt, todos ellos menores de 0,05; se puede rechazar las hipótesis H0-AER, H0-CER y H0-GPR respectivamente; por lo que se puede afirmar que el número de revisiones realizadas a los tres modelos de gestión en las tres muestras son diferente. Como se puede observar en la Figura 4-20, se puede afirmar que el número de revisiones

realizados en los proyectos donde los *patrones de producto* y el marco metodológico fueron utilizados es menor que en el resto de los proyectos.

4.4.2 Conclusiones extraídas del análisis Cuantitativo

Del análisis cuantitativo se puede afirmar que:

- Los *patrones de producto* aportan al equipo de desarrollo el conocimiento de las mejores prácticas de la ingeniería del software y la experiencia de los ingenieros expertos de proyectos previos reduciendo el tiempo de desarrollo de los proyectos, mejorando la eficiencia de uso de los procesos software y la calidad de los productos obtenidos (**Hipótesis H1.1**).
- El marco metodológico que incluye la utilización de los *patrones de producto* y la estrategia corporativa, a través de la plataforma colaborativa, facilita la difusión, uso y reutilización de los activos de procesos y proyectos en las organizaciones mejorando la productividad (eficiencia de uso) de los proyectos software definido en esta tesis doctoral (**Hipótesis H1.2**).

4.4.3 Análisis cualitativo

El análisis de las variables cualitativas se ha realizado a través de los cuestionarios. El objetivo del análisis cualitativo es: comprobar la satisfacción de los ingenieros software respecto a los *patrones de producto* y al marco metodológico; se va a comprobar si se mejora el entendimiento y el aprendizaje del proceso y la comunicación entre el equipo de trabajo en el desarrollo de los proyectos software (Tabla 4-28).

		Cuestionarios Validación Fase II Sub Fase I (1ª Versión <i>Patrones de Producto</i>)	Cuestionarios Validación Fase II Sub Fase II (2ª Versión <i>Patrones de Producto</i>)	Cuestionarios Validación Fase III
1	Los patrones de producto proporcionan el conocimiento para desarrollar las actividades de un proyecto software. 1: muy poco; 2:poco; 3: normal; 4: bastante 5:mucho.	4.11	4.22	4.17
2	He requerido conocimiento/información que no proporcionaba el <i>patrón de producto</i> . 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	1.56	1.11	1
3	Valore los siguientes campos: 1: muy poco valorado; 2:poco; 3: normal; 4: bastante 5: mucho.			
	Solution	4.44	4.44	4.06
	Template	3.67	4	3.78
	Lessons Learned	3.33	3.78	3.94
	Information Resource		2.89	2.78
4	Considera que sería más usable tener on-line, via web, los patrones de producto.	4.67	4.89	4.83
5	Considera útil las horas de formación dedicadas al training de patrones de producto 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	4.44	4.56	4.28
6	Le facilitó el <i>patrón de producto</i> reutilizar conocimiento de otros proyectos 1: nada; 2:poco; 3:normal; 4:bastante; 5: mucho	4	4.44	4.33
7	Conteste si usted realizó el rol de jefe de Proyecto: La elección del <i>patrón de producto</i> me resultó difícil 1: nada; 2:poco; 3:normal; 4: bastante difícil 5: muy difícil	3	3	3.17
8	Conteste si usted realizó el rol de jefe de Proyecto: A la hora de realizar la planificación en un diagrama Gantt, le facilitó el campo solution (development time) el cálculo del tiempo estimado que se tarda en desarrollar una actividad con el <i>patrón de producto</i> . 1: nada; 2:poco; 3:normal; 4: bastante 5: mucho	4.33	5	5
9	Conteste si usted realizó el rol de jefe de Proyecto: Valore si el campo "entries" le sirvió para establecer las precedencias entre los patrones de producto y por tanto entre actividades 1: nada; 2:poco; 3:normal; 4: bastante 5: mucho	4	4.33	4.33
10	Considera que los <i>patrones de producto</i> facilitan el entendimiento del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	4.68	4.72	4.75
11	Considera que los <i>patrones de producto</i> facilitan el aprendizaje del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	4.58	4.49	4.81
12	Considera que la estrategia corporativa facilita el entendimiento del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	4.24	4.37	4.77
13	Considera que la estrategia corporativa facilita el aprendizaje del proceso. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	4.11	4.27	4.43
14	Considera que la estrategia corporativa junto con la herramienta colaborativa facilita la transferencia del conocimiento. 1: nada 2: muy poco; 3: algo 4: bastante 5:mucho	4.73	4.81	4.96

Tabla 4 -28 Resumen de las respuestas al Cuestionario 2 y 3

4.4.4 Conclusiones Extraídas del análisis cualitativo

Del resumen de las respuestas realizadas por los ingenieros software al Cuestionario 2, se pueden extraer las siguientes conclusiones.

