



Universidad  
Carlos III de Madrid

## **TESIS DOCTORAL**

# **ITAKA: Gestión Interactiva del Conocimiento en Organizaciones de Desarrollo de Software**

Autor:

**Alberto Heredia García**

Directores:

**Dr. Antonio de Amescua Seco**

**Dr. Javier García Guzmán**

Escuela Politécnica Superior

Departamento de Informática

Leganés, Junio 2012



# TESIS DOCTORAL

ITAKA: Gestión Interactiva del Conocimiento  
en Organizaciones de Desarrollo de Software

Autor: **Alberto Heredia García**

Directores: **Dr. Antonio de Amescua Seco**  
**Dr. Javier García Guzmán**

Firma del tribunal calificador:

Nombre

Firma

Presidente:

Vocal:

Vocal:

Vocal:

Secretario:

Calificación:

Leganés, de de



*El conocimiento es poder.*

Sir Francis Bacon

*En los momentos de crisis,  
sólo la imaginación es más importante que el conocimiento.*

Albert Einstein



## Resumen

El conocimiento es uno de los principales activos en un entorno económico e industrial tan competitivo como el actual. La necesidad de diferenciación de los competidores lleva a las empresas a gestionar su conocimiento organizacional. Cuando es gestionado de manera adecuada, el conocimiento se convierte en un recurso clave para muchas organizaciones porque permite mejorar su competitividad, siendo el descubrimiento de conocimiento visto como una fuente de innovación.

Puesto que la calidad del software está directamente relacionada con la calidad del proceso a través del cual es desarrollado este, muchas organizaciones de ingeniería del software tratan de mejorar la calidad de sus productos mejorando sus procesos organizacionales. Para ello, las compañías deben esforzarse continuamente en maximizar el uso de su conocimiento organizacional y/o experiencia. La manera en que las organizaciones recogen, gestionan, comparten, y reutilizan su conocimiento es habitualmente conocida como Gestión del Conocimiento (*Knowledge Management – KM*).

Los Sistemas de Gestión del Conocimiento (*Knowledge Management Systems – KMS*) proporcionan soporte para manejar el conocimiento en las organizaciones y la implementación de un KMS es una manera positiva de ayudar al aprendizaje basado en el trabajo a través de la aplicación de las tecnologías de la información. Los repositorios de conocimiento son habitualmente una de las estructuras básicas en todo KMS. Estos repositorios almacenan artefactos de conocimiento de tal manera que pueden ser recuperados y reutilizados para producir software de mayor calidad y también para mejorar el rendimiento de la organización.

En las organizaciones de ingeniería de software, este tipo de repositorio de conocimiento es habitualmente conocido como Biblioteca de Activos de Proceso (*Process Asset Libraries – PAL*). Una PAL se puede definir como un repositorio de activos de proceso que está organizado, bien indexado y que permite búsquedas, siendo fácilmente accesible por quienquiera que necesite información para el guiado de procesos, ejemplos, datos, plantillas u otros materiales de ayuda a los procesos. Una PAL puede ser utilizada como el elemento clave de la infraestructura necesaria para diseminar y aprender prácticas efectivas entre los ingenieros de software de la organización, aumentando sus competencias y sus capacidades.

El conocimiento útil y relevante es sin embargo difícil de encontrar, y cuando es encontrado su replicación en la práctica. Gran parte del conocimiento útil para la ejecución de los procesos de software es personal, específico del contexto y difícil de formalizar y comunicar. Esta tesis doctoral se enfoca al establecimiento de mecanismos para enriquecer el conocimiento organizacional dentro del contexto del desarrollo de software añadiendo conocimiento tácito al ya existente de una manera asistida y semi-automática, haciendo además que su diseminación a lo largo de la organización sea sencilla.

ITAKA (*Interactive TAcit Knowledge Administration*) es la solución propuesta en esta tesis doctoral para tratar de resolver estos problemas. Basándose en los principios del KM, ITAKA proporciona un marco de trabajo compuesto por un modelo de conocimiento, los procesos para su gestión y una plataforma tecnológica para la administración interactiva y colaborativa

del conocimiento tácito necesario para facilitar el aprendizaje y la adopción de nuevos procesos de desarrollo de software.

ITAKA implementa mecanismos para (1) adquirir el conocimiento tácito que las personas obtienen mientras trabajan en distintos equipos para enriquecer colaborativamente el conocimiento organizacional, y (2) diseminar el conocimiento organizacional mejorado entre los diferentes equipos de proyecto. Esta propuesta está inicialmente orientada –pero no limitada– a entornos que requieran el aprendizaje y la adopción de nuevos procesos (p.ej. iniciativas de mejora continua de procesos y la integración de nuevas personas en la organización).

ITAKA ha sido validada en un experimento controlado realizado durante dos años en un curso de formación en el que participaron ingenieros junior. Los resultados de esta validación muestran que ITAKA es efectivo en la captura, formalización y distribución del conocimiento que surge de las diferentes interacciones entre los miembros de una organización de ingeniería de software. ITAKA también proporciona un entorno efectivo para el aprendizaje y la adopción de nuevos procesos y prácticas para el desarrollo de software de calidad. A pesar del poco esfuerzo requerido para utilizar las funcionalidades del sistema, se necesita de cierto tiempo para acostumbrarse a su uso y para lograr que los usuarios se involucren. Así, la calidad de los productos desarrollados mejora con el paso del tiempo.

Los resultados también sugieren que el éxito de ITAKA depende de factores tales como: a) la utilidad percibida de las herramientas, b) el número de activos de conocimiento publicados, c) la cantidad de información duplicada, d) la distribución del nuevo conocimiento aportado, e) la usabilidad de herramientas y procesos, f) el conocimiento adquirido por todos los usuarios, y g) el cambio cultural que supone. Estos factores resaltan áreas importantes para futuros investigadores y profesionales a la hora de desarrollar entornos colaborativos de aprendizaje en ingeniería del software.



## Abstract

Knowledge is one of the main assets in today's highly competitive environment. The need for differentiation from competitors led companies to manage their existing organizational knowledge. When managed properly, knowledge becomes a key resource for many organizations because it can improve their competitiveness, knowledge discovery being seen as a source of innovation.

Given that the quality of software is directly related to the quality of the process through which software is developed, many software engineering organizations try to enhance the quality of their products by improving their organizational processes. To do so, companies must continually strive to maximize the use of their organizational knowledge and/or experience. The way organizations gather, manage, share, and reuse their knowledge is often referred to as Knowledge Management (KM).

Knowledge Management Systems (KMS) support the handling of knowledge in organizations and the implementation of a KMS is a positive way of supporting work-based learning through the application of information technology. Knowledge repositories are usually one of the base structures in every KMS. A knowledge repository stores knowledge artefacts in such manner they can be retrieved and reused in order to produce software of higher quality and also to improve the organizational performance.

In software engineering organisations, this kind of knowledge repository is usually known as Process Asset Library (PAL). A PAL can be defined as an organised, well-indexed, searchable repository of process assets that is easily accessible by anyone who needs process guidance information, examples, data, templates or other process support materials. A PAL can be used as the key infrastructure element that is required to support the dissemination and learning of effective practices among an organisation's software engineers, increasing their competences and capabilities.

Useful and relevant knowledge, however, is difficult to find and when found, is difficult to replicate in practice. Most of the knowledge useful for the execution of software processes is personal, context-specific and hard to formalise and communicate. This doctoral thesis aimed to establish mechanisms to enrich the organisational knowledge within a software development context by adding tacit knowledge to the pre-existing one in an assisted and semi-automatic way, making it easier to disseminate throughout the organisation.

ITAKA (Interactive TAcit Knowledge Administration) is the solution proposed in this doctoral thesis to tackle these problems. Based on KM principles, ITAKA provides a framework composed of a knowledge model, the processes for its management and a technological platform to interactively and collaboratively administrate the tacit knowledge needed for helping in the learning and adoption of new software development processes.