- En general, se puede afirmar que los patrones de producto han sido muy bien evaluados, con una media de 4.11 sobre 5.

- De media, el campo “solution” fue el campo mejor valorado por los ingenieros software, 4.44 sobre 5. Este campo proporciona los pasos que los ingenieros software tienen que realizar para obtener el producto software deseado. El campo “Plantillas” fue el segundo campo mejor valorado (3.67 sobre 5), seguido por el campo “lessons learned” (3.33 sobre 5), estos campos ayudan a los ingenieros software a ser más eficientes.

- Haciendo un estudio entre la relación existente entre la experiencia de cada ingeniero software con los valores asignados a los campos del *patrón de producto*, se pueden extraer las siguientes conclusiones:

- Para los ingenieros software con menos experiencia, la valoración de los campos de mayor a menor fue la siguiente: “solucion”, “plantilla”, “ejemplos” y “referencias”.
- Para los ingenieros software con mayor experiencia, los campos más valorados son: “plantillas”, “ejemplos” y “lecciones aprendidas”.

- De los comentarios aportados por los ingenieros software en la afirmación 2, se observó que algunos ingenieros software con menos experiencia, que participaron en la validación de los *patrones de producto* en la Fase II- Sub Fase I en los Proyectos 1, 2 y 3, sugirieron la incorporación de conocimiento básico sobre la actividad en la representación del *patrón de producto*, para que les ayudara en el desarrollo de los pasos del diagrama de actividad del campo “Solución”. Por lo que este campo llamado “conocimiento base” fue incorporado en la segunda versión de los *patrones de producto*, los cuales fueron validados por el resto de los proyectos desarrollados en la Fase II- Sub Fase II. En la Fase II-Sub Fase II, los ingenieros software no hicieron aportaciones referentes a los campos del *patrón de producto*.

- En general, se puede afirmar que los *patrones de producto* permiten la reutilización del conocimiento en los proyectos software, 4,67 sobre 5 ingenieros software afirmaron que los *patrones de producto* deben ser accesibles.

- Las conclusiones que se han obtenido de las respuestas de los ingenieros software que ejercieron el rol de jefe de proyectos fueron las siguientes:

- El grado de dificultad para seleccionar el *patrón de producto* apropiado para utilizar en una tarea es media y se reduce el grado de dificultad mediante usan más veces los patrones de producto.
- El impacto que los patrones de producto tienen en el desarrollo de un diagrama Gantt es muy positivo. Los jefes de proyecto afirmaron que la información que les proporciona del tiempo estimado en realizar la tarea les sirve para la planificación del proyecto.

- Se puede afirmar que el marco metodológico ha sido bastante bien evaluado, con una media de 4.06 sobre 5.

- La valoración de las horas de formación fueron muy positivas 4.44/5 (formación de los patrones de producto) y 4.31/5 (formación de los patrones de producto y marco metodológico).

- Los ingenieros software valoraron positivamente la afirmación de que el marco metodológico les proporciona los pasos que tienen que realizar a lo largo del desarrollo del proyecto, permitiéndoles en todo momento en qué punto está el proyecto y qué tienen que hacer.

- La valoración de los patrones de producto y el marco metodológico respecto a si ayudan a saber con qué otros ingenieros software tienen que colaborar y cuándo, fue positivo (3.8/5 patrones de producto; 4,1/5 marco metodológico).

En resumen, se puede concluir que las organizaciones implicadas en la validación de esta tesis doctoral han valorado la retroalimentación aportada respecto a los *patrones de producto* y a la estrategia corporativa descrita. Los miembros del equipo de desarrollo de los proyectos software llevados a cabo en la validación, han evaluado los campos del *patrón de producto*, siendo los campos más valorados, la “solución” que permite tener los pasos a seguir

para obtener el producto software, la “plantilla” que permite realizar el proceso software de una forma más usable, los “ejemplos” que proporcionan casos similares del desarrollo de la misma actividad con el *patrón de producto* y las “lecciones aprendidas” que proporciona conocimiento muy específico de la aplicación del *patrón de producto* en proyectos previos.

Los jefes de proyecto valoraron el grado de dificultad para seleccionar el *patrón de producto* apropiado para utilizarlo en una tarea como medio, reduciéndose dicha dificultad con el uso reiterado de los *patrones de producto*. Es por ello, que también los ingenieros software, por lo general, en las primeras actividades que utilizaron los *patrones de producto* tardaron en desarrollar el producto más, hasta que pasaba un pequeño periodo de aprendizaje.