ITAKA implements mechanisms to (1) acquire the tacit knowledge individuals obtain while working in different teams to collaboratively enrich organizational knowledge, and (2) disseminate the upgraded organizational knowledge among different project teams. This proposal is initially targeted at –but not restricted to– environments that require the learning

and adoption of new processes (e.g. continuous process improvement initiatives and the integration of new people into the organization).

ITAKA was validated in a controlled experiment involving junior engineers in a training course over two years. The results of this validation show that ITAKA is effective in capturing, formalising and distributing the knowledge that arises in different interactions among the members of a software engineering organization. ITAKA also provides an effective environment in order to learn and adopt new processes and practices to develop quality software. In spite of the little effort required for using the features of the system, it requires some time to get used to them and to get users involved. Thus, the quality of the products developed improves with the time.

The results also suggest that success of ITAKA depends on factors such as: a) perceived use of the tools, b) number of knowledge assets published, c) duplication of information, d) distribution of the contributed new knowledge, e) usability of tools and processes, f) knowledge acquired by all users, and g) cultural change. These factors highlight important areas for future researchers and practitioners when developing collaborative learning environments in software engineering.

## Agradecimientos

Quisiera dar las gracias en primer lugar a mis padres, Dori y Pedro, que siempre están ahí para darme su apoyo incondicional y para hacer que cada vez piense un poco más en positivo.

Gracias por supuesto a mis directores de tesis. Antonio, gracias por darme esta oportunidad de orientar mi vida profesional a la investigación, por tu apoyo durante estos años y, sobre todo, por pensar siempre primero en las personas. Javier, gracias por tu dedicación, por estar en el día a día y por dedicar parte de tu tiempo a mi formación profesional.

Muchas gracias también a Ricardo. Gracias por tu ayuda y tus siempre sabios consejos, pero especialmente gracias por ser un gran compañero.

También quiero dar las gracias al resto de compañeros investigadores de distintos departamentos y grupos de investigación con los que he compartido los buenos y los malos momentos todos estos años. Gracias a los que todavía estáis por aquí, gracias a los que habéis decidido buscar, por unos motivos o por otros, un futuro fuera de la Universidad, y gracias a los que habéis estado aquí pero habéis vuelto a vuestros países de origen. Gracias por vuestro cariño, vuestro apoyo y vuestras sonrisas.

Por último, que no por ello menos importantes, gracias a mi familia y amigos, con los que la distancia es solo una barrera física pero no emocional. Muchas gracias por estar siempre atentos de mis avances y por darme vuestros ánimos.

A todos, GRACIAS.



## Tabla de contenido

Resumen.....	I
Abstract .....	III
Agradecimientos .....	V
Tabla de contenido.....	VII
Índice de figuras .....	IX
Índice de tablas .....	XI
Capítulo 1: Introducción.....	1
1.1. Contexto .....	3
1.2. Descripción del problema y motivación .....	4
1.3. Hipótesis de trabajo.....	5
1.4. Objetivos de la tesis doctoral.....	5
1.5. Aproximación a la solución .....	6
1.6. Aportaciones a la investigación .....	7
1.7. Validez de la solución.....	9
1.8. Estructura de la tesis doctoral .....	13
Capítulo 2: Estado de la Cuestión .....	15
2.1. Gestión del Conocimiento .....	17
2.2. Gestión del Conocimiento en Ingeniería del Software .....	28
2.3. Soluciones existentes para la gestión del conocimiento acerca de procesos de software .....	45
2.4. Análisis de soluciones existentes y conclusiones .....	58
2.5. Conclusiones del estado de la cuestión .....	63
Capítulo 3: Solución Propuesta .....	65
3.1. Visión General.....	67
3.2. Estructura del Conocimiento .....	68
3.3. Modelo de Procesos .....	70

Capítulo 4: Implementación de la Solución Propuesta.....	113
4.1. Arquitectura del sistema.....	115
4.2. Componentes para la persistencia del conocimiento organizacional .....	116
4.3. Componentes para la formalización del conocimiento pre-existente .....	118
4.4. Componentes para la distribución del conocimiento organizacional .....	118
4.5. Componentes para la aplicación y preservación de conocimiento .....	123
4.6. Componentes para la gestión de cambios en el conocimiento organizacional .....	126
4.7. Componentes para la medición de conocimiento.....	129
 Capítulo 5: Validación de la Solución Propuesta.....	 131
5.1. Descripción del método.....	134
5.2. Contexto .....	135
5.3. Planificación.....	137
5.4. Recogida de datos y métodos de análisis.....	140
5.5. Resultados.....	143
5.6. Conclusiones de la validación .....	163
 Capítulo 6: Conclusiones y Futuras Líneas de Investigación .....	 167
6.1. Conclusiones .....	169
6.2. Líneas futuras.....	173
 Capítulo 7: Referencias .....	 175
 Anexos.....	 187
Anexo A: Criterios para la Evaluación de Productos .....	189
Anexo B: Plantillas para Editar el Conocimiento Pre-Existente.....	191

## Índice de figuras

Figura 1.1. Temas y nodos de investigación definidos por el IPRC. ....	8
Figura 2.1. Modelo SECI. ....	19
Figura 2.2. Ciclo de vida del conocimiento de Birkinshaw y Sheehan. ....	20
Figura 2.3. Modelo IDEAL. ....	33
Figura 2.4. Marco conceptual para procesos de software. ....	42
Figura 2.5. Esquema conceptual extendido. ....	43
Figura 2.6. Framework de WAGNER. ....	47
Figura 2.7: Framework de VSTS Process Guidance Generator. ....	49
Figura 3.1. Visión conceptual de ITAKA. ....	67
Figura 3.2. Modelo de la estructura del repositorio de conocimiento organizacional de ITAKA. ....	68
Figura 3.3. Clasificación de activos dentro del repositorio de conocimiento organizacional de ITAKA. ....	69
Figura 3.4. Procesos definidos dentro de ITAKA. ....	72
Figura 3.5. Proceso 1: Identificación del Conocimiento. ....	73
Figura 3.6. Actividades del Proceso 1: Identificación del Conocimiento. ....	74
Figura 3.7. Actividad 1.1: Recopilar procesos. ....	75
Figura 3.8. Actividad 1.2: Recopilar bibliografía. ....	76
Figura 3.9. Actividad 1.3: Recopilar experiencias. ....	77
Figura 3.10. Proceso 2: Formalización del Conocimiento. ....	78
Figura 3.11. Actividades del Proceso 2: Formalización del Conocimiento. ....	79
Figura 3.12. Actividad 2.1: Crear activos de conocimiento. ....	80
Figura 3.13. Tareas de la Actividad 2.1: Crear activos de conocimiento. ....	80
Figura 3.14. Actividad 2.2: Relacionar activos de conocimiento. ....	82
Figura 3.15. Actividad 2.3: Registrar activos de conocimiento. ....	83
Figura 3.16. Proceso 3: Distribución del Conocimiento. ....	84
Figura 3.17. Actividades del Proceso 3: Distribución del Conocimiento. ....	85
Figura 3.18. Actividad 3.1: Seleccionar activos. ....	86
Figura 3.19. Actividad 3.2: Integrar activos. ....	87
Figura 3.20. Actividad 3.3: Generar vistas. ....	88
Figura 3.21. Proceso 4: Aplicación del Conocimiento. ....	89
Figura 3.22. Actividades del Proceso 4: Aplicación del Conocimiento. ....	90
Figura 3.23. Actividad 4.1: Consultar activos. ....	91
Figura 3.24. Actividad 4.2: Instanciar activos. ....	92
Figura 3.25. Proceso 5: Preservación del Conocimiento. ....	93
Figura 3.26. Actividades del Proceso 5: Preservación del Conocimiento. ....	94
Figura 3.27. Actividad 5.1: Registrar ejemplos. ....	95
Figura 3.28. Tareas de la Actividad 5.1: Registrar ejemplos. ....	95
Figura 3.29. Actividad 5.2: Participar en discusiones. ....	97
Figura 3.30. Tareas de la Actividad 5.2: Participar en discusiones. ....	97

Figura 3.31. Actividad 5.3: Editar activos. ....	99
Figura 3.32. Tareas de la Actividad 5.3: Editar activos. ....	99
Figura 3.33. Proceso 6: Gestión de Cambios. ....	100
Figura 3.34. Actividades del Proceso 6: Gestión de Cambios. ....	101
Figura 3.35. Actividad 6.1: Identificar cambios. ....	102
Figura 3.36. Tareas de la Actividad 6.1: Identificar cambios. ....	102
Figura 3.37. Actividad 6.2: Incorporar cambios con revisión. ....	103
Figura 3.38. Tareas de la Actividad 6.2: Incorporar cambios con revisión. ....	103
Figura 3.39. Actividad 6.2: Incorporar cambios sin revisión. ....	105
Figura 3.40. Tareas de la Actividad 6.2: Incorporar cambios sin revisión. ....	105
Figura 3.41. Proceso 7: Medición del Conocimiento. ....	107
Figura 3.42. Actividades del Proceso 7: Medición del Conocimiento. ....	108
Figura 3.43. Actividad 7.1: Determinar efectividad del sistema. ....	109
Figura 3.44. Actividad 7.2: Determinar calidad del conocimiento. ....	110
Figura 3.45. Actividad 7.3: Determinar satisfacción de los usuarios. ....	111
Figura 4.1. Arquitectura de ITAKA. ....	115
Figura 4.2. Diagrama de componentes del sistema propuesto. ....	116
Figura 4.3. Diagrama de componentes para la distribución del conocimiento organizacional. ....	119
Figura 4.4. Diagrama de clases del Integrador. ....	120
Figura 4.5. Relaciones entre activos de conocimiento. ....	120
Figura 4.6. Diagrama de clases del Conversor. ....	122
Figura 4.7. Diagrama de componentes para la distribución del conocimiento organizacional. ....	126
Figura 4.8. Diagrama de clases del Editor de cambios. ....	127
Figura 4.9. Diagrama de clases del Editor de lecciones aprendidas. ....	128
Figura 4.10. Fusión de discusiones en forma de lecciones aprendidas utilizando el editor. ....	128
Figura 4.11. Diagrama de clases del Extractor de ejemplos. ....	129
Figura 5.1. Escala de tiempos del experimento. ....	136
Figura 5.2. Etapas de ejecución del experimento. ....	138
Figura 5.3. Aplicación software desarrollada durante el experimento. ....	139
Figura 5.4. Número de visitas por usuario vs. Duración de las visitas. ....	148
Figura 5.5. Número de visitas por usuario vs. Páginas vistas durante las visitas. ....	148
Figura 5.6. Nuevo conocimiento en la etapa de proyecto de la fase 2. ....	160



## Índice de tablas

Tabla 2.1. Comparativa de wikis. ....	26
Tabla 2.2. Soporte del KM en las fases del modelo IDEAL. ....	33
Tabla 2.3. Resumen del análisis de soluciones existentes. ....	59
Tabla 3.1. Descripción de elementos SPEM 2.0. ....	71
Tabla 3.2. Roles y responsabilidades dentro de ITAKA. ....	72
Tabla 5.1. Relación de hipótesis y preguntas de investigación. ....	134
Tabla 5.2. Número total de contribuciones. ....	144
Tabla 5.3. Estadísticas acerca de las nuevas contribuciones y del uso de las wikis en la fase 2. ....	146
Tabla 5.4. Percepción de la calidad del nuevo conocimiento según los instructores. ....	151
Tabla 5.5. Percepción de la calidad del nuevo conocimiento según los participantes. ....	151
Tabla 5.6. Factores relacionados con la calidad de los ejemplos en la etapa de proyecto en la fase 2. ....	154
Tabla 5.7. Factores relacionados con la calidad de las discusiones en la etapa de proyecto en la fase 2. ....	155
Tabla 5.8. Factores relacionados con la calidad de las lecciones aprendidas en la etapa de proyecto en la fase 2. ....	156
Tabla 5.9. Niveles de calidad percibidos utilizando distintos mecanismos de realimentación (supervisado y no-supervisado). ....	158
Tabla 5.10. Niveles de calidad de los productos en la etapa de proyecto de las fases 1 y 2. ....	159
Tabla 5.11. Utilidad del nuevo conocimiento percibida por los participantes. ....	161
Tabla 5.12. Niveles de utilidad percibidos utilizando distintos mecanismos de realimentación (supervisado y no-supervisado). ....	162



# Capítulo 1: Introducción

1.1. Contexto.....	3
1.2. Descripción del problema y motivación .....	4
1.3. Hipótesis de trabajo .....	5
1.4. Objetivos de la tesis doctoral.....	5
1.5. Aproximación a la solución .....	6
1.6. Aportaciones a la investigación .....	7
1.7. Validez de la solución.....	9
1.7.1. Aproximación a la validación de la solución .....	9
1.7.2. Limitaciones de la solución propuesta .....	10
1.8. Estructura de la tesis doctoral .....	13



Esta tesis doctoral propone una aplicación práctica de la Gestión del Conocimiento (*Knowledge Management* – KM) en la Ingeniería del Software para ayudar a gestionar y a mejorar los procesos de software.

En este primer capítulo se hará una introducción a la tesis doctoral, presentando el contexto dentro del cual ha sido realizada y la problemática actual encontrada dentro de dicho contexto, se plantearán las hipótesis de trabajo y los objetivos de investigación, se expondrá una breve aproximación a la solución, se detallarán las aportaciones de esta tesis doctoral a la investigación y, por último, se analizará la validez de la solución propuesta.

## 1.1. Contexto

El conocimiento es uno de los activos más importantes que poseen las organizaciones ya que, si es gestionado adecuadamente, puede ayudar a mejorar su competitividad [Choi, 2008]. Por ello, el KM juega un papel cada vez más relevante en las organizaciones actuales hasta el punto que muchos académicos y profesionales consideran este dominio como uno de los factores más importantes en la producción de software, que es el ámbito en el que se centra esta tesis doctoral. Las propias organizaciones reconocen que el conocimiento y su transferencia son la clave para su crecimiento, para la efectiva ejecución de su estrategia de negocio y para promover la innovación, para lo cual se hace vital potenciar una cultura organizacional que promueva una gestión efectiva del conocimiento [Ajmal, 2008].

Los estudios empíricos realizados han encontrado evidencias de la relación existente entre la inversión en Tecnologías de la Información y la Comunicación (TICs) y las mejoras en la productividad [Sircar, 2009]. Las TICs actuales permiten a los sistemas de gestión del conocimiento ser capaces de manejar dicho conocimiento de una manera efectiva y eficiente [Maier, 2007].

Los repositorios de conocimiento son habitualmente una de las estructuras básicas en todo sistema de gestión del conocimiento. Un repositorio de conocimiento almacena artefactos de conocimiento de tal manera que pueden ser fácilmente recuperados y reutilizados [Alavi, 2001] con el objetivo de generar productos de mayor calidad [Gupta, 2009] y de mejorar el rendimiento de la organización [Maier, 2007].

Centrando la atención en las organizaciones de ingeniería del software, este tipo de repositorio que puede gestionar el conocimiento organizacional es conocido como Biblioteca de Activos de Proceso (*Process Asset Library* – PAL) [SEI, 2006]. Una PAL es un repositorio de activos de proceso que está organizado, bien indexado y que permite búsquedas, siendo fácilmente accesible por quienquiera que necesite información para el guiado de procesos, ejemplos, datos, plantillas u otros materiales de ayuda a los procesos [García, 2004].

En las organizaciones de software, una PAL puede ser utilizada como elemento clave de la infraestructura que es necesaria para dar soporte a la diseminación y al aprendizaje de prácticas efectivas entre los ingenieros de software de la organización, aumentando sus competencias y habilidades [Calvo-Manzano, 2008]. Una PAL no solo contiene conocimiento acerca de cómo realizar los procesos, sino que también debería contener las lecciones aprendidas en proyectos anteriores con el objetivo de mejorar los procesos. Los proyectos de

desarrollo de software pueden mejorar su rendimiento (en términos de coste, calidad y planificación) reutilizando las lecciones aprendidas en proyectos anteriores [Basili, 2002] almacenadas en una PAL.