Otros beneficios que las organizaciones valoraron positivamente fue el grado de satisfacción del equipo de trabajo utilizando los *patrones de producto* y el marco metodológico así como la mejora en la comunicación entre los miembros del equipo de trabajo permitiendo la transferencia del conocimiento en las organizaciones. Además, el entendimiento y el aprendizaje del proceso se mejora gracias a que se representan gráficamente los pasos que se tienen que realizar para desarrollar un producto software, de forma que en cada momento se sabe qué, quién y cómo se tiene que realizar las actividades del proyecto software facilitando la trazabilidad del proceso.

5. CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

5	CONCLUSIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN.....	217
5.1	Conclusiones	221
5.2	Futuras líneas de investigación	222

5.1 CONCLUSIONES

Esta tesis doctoral se ha centrado en el desarrollo de una solución práctica que permite la difusión, uso y reutilización de activos de procesos y proyectos, apoyándose en las áreas de Gestión de conocimiento, Reutilización y Entornos colaborativos.

Las aportaciones de este trabajo de investigación han sido múltiples. A continuación se describe cada una de ellas brevemente:

1. Se ha proporcionado una nueva vía para la definición de procesos en base a los productos que se quieran obtener, lo que permite el desarrollo dinámico de procesos y la visión multimodelo demandada por las organizaciones actualmente.
2. Se ha definido el concepto de *Patrón de Producto*, así como el proceso de retroalimentación y creación de nuevos *patrones de producto*. Este artefacto permite definir y almacenar el conocimiento de las mejores prácticas de la Ingeniería del Software y la experiencia de los ingenieros expertos, permitiendo su uso y reutilización en el desarrollo de procesos software.
3. Se ha definido una taxonomía de *Patrones de Producto* así como las reglas que gobiernan la recuperación de los mismos para su uso en proyectos software, la lógica de la ejecución de un proyecto software, así como sus mecanismos de colaboración. Todo lo cual, sirve como soporte al uso del concepto de *Patrón de producto*.
4. Se ha construido un catálogo de *Patrones de Producto* que sirve como repositorio de conocimiento de los activos de procesos y proyectos, que sirve de base de conocimiento para casos en los que las organizaciones no dispongan de conocimiento acerca de su propio proceso de desarrollo.
5. Se ha definido la estrategia corporativa de despliegue de la solución que describe los procesos para realizar la difusión, uso y reutilización de los activos de proceso en una organización.
6. Se ha construido la solución tecnológica basada en una plataforma colaborativa que da soporte a la solución planteada en esta tesis doctoral. Así mismo, se ha descrito cómo adaptar la estrategia corporativa a la solución tecnológica planteada.

Los grandes beneficios aportados en esta tesis doctoral son:

- De la implementación de la solución propuesta en proyectos reales con los *patrones de producto*, se puede concluir, que el uso de los *patrones de producto* mejora la eficiencia de uso de los procesos software, la calidad de los productos software y reducen el tiempo de rehacer dichos productos software.

- De la utilización del marco metodológico, que incluye a los patrones de producto, la estrategia corporativa y la plataforma colaborativa, los resultados obtenidos en el desarrollo de proyectos software, se puede concluir que la productividad de las organizaciones se incrementa debido al aumento de la eficiencia de uso de los procesos software. Esta aportación es debida, además de la utilización de los *patrones de producto*, al uso de la estrategia corporativa que proporciona los pasos que tiene que realizar el equipo de desarrollo para llevar a cabo el desarrollo de sus procesos software, lo que permite una mayor organización en el trabajo diario del equipo; y a las funcionalidades colaborativas que permiten una mejora en la comunicación entre los miembros del equipo de desarrollo.

5.2 FUTURAS LÍNEAS DE INVESTIGACIÓN

Como futuros trabajos que han quedado sin abordar o que han surgido a medida que se desarrollaba la tesis se propone:

- Adaptar el *patrón de producto* a formato multimedia para mejorar la usabilidad del *patrón de producto*.
- Incorporar mecanismos de medición a los *patrones de producto* para ayudar a alcanzar niveles de madurez más rápidamente.
- Construir *patrones de producto* para otros procesos implicados en las organizaciones e implantar el marco metodológico a todos los niveles organizativos y medir su impacto.
- Conectar los patrones de producto con controladores de la calidad del producto obtenido en el patrón, que serán alineados con objetivos estratégicos del negocio.
- Hacer minería de datos para extraer el conocimiento de qué modelos de procesos se adaptan más al tipo de organizaciones y proyectos, así como, extraer de los *patrones de producto* a qué modelos de procesos se han aplicado y poder educir las relaciones

potenciales entre modelos de proceso que permitiría un avance en el estudio de la ingeniería del software basada en multimodelos.