## 1.2. Descripción del problema y motivación

Aunque el conocimiento existente dentro del entorno de una organización puede ser beneficioso para esta, habitualmente resulta difícil encontrar conocimiento que sea útil y relevante, y cuando es encontrado, su reutilización es difícil de conseguir en la práctica [Komi-Sirviö, 2002] [Rus, 2002]. La razón principal es que gran parte del conocimiento relacionado con los procesos de software es personal, específico de un contexto, y difícil de formalizar y de comunicar entre las personas [Assimakopoulos, 2006]. Además, este tipo de conocimiento solamente se hace presente por medio de la práctica y, para su captura, suele ser necesario que las personas formen parte de un equipo de trabajo [Goffin, 2011].

A pesar de las malas condiciones económicas de los últimos años, las organizaciones continúan invirtiendo cantidades sustanciales de sus recursos para la gestión de su propio conocimiento y para ofrecer oportunidades a sus empleados para el aprendizaje de los procesos de la organización, según datos de la American Association for Training & Development (ASTD). Sin embargo, son pocas las organizaciones que alcanzan los beneficios esperados en relación con la inversión realizada en gestión y transferencia de conocimiento organizativo [Strong, 2008].

Por lo tanto, sería interesante poder encontrar mecanismos que permitieran enriquecer el conocimiento organizacional acerca de los procesos de software añadiendo conocimiento personal (tácito) al conocimiento pre-existente y facilitando además su reutilización para el aprendizaje de los procesos de la organización.

Para que el conocimiento organizacional evolucione y se enriquezca es necesario que se produzca una transformación de conocimiento tácito en explícito, y viceversa, puesto que es dicha alternancia la que puede dar lugar a la creación de nuevo conocimiento [Nonaka, 2009]. Para fomentar esta alternancia de conocimiento es necesario que todos los miembros de la organización puedan crear, modificar y/o reutilizar el conocimiento de una manera colaborativa.

Las wikis son una de las tecnologías que permitirían la generación colaborativa de este conocimiento, apoyando así la gestión del conocimiento a acerca de los procesos de una organización [Chau, 2005]. De acuerdo con [Majchrzak, 2006], la participación en wikis organizacionales y su utilización para el aprendizaje de procesos permite obtener tres tipos principales de beneficios: mejorar la reputación, un aprendizaje más sencillo, y ayudar a la organización a mejorar sus procesos por medio del uso efectivo del conocimiento recogido durante los procesos de aprendizaje colaborativo.

Los beneficios obtenidos gracias al uso sistemático de wikis para el aprendizaje de procesos son más probables de obtener cuando las tareas a realizar requieren obtener soluciones novedosas (más que tareas rutinarias) y cuando se confía en que el resto de participantes en la wiki aportan información fehaciente. Así, para poder asegurar la credibilidad, y por tanto para asegurar el enriquecimiento del conocimiento organizacional, los mecanismos para enriquecer

el conocimiento deberían garantizar que el nuevo conocimiento capturado tenga una alta calidad según los criterios de la organización.

### 1.3. Hipótesis de trabajo

La hipótesis formulada para este trabajo de investigación que se tratará de validar en la presente tesis doctoral es la siguiente:

*Si se dispone de un marco de trabajo para la gestión del conocimiento destinado al aprendizaje y adopción de procesos de software, basado en una PAL y que incorpora mecanismos de gestión de la configuración y de gestión del cambio, entonces:*

- **H1.1:** *Se incrementa la efectividad en la captura del conocimiento tácito generado por los ingenieros de software durante la ejecución de los proyectos de la organización, convirtiendo la PAL en una fuente relevante de conocimiento.*
- **H1.2:** *Se facilita el enriquecimiento y la diseminación de activos de conocimiento de alta calidad elicitados durante la ejecución de la iniciativa de aprendizaje.*
- **H1.3:** *Se ayuda a mejorar la capacidad para el aprendizaje y la adopción de nuevas prácticas para el desarrollo de software dentro de la organización.*

En el Capítulo 5: Validación de la Solución Propuesta de esta tesis doctoral se detalla el trabajo de validación que se ha llevado a cabo para comprobar que esta hipótesis es cierta, estableciendo la forma en que el marco de trabajo para la gestión del conocimiento propuesto en esta tesis ayuda a afrontar los problemas mencionados en la sección anterior.

### 1.4. Objetivos de la tesis doctoral

Para poder validar las hipótesis enumeradas anteriormente, el **objetivo general** de esta tesis doctoral es definir e implementar un marco de trabajo para el enriquecimiento del conocimiento que ayude a ingenieros de software que trabajen en distintos equipos al aprendizaje colaborativo y a la adopción de procesos de software gracias al enriquecimiento del conocimiento y a la aplicación de los principios de gestión de la configuración y de gestión del cambio.

La tesis doctoral se orienta por tanto hacia el establecimiento de mecanismos para enriquecer el conocimiento organizacional en el contexto del desarrollo de software, añadiendo conocimiento tácito al conocimiento pre-existente de una manera asistida y semi-automática, enriqueciendo así el conocimiento organizacional y haciendo más sencilla la difusión de este conocimiento enriquecido a lo largo de la organización. Basándose en los principios de la gestión del conocimiento y de la gestión de la configuración a través de la utilización de wikis, esta tesis doctoral propone un marco de trabajo para mejorar el aprendizaje colaborativo acerca de los procesos de desarrollo de software y su adopción. Aunque inicialmente está orientado a entornos donde la adopción y el aprendizaje de procesos de software son importantes (por ejemplo, iniciativas de mejora de procesos de software o la incorporación de nuevos ingenieros a una organización de software) y a entornos donde el aprendizaje

colaborativo es esencial (por ejemplo, para ingenieros de software junior), su aplicación no está necesariamente restringida a dichos entornos.

Para alcanzar el objetivo general mencionado antes se han considerado varios **objetivos específicos**, que son:

- **O1:** Definir un proceso repetible y sistematizable para la gestión del conocimiento destinado al aprendizaje y adopción de procesos de software, basado en una PAL y que incorpore mecanismos de gestión de la configuración y de gestión del cambio.
- **O2:** Diseñar y desarrollar una plataforma para la gestión del conocimiento destinado al aprendizaje y adopción de procesos de software, basado en una PAL y que incorpore mecanismos de gestión de la configuración y de gestión del cambio.
- **O3:** Determinar cuáles son los factores que promueven la utilización de la solución propuesta en esta tesis doctoral como fuente de conocimiento relevante para los ingenieros que buscan aprender y adoptar un nuevo proceso de software.
- **O4:** Validar experimentalmente que la plataforma desarrollada permite:
  - Incrementar la efectividad en la captura del conocimiento tácito generado por los ingenieros de software durante la ejecución de los proyectos de la organización, convirtiendo la PAL en una fuente relevante de conocimiento.
  - Facilitar el enriquecimiento y la diseminación de activos de conocimiento de alta calidad elicitados durante la ejecución de la iniciativa de aprendizaje.
  - Ayudar a mejorar la capacidad para el aprendizaje y la adopción de nuevas prácticas para el desarrollo de software dentro de la organización.

## 1.5. Aproximación a la solución

En este apartado se realiza una descripción conceptual de ITAKA (Interactive TAcit Knowledge Administration), la solución que esta tesis doctoral propone a los problemas mencionados en el apartado 1.2. Esta es una solución teórico-práctica para el enriquecimiento del conocimiento organizacional, la cual está orientada a facilitar el aprendizaje colaborativo y la adopción de nuevos procesos por parte de ingenieros de software que trabajen en distintos equipos de desarrollo.

Para ello, ITAKA aplica los principios de KM en el ámbito de la Ingeniería del Software, incorporando además mecanismos de gestión de la configuración y de gestión del cambio, con el objetivo de que el conocimiento tácito que se genera a partir de las interacciones entre los miembros de los equipos de trabajo durante la realización de los proyectos se fusione con el resto del conocimiento ya existente en la organización. Por otro lado, la solución también utiliza las tecnologías de la información para dar soporte al despliegue práctico del marco de trabajo que se propone en esta tesis doctoral.

De esta manera, ITAKA se compone de dos elementos principales: un proceso para ayudar a las organizaciones a llevar a cabo una buena gestión de su conocimiento organizativo, y una plataforma para dar soporte a dicho proceso. A continuación se describen más en detalle los conceptos que abarcan estos componentes.



En relación con el **proceso**, este describe de manera formal el método general de trabajo que debe implementarse dentro de una organización para utilizar ITAKA y poder así llevar a cabo la gestión y enriquecimiento de su conocimiento. Este proceso describe actividades que permiten:

- La adquisición del conocimiento pre-existente necesario para llevar a cabo los procesos de la organización, quedando este almacenado en un repositorio.
- La distribución del conocimiento pre-existente en el repositorio, difundiéndolo a lo largo de la organización a través de distintas plataformas.
- La reutilización del conocimiento organizacional por parte de las personas que participen en los procesos de los distintos proyectos de la organización.
- La preservación de aquel conocimiento tácito generado durante la realización de los proyectos que sea útil y relevante para la organización, añadiéndolo al conocimiento pre-existente para enriquecer así el conocimiento organizacional.
- La distribución del conocimiento enriquecido, de manera que el conocimiento tácito generado por un equipo de trabajo sea difundido al resto de equipos de desarrollo de la organización.

En cuanto a la **plataforma**, esta describe un conjunto de herramientas tecnológicas que pueden ser utilizadas para implementar el proceso en la práctica. Estas herramientas de software permitirán automatizar gran parte de las actividades que componen el proceso. Los componentes principales de esta plataforma serán:

- Codificador de contenidos: permitirá un mejor manejo del conocimiento almacenado.
- Integrador de contenidos: permitirá reunir todo el conocimiento almacenado bajo un único soporte.
- Generador de contenidos con salida multi-formato: permitirá crear distintas formas de acceso a los activos de conocimiento almacenados.
- Gestor de cambios de contenidos: permitirá que el conocimiento tácito que se haya generado durante la realización de los proyectos de la organización se fusione con el resto del conocimiento almacenado.

## 1.6. Aportaciones a la investigación

En Agosto del año 2004, el Software Engineering Institute (SEI) fundó el International Process Research Consortium (IPRC), un foro compuesto por investigadores y profesionales líderes en el ámbito de procesos cuyo propósito es explorar las necesidades actuales y futuras relacionadas con la gestión de procesos de software. Como consecuencia, dos años después se publicó el Marco de Investigación de Procesos [IPRC, 2006], el cual recoge las bases de las principales líneas de investigación en procesos hasta el año 2014.

Debido al reconocimiento internacional tanto del SEI como de los miembros del IPRC, el trabajo realizado en esta tesis doctoral se alinea dentro de este Marco de Investigación de Procesos, asegurando así que ITAKA responde a la problemática actual y futura de la investigación en relación con los procesos de software.

Más concretamente, la solución propuesta en esta tesis doctoral se enmarca dentro de los siguientes temas y nodos de investigación descritos en [IPRC, 2006] (Figura 1.1):

- **Tema Q: Relación entre las Calidades de los Procesos y del Producto.** Este tema se centra en la perspectiva de producto y está relacionado con comprender cómo algunas características particulares del proceso pueden afectar a la calidad del producto o servicio deseado.
  - **Nodo Q.2: Establecimiento de relaciones entre calidad del producto y del proceso.** En concreto, esta tesis doctoral está relacionada con el análisis de los factores que influyen en la calidad de los activos de conocimiento acerca de un proceso de desarrollo de software y sus efectos en la calidad de los productos desarrollados utilizando dichos procesos.
- **Tema E: Ingeniería de Procesos.** Este tema se centra en la perspectiva de proceso y está relacionado con cómo definir y construir procesos y comprender su ejecución.
  - **Nodo E.2: Organización de procesos para reutilización.** En concreto, esta tesis doctoral está relacionada con cómo hacer que un proceso evolucione a partir de la realimentación obtenida de la ejecución del propio proceso.
  - **Nodo E.3: Provisión de infraestructuras para la ingeniería de procesos.** En concreto, esta tesis doctoral está relacionada con cómo crear repositorios de experiencias que permitan que el conocimiento sea almacenado, actualizado y consultado por los desarrolladores.
- **Tema D: Despliegue y Uso del Proceso.** Este tema se centra en la perspectiva de las personas y está relacionado con obtener los procesos adecuados desplegados de manera efectiva en las estructuras organizativas convenientes de manera que se permita a las personas hacer frente a las necesidades de negocio de manera eficiente.
  - **Nodo D.3.2: Ayuda a la adopción efectiva de procesos.** En concreto, esta tesis doctoral está relacionada con la mejora de los procedimientos utilizados para desplegar nuevos procesos en una organización de manera que su adopción por parte de los roles involucrados se realice de manera adecuada.

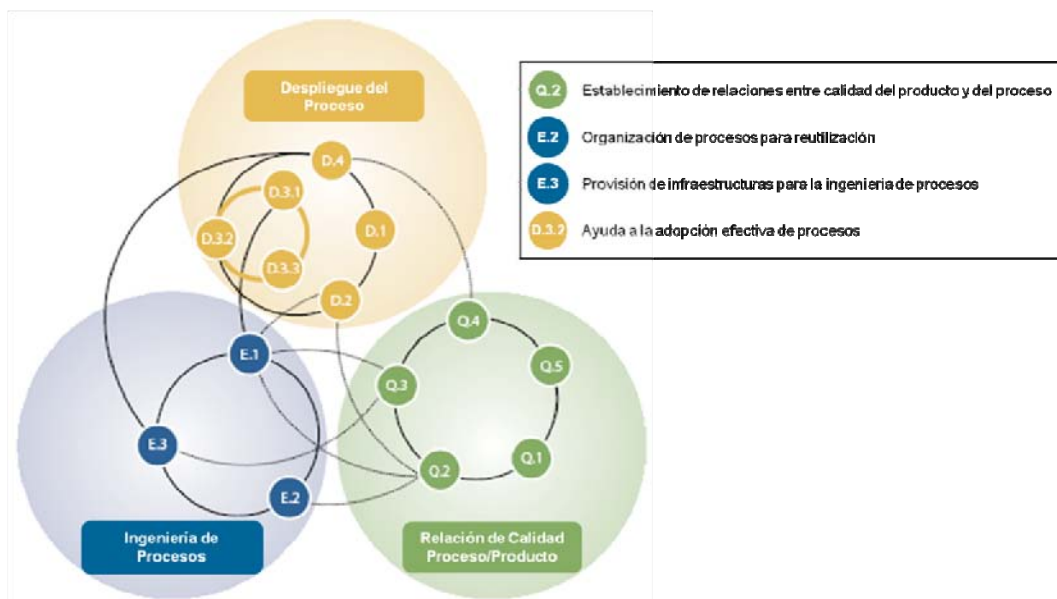


Figura 1.1. Temas y nodos de investigación definidos por el IPRC.

## 1.7. Validez de la solución

La validación llevada a cabo en esta tesis doctoral ayuda a analizar los factores que mejoran la creación, enriquecimiento y diseminación efectiva del conocimiento tácito convirtiendo ITAKA en una fuente de conocimiento relevante para ayudar a la adopción y al aprendizaje de nuevos procesos de software.

El método de investigación elegido para esta validación es el método empírico, por el cual se propone una teoría formal que es evaluada a través de estudios empíricos que se realizan mediante casos de estudio, experimentos, encuestas y entrevistas.

En este apartado se realiza primero una aproximación a la validación de ITAKA y, posteriormente, se discuten las limitaciones para la validez del experimento realizado.

### 1.7.1. Aproximación a la validación de la solución

El proceso de validación de esta tesis doctoral se basa en el planteamiento de las preguntas de investigación que se indican a continuación.