- Gracias a que los *patrones de producto* proporcionan una forma de definir y abordar los procesos, se pueden identificar nuevas estructuras de procesos y definiciones de procesos que surgen de forma natural del uso de los mismos.
- Incorporar mecanismos de trazabilidad de los activos de procesos y proyectos permitiendo medir y cuantificar el retorno de la inversión de la transferencia del conocimiento, convirtiéndolo en un activo tangible.
- Implementar mecanismos que permitan la comunicación interpersonal en la organización, rompiendo barreras humanas e institucionales.
- Tratar de convertir a las organizaciones en un ente inteligente tratando de que sepa lo que saben sus empleados y alineando los objetivos estratégicos con los objetivos de calidad de la organización.
- Aplicar los *patrones de producto* a otros ámbitos, como el académico, debido a que también proporcionan beneficios, como puede verse en IEEE Transaction on Education (Medina-Dominguez et al, 2009), permitiendo a los alumnos adquirir el conocimiento y las mejores prácticas para realizar sus actividades.

6. BIBLIOGRAFÍA

(Alexander et al, 1977) Alexander, C., Ishikawa, S., Silverstein, M., (1977). A Pattern Language. Oxford University Press.

(Alexander, 1979) Alexander, C., (1979). The Timeless Way of Building. Oxford: Oxford University Press.

(Allison et al, 2007). Allison, I., Merali, Y., (2007). Software process improvement as emergent change: A structural analysis. *Information and Software Technology*, 49, 6, pp. 668–681.

(Alur et al, 2003) Alur, D., Crupi, J., Malks, D.,(2003). Core J2EE patterns, Best Practices and Design Strategies. Prentice Hall, ISBN 0-13-142246-4.

(Ambler, 1998) Ambler, S.W., (1998). Process Patterns: Building large-scale systems using object technology. Cambridge University Press.

(Ambler, 1999) Ambler S.W., (1999). More Process Patterns: Delivering Large-Scale Systems Using Object Technology. Cambridge University Press.

(Antunes et al, 2007) Antunes, B., Seco, N., Gomes, P., (2007). Knowledge Management using Semantic Web Technologies: An Application in Software Development. K-CAP '07: Proceedings of the 4th international conference on Knowledge capture. New York, NY, USA: ACM, pp. 187–188.

(Appleton, 1997) Appleton, B., (1997). Patterns for conducting process improvement. In Proceedings of PLoP'97.

(Arent et al, 2000) Arent J., Norbjerg J. (2000). Software Process Improvement as Organizational Knowledge Creation: A Multiple Case Analysis. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. IEEE Computer Society Press.

(Assimakopoulos et al, 2006). Assimakopoulos, D., Yan, J., (2006) Sources of Knowledge Acquisition for Chinese Software Engineers, *R&D Management*, 36, 1, pp. 97-106.

(Basili et al, 1994) Basili, V., Caldiera, G., Rombach, D.H., (1994). The Experience Factory. *Encyclopedia of Software Engineering*, pp. 470-476, John Wiley Sons.

(Basili et al, 2007) Basili, V., Bomarius, F., Feldmann, R.L., (2007). Get Your Experience Factory Ready for the Next Decade--Ten Years after "How to Build and Run One". ICSE 2007, 29th International Conference on Software Engineering. 20, 26, pp.167–168.

(Baskerville et al, 1998) Baskerville R., Heje J.P, (1998). Managing knowledge Capability and Maturity. Presented at IFIP 8.2-8.6, Helsinki, Finland.

(Berczuk et al, 2003) Berczuk, S.P., Appleton, B., Brown, K. (2003). Software configuration management. Effective team. Practical integration. Addison-Wesley.

(Bjørnson et al, 2005) Bjørnson F.O., Stålhane,T., (2005). Harvesting Knowledge Through a Method Framework in an Electronic Process Guide. *Lecture Notes in Computer Science*, pp. 86-90.

(Bots et al, 2002) Bots, P.W.G., Bruin, H., (2002). Effective knowledge management in professional organizations: Going by the rules. Proceedings of the 35th Hawaii International Conference on System Sciences.

(Brede et al, 2006). Brede, N., Dyba, T., (2006). The use of an Electronic Process Guide in a Medium-sized Software Development Company. Software Process Improvement and Practice. 11, pp.21-34.