En relación con el análisis de la efectividad de los mecanismos de adquisición de conocimiento tácito para crear un repositorio de conocimiento organizacional (H1.1):

- **PI1:** ¿Constituye ITAKA una aproximación efectiva para capturar el conocimiento tácito generado por los ingenieros de software durante la ejecución de los proyectos y almacenarlo junto con el restante conocimiento organizacional?
- **PI2:** ¿Cuáles son los factores que facilitan el uso de ITAKA como fuente de conocimiento relevante para los ingenieros de software para el aprendizaje y adopción de un nuevo proceso software?

En relación con el análisis de la calidad de los activos de conocimiento (ejemplos, discusiones y lecciones aprendidas) elicitados durante la ejecución de la iniciativa de aprendizaje (H1.2):

- **PI3:** ¿Qué nivel de calidad tiene el nuevo conocimiento elicitado por los ingenieros de software al utilizar ITAKA?
- **PI4:** ¿Cuáles son los factores que influyen en la calidad de los activos de conocimiento generados al utilizar ITAKA?
- **PI5:** ¿Cuál es la efectividad de las distintas alternativas –supervisada y no-supervisada– para la incorporación de los cambios propuestos por los ingenieros de software durante el proceso de gestión de cambios?

En relación con la mejora de la facilidad en el aprendizaje y la adopción de nuevas prácticas para el desarrollo de software dentro de la organización (H1.3):

- **PI6:** ¿Contribuye ITAKA a la mejora de la facilidad del aprendizaje de nuevos procesos software que deban ser adoptados por los ingenieros de software?
- **PI7:** ¿Cuál es la evaluación subjetiva de la utilidad de los mecanismos de gestión de la configuración propuestos para preservar y enriquecer el conocimiento organizacional?

Para tratar de encontrar respuestas a estas preguntas de investigación se llevó a cabo un experimento controlado durante el desarrollo de un curso de formación y aprendizaje de procesos ágiles de desarrollo. Siguiendo un riguroso protocolo para guiar la ejecución de dicho experimento, la validación fue realizada en dos fases tal y como se describe a continuación.

### **Fase 1: PAL sin gestión del cambio ni gestión de la configuración (PAL-Wiki)**

El objetivo de esta fase fue el desarrollo de sesiones prácticas y de un proyecto de desarrollo de software utilizando procesos de desarrollo ágil utilizando como fuente de conocimiento una PAL en forma de wiki que no disponía de ningún mecanismo de realimentación para enriquecer el conocimiento organizacional.

Esta fase se dividió en dos etapas:

- **Etapas de formación:** durante la cual se realizaron sesiones de prácticas para entender los fundamentos y conceptos acerca de los procesos de desarrollo ágil.
- **Etapas de proyecto:** durante la cual etapa se aplicaron los conocimientos obtenidos en la etapa anterior para realizar un proyecto de desarrollo de software.

### **Fase 2: PAL con gestión del cambio y gestión de la configuración (ITAKA)**

El objetivo de esta fase fue la validación de la solución propuesta en esta tesis doctoral. Al igual que la fase anterior, esta fase también se dividió en dos etapas con contenidos análogos:

- **Etapas de formación:** durante la cual se realizaron sesiones de prácticas para entender los fundamentos y conceptos acerca de los procesos de desarrollo ágil.
- **Etapas de proyecto:** durante la cual etapa se aplicaron los conocimientos obtenidos en la etapa anterior para realizar un proyecto de desarrollo de software.

Al finalizar esta fase se comprobó si la utilización de ITAKA ayudó a incrementar la efectividad en la captura, enriquecimiento y diseminación del conocimiento tácito generado por los ingenieros de software durante la ejecución de los proyectos de la organización, convirtiendo ITAKA en una fuente relevante de conocimiento para mejorar la capacidad para la adopción y el aprendizaje de nuevas prácticas para el desarrollo de software dentro de la organización.

## **1.7.2. Limitaciones de la solución propuesta**

Esta sección discute las limitaciones para la validez del experimento realizado en el ámbito de esta tesis doctoral. De acuerdo con [Juristo, 2001] (p.354), esta validez puede ser discutida en términos de validez del constructo, validez interna y validez externa.

### **Validez del constructo**

La validez del constructo se refiere a si las medidas tomadas están correlacionadas con el constructo<sup>1</sup> científico teórico que se afirma estar midiendo, es decir, es el punto hasta el cual se midió realmente lo que se pretendía medir. En el experimento realizado en esta tesis

---

<sup>1</sup> Constructo: construcción teórica para resolver un problema científico determinado.

doctoral, la validez del constructo analiza la precisión de la estrategia del experimento y de las variables consideradas para discutir de forma adecuada las preguntas de investigación formuladas antes.

Los datos considerados para analizar si ITAKA constituye una aproximación efectiva para capturar el conocimiento tácito de los ingenieros de software están basados en las estadísticas proporcionadas por la herramienta Google Analytics, la cual monitoriza las variables consideradas en la definición del experimento. Sin embargo, los instrumentos utilizados por Google Analytics presentan algunas limitaciones que impiden la adquisición de información adicional, como por ejemplo la identificación del usuario que está accediendo a los servicios de la wiki en cada sesión. Esta información hubiera resultado de gran utilidad para analizar con mayor profundidad algunas evidencias relacionadas con el comportamiento de los ingenieros de software que participaron en el experimento. Sin embargo, esta limitación fue parcialmente resuelta mediante el análisis cualitativo de los datos obtenidos por medio de las encuestas realizadas y por las observaciones anotadas a lo largo del experimento.

Las encuestas realizadas durante la ejecución del experimento fueron útiles para analizar la evolución de la actitud y de la aptitud de los ingenieros de software para la utilización de los principios de gestión de la configuración y de gestión del cambio para la gestión del conocimiento organizacional presentados en esta tesis doctoral. Sin embargo, la obtención de información cuantitativa adicional hubiera resultado útil para analizar de manera estadística la influencia de determinados factores humanos, como son la cultura de compartición del conocimiento, las capacidades de trabajo en equipo, o las capacidades de abstracción y de formalización de elementos de conocimiento. La consideración de dichos factores podría haber proporcionado una visión más integral de los factores que contribuyen a la utilización de ITAKA como una fuente relevante del conocimiento organizacional.

Finalmente, la principal amenaza para la validez de la experimentación en relación con el análisis de la contribución de ITAKA para mejorar la adopción y el aprendizaje de nuevos procesos de software podría ser la fiabilidad de la evaluación de la calidad de los productos desarrollados por los distintos equipos de trabajo durante el experimento. Para tratar de minimizar esta amenaza se aplicó un proceso de evaluación multi-fuente por el cual dos expertos evaluaron de manera individual los mismos productos utilizando unos criterios comunes pre-establecidos (ver Anexo A: Criterios para la Evaluación de Productos). Para comprobar el nivel de acuerdo entre los expertos se calculaba el coeficiente Kappa de Cohen con pesos al cuadrado. En caso de haber acuerdo ( $\kappa > 0.9$ ), la calidad del producto se asignaba obteniendo la media de las evaluaciones de ambos expertos; en caso contrario, el producto era re-evaluado de manera conjunta por ambos expertos.

## Validez interna

La validez interna se refiere al grado hasta el cual las afirmaciones de causa-efecto son válidas.

Durante la realización del experimento, la opinión de los investigadores involucrados no fue impuesta a los participantes ya que fueron libres de expresar sus opiniones en todo momento. Sin embargo, resulta prácticamente imposible que los participantes no resulten influenciados

en cierta medida puesto que el simple hecho de guiar el experimento implica la presuposición de las opiniones de los investigadores por parte de los participantes.

En relación con el análisis de la efectividad de ITAKA, los datos fueron recogidos y analizados semanalmente. Esta periodicidad permitió analizar de manera estadística los factores que contribuyeron a la utilización efectiva de los mecanismos de discusión, pero no así los mecanismos relacionados con la creación y distribución de ejemplos y lecciones aprendidas ya que la frecuencia de creación de estos últimos no se producía semanalmente. Esto impidió la identificación de evidencias estadísticas acerca de los factores que contribuyen a la utilización efectiva de los mecanismos de creación y distribución de ejemplos y lecciones aprendidas. Las evidencias cualitativas obtenidas proporcionaron un primer análisis de los factores que influyen en las lecciones aprendidas, pero se podría obtener una mejor perspectiva ajustando las frecuencias de recopilación de datos estadísticos y de creación de ejemplos y lecciones aprendidas.