(Brunner et al, 2007) Brunner, J., Ma, L., Wang, C., Zhang, L., (2007). Explorations in the Use of Semantic Web Technologies for Product Information Management. Proceedings of the 16th international conference on World Wide Web, pp. 747 – 756.

(Burke et al, 2005) Burke,D., Howard,W., (2005). Knowledge Management and Process Improvement: A union of two disciplines. The journal of defense software engineering, Jun 2005. Disponible en: <http://www.stsc.hill.af.mil/crosstalk/2005/06/0506Burke.html>. Fecha último acceso: Diciembre, 2009.

(Buschmann et al, 1996). Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M., (1996). *Pattern Oriented Software Architecture. A system of Patterns*. New York: John Wiley & Sons.

(Butler, 1995). Butler, K. (1995). The economics benefits of software process improvement. CrossTalk, July, pp. 14-17.

(Campbell, 2004) Campbell, J.D., (2004). Interaction in collaborative computer supported diagram development. Computers in human behaviour. 20, pp. 289-310.

(Capell, 2004) Capell, P., (2004). Benefits of Improvement Efforts. Special Report CMU/SEI-2004-SR-010.

(Carstensen et al, 2002) Carstensen, P. H., Schmidt, K., (2002). Computer supported cooperative work: New challenges to systems design. Handbook of Human Factors. K. Itoh. Tokyo.

(Chau, T. et al, 2005) Chau, T., Maurer, F., (2005). A case Study of Wiki-based Experience Repository at a Medium-sized Software Company. Proceedings of the 3rd international conference on Knowledge capture, Canada, pp. 185-186.

(Christensen, 2003) Christensen, P.H., (2003). Knowledge management: perspectives and pitfalls. Copenhagen Business School Press.

Clark D. (2004) Understanding and Performance. Disponible en: <http://www.nwlink.com/~donclark/performance/understanding.html>. Fecha último acceso: Diciembre, 2009.

(Coleman, 1997) Coleman, D., (1997). Groupware: Collaborative Strategies for Corporate LANs and Intranets. Prentice Hall, Upper Saddle River, NJ.

(CMU/SEI, 2006) CMU/SEI-2006-TR-008 (2006). CMMI for Development, Version 1.2.

(Cooper, 1998) Cooper, J.W., (1998). The Design Patterns: Java Companion, Addison-Wesley.

(Coplien, 2004) Coplien, J., (2004). Patterns of engineering. Potentials, IEEE. 23, 2, pp.4-8.

(Crowe, 1994) Crowe M.K., (1994). Cooperative work with multimedia. Springer-Verlag.

(Cunningham et al, 2001) Cunningham, W., Leuf, B., (2001). The Wiki Way Quick Collaboration on the Web. Addison Wesley.

(Dave et al, 2000) Dave B., Danahy J., (2000). Virtual study abroad and exchange studio. International Journal of Automation in Construction. 9,1, pp 57-71.

(Deelstra et al, 2004) Deelstra, S., Sinnema, M., Bosch, J. (2004). Product derivation in software product families: a case study. The journal of System and Software. 74, pp. 173-194.

(Dewiyanti et al, 2004) Dewiyanti, S, Brand-Gruwel, S., Jochems, W., Broers, N.J., (2004). Students' experiences with collaborative learning in asynchronous Computer-Supported Collaborative Learning environments. International Journal of Computers in Human Behavior. 23, 1, 496-514.

(Dingsoyr, 2000) Dingsoyr, T., (2000). An Evaluation of Research on Experience Factory. Proceedings of the 2nd International Workshop on Learning Software Organizations (PROFES 2000), Oulu, Finland, pp. 55-66.

(Dingsoyr et al, 2009) Dingsøyr, T, Bjørnson, F. O., Shull, F. (2009). What Do We Know about Knowledge Management? Practical Implications for Software Engineering. IEEE Software. 26, 3, pp. 100-103.

(D'Souza et al, 2003) D'Souza, M., Greenstein, J.S., (2003). Listening to users in a manufacturing organization: A context-based approach to the development of a computer-supported collaborative work system. International Journal of Industrial Ergonomics. 32, pp 251-264.

(Falbo et al, 2004) Falbo, R.A., Borges, L.S.M., Valente, F.F.R., (2004). Using knowledge management to improve software process performance in a CMM level 3 organization. Proceedings of the 4th International Conference on Quality Software, pp. 162-169.

(Feher et al, 2006). Feher, P., Gabor, A., (2006). The Role of knowledge Management supporters in software development companies. Software Process improvement and practice. 11, pp. 251-260.

(Feldmann et al, 2003) Feldmann, R.L., Carbon, R., (2003). Experience Base Schema Building Blocks of the PLEASERS library. Journal of Universal Computer Science, 9, 7, pp. 659-669.