Por último, la contribución de ITAKA a la mejora de la adopción y aprendizaje de nuevos procesos de software fue analizada mediante la comparación de los niveles de calidad de los productos obtenidos durante las etapas de proyecto de ambas fases del experimento (fase sin ITAKA y fase con ITAKA). Puesto que los participantes en ambas fases presentaban perfiles similares, los resultados obtenidos contribuyeron a analizar de manera adecuada la contribución de ITAKA a la mejora del aprendizaje de nuevos procesos de software, pero se podría obtener un análisis más refinado comparando ambas aproximaciones durante varios años y analizando la evolución de la capacidad de los participantes para adoptar nuevas prácticas eficientes para el desarrollo de software.

### Validez externa

La validez externa limita el grado hasta el cual se pueden hacer generalizaciones a partir de los resultados de un experimento.

A partir de la experimentación de esta tesis doctoral se pueden prever varias limitaciones contextuales en relación al número de participantes, al número y al tipo de proyectos desarrollados, y a la experiencia previa de los participantes. Este experimento no trata de crear una teoría general a partir de las conclusiones del mismo, sino contribuir a la discusión de los factores que influyen en el correcto modelado y en el uso efectivo de patrones de proyecto en contextos comunes del desarrollo de software. Normalmente, las unidades de desarrollo de software trabajan en proyectos que se enmarcan dentro de un conjunto limitado de tipos de proyectos centrados en tecnologías y ámbitos de negocio concretos. Cuando una gran organización quiere diversificar su negocio, habitualmente crea unidades independientes. Es más, las personas que han participado en este experimento tienen perfiles muy similares al personal que trabaja habitualmente en organizaciones TIC de innovación, con poca experiencia en procesos de desarrollo de software y en modelado de la información. Así, las restricciones contextuales mencionadas al comienzo de este párrafo pueden ser consideradas como representativas de un conjunto relevante de organizaciones de ingeniería del software.

En relación con las características de los ingenieros de software que participaron en el experimento realizado también se pueden mencionar algunas limitaciones. La mayoría de los

participantes eran estudiantes, de forma que no son necesariamente representativos de todos los desarrolladores. Sin embargo, algunos de los participantes trabajaban como desarrolladores en el mundo laboral y, en general, sus respuestas ante el experimento estuvieron alineadas con las de los estudiantes.

Finalmente, es importante remarcar que los resultados obtenidos en experimentos que involucran el uso de wikis están habitualmente influenciados por la cultura de compartición del conocimiento existente en la organización en la cual se lleva a cabo el experimento. Los ingenieros de software que participaron en este experimento tenían este tipo de cultura. Los resultados obtenidos demuestran la importancia de este tipo de condiciones de contexto para el éxito de la implementación de las prácticas de gestión del conocimiento que se presentan en esta tesis doctoral. Así, la efectividad de esta clase de aproximaciones sería menor en otros contextos organizacionales que no presentaran esta cultura de compartición del conocimiento.

## 1.8. Estructura de la tesis doctoral

La memoria de esta tesis doctoral está organizada en capítulos de la siguiente manera:

- **Capítulo 1: Introducción.** Es el capítulo actual, en el cual se introduce el contexto del trabajo realizado en esta tesis doctoral, y se describen los problemas encontrados en dicho contexto, así como la motivación para la resolución de los mismos. También se formula la hipótesis que se tratará de validar y los objetivos de investigación. Por último, en este capítulo también se describen brevemente la aproximación a la solución, su interés investigador y su validez.
- **Capítulo 2: Estado de la Cuestión.** En este capítulo se describe el estado de la cuestión realizando una revisión crítica de los trabajos existentes y comparando las soluciones propuestas por otros autores con la solución que se propone en esta tesis doctoral.
- **Capítulo 3: Solución Propuesta.** Este capítulo presenta ITAKA, un marco de trabajo propuesto para el enriquecimiento del conocimiento que ayuda a ingenieros de software que trabajan en distintos equipos al aprendizaje colaborativo de procesos de software gracias al enriquecimiento del conocimiento y a la aplicación de los principios de gestión de la configuración y de gestión del cambio.
- **Capítulo 4: Implementación de la Solución Propuesta.** En este capítulo se describe la implementación realizada para validar experimentalmente ITAKA.
- **Capítulo 5: Validación de la Solución Propuesta.** En este capítulo se detalla la planificación, la ejecución y el análisis de los resultados obtenidos en la validación experimental realizada en el ámbito de esta tesis doctoral.
- **Capítulo 6: Conclusiones y Futuras Líneas de Investigación.** Este capítulo presenta las conclusiones extraídas de la realización de esta tesis doctoral y se plantean las posibles líneas futuras que se pueden abordar en relación a esta propuesta de investigación.
- **Capítulo 7: Referencias.** En este capítulo se listan las distintas fuentes bibliográficas consultadas y utilizadas para el desarrollo de esta tesis doctoral.





# Capítulo 2:

## Estado de la Cuestión

2.1. Gestión del Conocimiento .....	17
2.1.1. Dimensiones del conocimiento .....	18
2.1.2. Ciclos de vida del conocimiento .....	18
2.1.3. Sistemas de Gestión del Conocimiento .....	20
2.1.4. Tecnologías Web 2.0 para gestionar el conocimiento .....	22
2.2. Gestión del Conocimiento en Ingeniería del Software .....	28
2.2.1. Proceso de software .....	29
2.2.2. Mejora del Proceso de Software .....	30
2.2.3. Gestión del Conocimiento en la Mejora de Procesos de Software .....	34
2.2.4. Bibliotecas de Activos de Procesos.....	34
2.3. Soluciones existentes para la gestión del conocimiento acerca de procesos de software .....	45
2.3.1. Spearmint .....	45
2.3.2. WAGNER.....	47
2.3.3. Microsoft Solutions Framework (MSF).....	48
2.3.4. ProKnowHow .....	49
2.3.5. OnSSPKR .....	50
2.3.6. Marco Metodológico basado en Patrones de Producto (MMPP) .....	51
2.3.7. PAL-Wiki.....	52
2.3.8. Experience Factory (EF) .....	53
2.3.9. Proceso de Realimentación de Experiencias (PRE).....	54
2.3.10. KiWi Systems.....	54
2.3.11. Eclipse Process Framework (EPF) .....	55
2.3.12. IRIS Process Author (IPA).....	57
2.4. Análisis de soluciones existentes y conclusiones.....	58
2.5. Conclusiones del estado de la cuestión .....	63



La presente tesis doctoral se apoya dos áreas de investigación: la Ingeniería del Software y la Gestión del Conocimiento. Respecto a la Ingeniería del Software, la aportación de esta tesis doctoral se centra en la gestión de procesos de software y en la mejora de procesos de software. En cuanto a la Gestión del Conocimiento, esta tesis doctoral propone la utilización de esta disciplina dentro de la Ingeniería del Software como herramienta que ayude a la gestión y a la mejora de los procesos de software.

En los siguientes apartados se introducen estos temas y se analizan algunos de los trabajos más relevantes publicados en relación con la aplicación de la Gestión del Conocimiento para la mejora de los procesos de software. El capítulo finaliza con el análisis crítico de estos trabajos publicados y las conclusiones extraídas a partir de los problemas que dichas soluciones existentes presentan en relación con la gestión y mejora de procesos de software.

## 2.1. Gestión del Conocimiento

El conocimiento es uno de los activos con mayor valor dentro del entorno empresarial actual. La necesidad de diferenciación lleva a las empresas a gestionar su conocimiento organizacional existente. Cuando es gestionado de manera adecuada, el conocimiento se convierte en un recurso clave para muchas organizaciones ya que puede ayudar a mejorar su competitividad [Choi, 2008].