(Fowler, 1997) Fowler, M., (1997). Analysis Patterns – Reusable Object Models. Addison Wesley.

(Fowler, 2003) Fowler, M., (2003). Patterns. IEEE Software. Published by the IEEE Computer Society, 0740-7459.

(Gamma et al, 1994). Gamma, E., Helm, R., Johnson, R., Vlissides, J., (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley.

(Garner et al, 2003) Garner, S., Mann, P., (2003). Interdisciplinarity: perceptions of the value of computer-supported collaborative work in design for the built environment. *International Journal of Automation in Construction*. 12, pp. 495-499.

(Gartner 2005) Gartner group. (2005) Disponible en: <http://www4.gartner.com>. Fecha último acceso: Diciembre, 2009.

(Garson, 2006) Garson, E., (2006). *A Whirlwind Introduction to Process Patterns*. Dunstan Thomas Consulting.

(Gibson et al, 2006) Gibson, D., Goldenson, D., Kost, K., (2006). Performance Results of CMMI[®]-Based Process Improvement. CMU/SEI-2006-TR-004 ESC-TR-2006-004.

(Gnatz et al, 2003) Gnatz, M., Marschall, F., Popp, G., Rausch, A., Schwerin, W., (2003). The Living Software Development Process. *Journal Software Quality Professional*, 5, 3.

(Gonzalez-Reinhart, 2005) Gonzalez-Reinhart, J., (2005). Wiki and the Wiki Way: Beyond a Knowledge Management Solution. Information Systems Research Center, Bauer College of Business, University of Houston. Disponible en: http://www.uhsrc.com/FTB/Wiki/wiki_way_brief%5B1%5D-Jennifer%2005.pdf. Fecha último acceso: Diciembre, 2009.

(Greif, 1988) Greif, I., (1988). *Computer-supported cooperative work: A book of readings*. San Mateo, CA.: Morgan Kaufman.

(Gupta et al, 2008) Gupta, J.N.D., Sharma, S.K., Hsu, J., (2008). An overview of Knowledge Management. *Knowledge management: Concepts, Methodologies, Tools, and Applications*. Murray E Jennex, 1.

(Hagen et al, 2004) Hagen, M., Gruhn, V., (2004) Towards Flexible Software Processes by using Process Patterns. 8th International Conference on Software Engineering and Applications. Cambridge, USA, p. 436-441.

(He, 2007) He, J., Yan, H., Liu, C., Maozhong, J., (2007). A Framework of Ontology-Supported Knowledge Representation in Software Process.

(Henninger, 2003) Henninger, S., (2003). Tool Support for Experience-Based Software Development Methodologies. *Advances in Computers*, 59, pp. 29-82.

(Hey, 2004) Hey, J. 2004. *The Data, Information, Knowledge, Wisdom Chain: The Metaphorical link*.

(Holsapple et al, 2004) Holsapple, C.W., Joshi, K., (2004). A formal knowledge management ontology: Conduct, activities, resources, and influences. *Journal of the American Society for Information Science and Technology*, 55, 7, 593-612.

(Holz et al, 2006) Holz, H., Rostanin, O., Dengel, A., Suzuki, T., Maeda, T., Kanasaki, K. (2006). Task-based process know-how reuse and proactive information delivery in TaskNavigator. pp. 522-531.

- (Hornbæk, 2006) Hornbæk, K., (2006). Current practice in measuring usability: Challenges to usability studies and research. *International Journal Human-Computer Studies*, 64, pp.79–102.
- (Iida, 1999) Iida, H., (1999). *Pattern-Oriented Approach to Software Process Evolution*.
- (Intland Software, 2007) Intland Software. 2007. codeBeamer. Disponible en: <http://www.intland.com/products/codebeamer.html>. Fecha último acceso: Diciembre, 2009.
- (Isla-Montes et al, 2004) Isla-Montes, J.J., Gutiérrez-Vela, F.L., (2004). Diseño en base a patrones. Aplicación a sistemas hipermedia Colaborativos.
- (IPRC, 2006) A Process Research Framework (2006). The International Process Research Consortium. Eileen Forrester, editor. Software Engineering Institute. Carnegie Mellon.
- (Jennex et al, 2004) Jennex, M.E., Olfman, L., (2004). Modeling knowledge management success. *Proceedings of the Conference on Information Science and Technology Management, CISTM*.
- (Kellner et al, 1998) Kellner, M.I., Becker-Kornstaedt, U., Riddle, W.E., Tomal, J., Verlage, M., (1998). Process Guides: Effective Guidance for Process Participants. Published in *Proc. 5th Int'l. Conf. on the Software Process: Computer Supported Organizational Work*, pp. 11-25.
- (Kodaganallur et al, 2006) Kodaganallur, V., Shim, S., (2006). Analysis patterns: A taxonomy and its implications. *Information System Managemeng*; 23, 3, pp. 52-61.
- (Komi-Sirviö et al, 2002) Komi-Sirviö, S., Mäntyniemi, A., Seppänen, V., (2002). Toward a Practical Solution of Capturing knowledge for software Projects. *IEEE Software*, 19, 3, 60-62.
- (Lau et al, 2003) Lau L.M.S., Adams C.A., Dew P.M., Leigh C.M. Use of Scenario Evaluation in Preparation for Deployment of a Collaborative System for Knowledge Transfer – the Case of KiMERA. *IEEE Computer Society*.
- (Lindsey, 2002) Lindsey, K., (2002). Measuring knowledge management effectiveness: A task-contingent organizational capabilities perspective. *Proceedings of the 8th Americas Conference on Information Systems*, pp. 2085-2090.
- (Lukosch et al, 2004) Lukosch, S. Schümmer, T., (2004). Communicating Design Knowledge with Groupware Technology Patterns. The Case of Shared Object Management. *Lecture Notes in Computer Science*. Springer (Berlin), 3198, pp. 223-237.
- (Lukosch et al, 2006) Lukosch, S., Schümmer, T., (2006). Groupware development support with technology patterns. *International Journal Human-Computer Studies*, 64, pp. 599-610.
- (Maier, 2002) Maier, R. (2002). *Knowledge management systems: Information and communication technologies for knowledge management*. Berlin, Germany: Springer-Verlag.
- (Marca, 1992) Marca, D., (1992). *Groupware: Software for Computer-Supported Cooperative Work*. USA. Editorial: IEEE Computer Society.

(Massey et al, 2002) Massey, A.P., Montoya-Weiss, M.M., O'Driscoll, T.M., (2002). Knowledge management in pursuit of performance: Insights from Nortel networks. *MIS Quarterly*, 26, 3, pp. 269-289.

(Medina-Dominguez et al, 2009) Medina-Domínguez, F., Sánchez-Segura, M., Mora-Soto, A., Amescua, A. (2009). Reverse engineering and software products reuse to teach collaborative web portals: a case study with final-year computer science students. *Journal of IEEE Transaction on Education (JCR, Índice de impacto: 1.4, Computer Science Software engineering 2008)* Aceptado Octubre 2009. IEEE-Inst Electrical Electronics Engineers Inc.

(Microsoft Corporation, 2007) Microsoft Corporation. 2007. Visual Studio Team System. Disponible en: <http://msdn2.microsoft.com/en-us/teamsystem/default.aspx>. Fecha último acceso: Diciembre, 2009.

(Mohagheghi et al, 2007) Mohagheghi, P., Conradi, R., (2007). R. Quality, productivity and economic benefits of software reuse: a review of industrial studies. *Empirical Software Engineering*, 12, 5, pp. 471 – 516.

(Moe et al, 2006) Moe, N.B., Dyba, T., (2006). The Use of an Electronic Process Guide in a Medium-sized Software Development Company. *Software Process Improvement and Practice*. 11, pp. 21-34.

(Nakayama et al, 2005) Nakayama, T., Hirai, N., Naito, M., (2005). Adoption of structured abstracts by general medical journals and format for a structured abstract, *Journal of the Medical Library Association*, 93, 2, pp. 237-242.

(Niazi et al, 2006). Niazi, M., Wilson, D., Zowghi, D., (2006). *Critical Success Factors for Software Process Improvement Implementation: An empirical study*. *Software Process Improvement and practice*. 11, pp. 193-211.

(Nonaka et al, 1998) Nonaka, I., Cono, N., (1998). The concept of “Ba”: building a foundation for knowledge creation. *California Management Review*, 40, 3, 40-54.

(Osellus, 2007) Osellus. 2007. IRIS Process Author. <http://www.osellus.com/IRIS-PA>.

(Petter et al, 2008) Petter, S. Vaishnavi, V., (2008). Facilitating experience reuse among software project manager. *Information Sciences: an International Journal*. 178 , 7, pp. 1783-1802.

(Phillips, 2008) Phillips, M., (2008). The CMMI - a De-facto Standard for Software Process Improvement. The status of its adoption. *Software Engineering Institute*. Carnegie Mellon University.

(Phongpaibul, 2007) Phongpaibul, M., Koolmanojwong, S., Lam, A., Boehm, B. (2007). Comparative Experiences with Electronic Process Guide Generator Tools. *International Conference on Software Process, ICSP 2007*, pp. 61-72.

(Pitterman, 2000). Pitterman, B. (2000). Telcordia technologies: The journey to high maturity. July/August. *IEEE Software*, pp. 89-96.

(Polanyi, 1966) Polanyi, M., (1966). The tacit Dimension, Doubleday & Co, Garden City.

(Pratt et al, 2004) Pratt, W., Reddy, M.C., McDonald, D.W., Tarczy-Hornoch, P., Gennari, J.H., (2004). Incorporating ideas from computer-supported cooperative work. *International Journal of Biomedical Informatics*, 37, 2, pp. 128-137.

(Purao et al, 2003) Purao, S., Storey V., Han, T., (2003). Improving Analysis Pattern Reuse in Conceptual Design: Augmenting Automated Processes with Supervised Learning. *Information Systems Research*, 14, 3, pp. 269-290.

(Ramachandran, 2006) Ramachandran, M. 2006. Automated Improvement for Component Reuse. *Software process improvement and practice*. 11, pp. 591-599.

(Ribo et al, 2002) Ribó J.M., Franch X., (2002). Supporting Process Reuse in PROMENADE. Research Report LSI-02-14-R, Dep. LSI, Politechnical University of Catalonia.

(Rus, 2002) Rus I., Lindvall M., (2002). Knowledge Management in Software Engineering. *IEEE Software*.

(Shaw et al, 1996) Shaw, M., Garlan, D., (1996). *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, Upper Saddle River.

(Select Business Solutions, 2006) Select Business Solutions, (2006). Select Solution Factory. from <http://www.selectbs.com/products/select-solution-factory.htm>

(SEI, 2006) Software Engineering Institute, CMMI Executive Overview, (2006). Retrieved December, 2009 from <http://www.sei.cmu.edu/cmml/adooption/pdf/cmml-exec-overview06.pdf>.

(Selby, 2007) Selby, R., (2007). *Software engineering: Barry W. Boehm's lifetime contributions to software development, management, and research*. Hoboken, N.J.: Wiley-Interscience : IEEE Computer Society, cop. 2007.

(Silveira et al, 2002) Silveira Borges, L., Almeida Falbo, R., (2002). *Managing Software Knowledge*

(Sommerville, 2004). Sommerville I., (2004). *Software Engineering*. Fifth Edition, Addison Wesley.

(Sourcingmag, 2007) Sourcingmag (2007). Disponible en: <http://www.sourcingmag.com/>. Fecha último acceso: Diciembre, 2009.

(Standish Group, 2004). Standish Group, (2004). Third Quarter Research Report, The Standish Group International, Inc., West Yarmouth, M A.

(Standish Group, 2009). Standish Group, (2009). *Chaos Manifesto. The laws of Chaos and the Chaos 100 Best Practices*. The Standish Group International, Inc., West Yarmouth, M A.

(Stelzer et al, 1998) Stelzer D., Mellis W. Herzwurm G, (1998). *Technology Diffusion in Software Development Processes: The Contribution of Organizational Learning to Software*

Process Improvement. In *Information Systems Innovation and Diffusion: Issues and Directions*, T. Larsen and E. McGuire. Eds Hershey, USA: Idea Group Publisher, 1998, pp 297-344.

(Succi et al, 2001) Succi, G. Benedicenti, L., Vernazza, T., (2001). Analysis of the effects of software reuse on customer satisfaction in an RPG environment. *IEEE Transactions on Software Engineering*. 27, 5, pp. 473 – 479.

(Tran et al, 2006) Tran H.N., Coulette B., Dong B.T., (2006). A UML-Based Process Meta-model Integrating a Rigorous Process Patterns Definition. 4034, pp. 429-434.

(Tseng et al, 2008) Tseng, H.P., Lin, Y., (2008). A knowledge management portal system for construction projects Using Knowledge map. *Knowledge Management: Concepts, methodologies, tools, and applications*. Murray E. Jennex Volume II. IGI Global.

(Vigotsky, 1987) Vigotsky, S., (1987). *Historia del desarrollo de las funciones psíquicas superiores*. La Habana: Ed. Científico técnica.

(Woo et al, 2004) Woo, J., Clayton, M.J., Johnson, R.E., Flores, B.E., Ellis, C. (2004). Dynamic Knowledge Map: reusing experts' tacit knowledge in the AEC industry. *Automation in Construction*, 13, pp. 203-207.

(Yamamura, 1999) Yamamura, G., (1999). Software process satisfied employees. *IEEE Software* September/October, pp. 83-85.