Desde la Grecia antigua, muchos son los autores que han buscado una definición para el término “conocimiento”. Platón estableció que, para conocer algo, es necesario cumplir tres condiciones: se cree en ese algo, ese algo es verdad, y uno está justificado en creer en ese algo. Ikujiro Nonaka adoptó esta definición de conocimiento entendido como *“una creencia justificada en algo cierto”* [Nonaka, 1994], añadiendo posteriormente que *“las personas justifican la veracidad de sus creencias en base a sus interacciones con el mundo”* [Nonaka, 2006]. Así, se puede reconocer que alguien tiene conocimiento sobre una tarea a través de cómo ejecuta dicha tarea y/o cómo define una situación para permitir su ejecución. De esta manera, el conocimiento permite al ser humano definir, preparar, dar forma y aprender a ejecutar tareas o resolver problemas [Nonaka, 2009].

Cabe destacar que, a menudo, los términos “conocimiento” e “información” son utilizados indistintamente en las organizaciones. Davenport y Prusak realizan una distinción clara entre ambos elementos [Davenport, 2000]. La información es un conjunto de datos organizados y relacionados que tienen un significado (relevancia y propósito) dentro de un contexto. El conocimiento es sin embargo información con un valor agregado. El conocimiento es una mezcla de experiencia, apreciaciones, interpretaciones e información contextual que sirve como marco para la incorporación de nuevas experiencias e información.

A mediados de los 90, las organizaciones empezaron a considerar seriamente la posibilidad de gestionar su conocimiento puesto que dicho conocimiento puede estar registrado no solo en documentos o repositorios, sino también en los procesos de la organización, prácticas, rutinas y normas. Este movimiento fue bautizado como Gestión del Conocimiento.

La Gestión del Conocimiento (*Knowledge Management – KM*) se puede definir como la disciplina que estudia la creación, preservación, aplicación y reutilización del conocimiento

disponible en una organización, siendo su objetivo la creación de conocimiento compartido entre todos los usuarios [Alavi, 2001]. El KM simplifica el proceso de compartición, distribución, creación, captura, y comprensión del conocimiento de una organización [Dingsøyr, 2009].

El KM es una gran fuente de poder dentro de nuestra sociedad actual ya que permite a las personas administrar la información necesaria para tomar decisiones. El conocimiento, por tanto, proporciona la capacidad de actuar y de producir resultados dentro de las organizaciones, siendo el descubrimiento de nuevo conocimiento una fuente de innovación.

### 2.1.1. Dimensiones del conocimiento

Hasta ahora, en este capítulo se ha presentado qué es el conocimiento y por qué es importante su gestión, pero no se ha tratado cómo se puede crear conocimiento y cómo se puede gestionar dicho proceso de creación de conocimiento.

Una de las dimensiones de este proceso de creación del conocimiento puede ser descrita a partir de la distinción entre dos tipos de conocimiento: tácito y explícito. Esta distinción entre conocimiento tácito y explícito fue inicialmente propuesta por el filósofo Michael Polanyi [Polanyi, 1966], aunque fue aplicada en el campo de los negocios y la gestión del conocimiento por el japonés Ikujiro Nonaka [Nonaka, 1994].

El conocimiento tácito es muy personal, difícil de formalizar y de comunicar, intuitivo y derivado de la experiencia y de las creencias. El conocimiento explícito, sin embargo, es formal y sistemático, y puede ser expresado sin ambigüedades por medio de textos, dibujos, bases de datos, etc. [Assimakopoulos, 2006].

Un ejemplo muy claro de la diferencia entre conocimiento tácito y explícito es el caso de cómo se monta en bicicleta. El conocimiento explícito nos dice que debemos pedalear para avanzar y utilizar el manillar para controlar la dirección, pero es el conocimiento explícito el que nos permite manejar la bicicleta mientras mantenemos el equilibrio sobre la misma.

A pesar de que el conocimiento tácito es utilizado por todo el mundo, no es sencillo de expresar y de compartir. Está formado por creencias, ideales, valores y modelos mentales profundamente arraigados en las personas [Nonaka, 2009]. Su adquisición suele realizarse sin utilizar el lenguaje sino por medio de la observación, la imitación y la práctica, siendo la experiencia la clave para su adquisición [Ivarsson, 2012].

### 2.1.2. Ciclos de vida del conocimiento

Aunque el conocimiento puede ser tácito o explícito, ambas formas no están separadas sino que son mutuamente complementarias ya que interactúan dinámicamente la una con la otra en las actividades creativas realizadas por las personas [Alavi, 2001] [Nonaka, 1994]. Los activos de conocimiento, por tanto, no son recursos estáticos sino que sufren cambios en función de la etapa del ciclo de vida en la que se encuentren. Las organizaciones que quieran gestionar y utilizar el conocimiento de manera eficiente deben tratar sus activos de conocimiento de acuerdo con las etapas de vida de los mismos [Freeze, 2011].

Varios investigadores en el área del KM han propuesto distintos puntos de vista sobre estas etapas. Como ejemplos de ciclos de vida más relevantes se presentan a continuación el modelo SECI de Nonaka y Takeuchi, el ciclo de vida del conocimiento de Birkinshaw y Sheehan, y el ciclo de evolución del conocimiento de Rus y Lindvall.

### Modelo SECI de Nonaka y Takeuchi

La teoría de la creación de conocimiento propuesta por Nonaka en [Nonaka, 1994] y más recientemente revisada en [Nonaka, 2009] sugiere que el conocimiento alterna entre sus formas tácita y explícita, de manera que el conocimiento tácito puede llevar a la creación de conocimiento explícito y viceversa. Esta conversión del conocimiento ocurre en cuatro procesos (socialización, externalización, combinación e internalización), bautizados por Nonaka y Takeuchi como “Modelo SECI” [Nonaka, 1995].

La **socialización** es el proceso de creación de conocimiento tácito a través de las experiencias compartidas entre las personas. La **externalización** es el proceso por el cual se trata de expresar el conocimiento tácito mediante conceptos explícitos. La **combinación** es el proceso por el cual se crea nuevo conocimiento explícito mediante la unión de diferentes entidades de conocimiento explícito. La **internalización** es el proceso mediante el cual las personas asimilan el conocimiento explícito convirtiéndolo en conocimiento tácito. La Figura 2.1 muestra gráficamente estos cuatro procesos de transformación del conocimiento entre sus formas tácita y explícita.

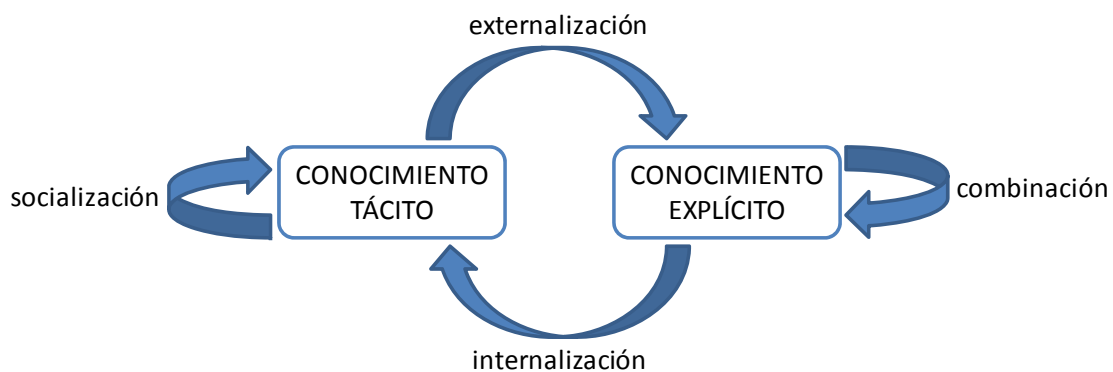


Figura 2.1. Modelo SECI.

Como se puede observar, este es un ciclo de vida continuo lo cual permite que el conocimiento evolucione y sea enriquecido de manera reiterada, algo que no ocurre con la mayoría de los ciclos de vida que se presentan a continuación. Debido a esta característica y a su sencillez, el modelo SECI será el que se tomará como base para la definición de los distintos procesos que forman parte del marco de trabajo ITAKA propuesto en esta tesis doctoral.

### Ciclo de vida del conocimiento de Birkinshaw y Sheehan

Julian Birkinshaw y Tony Sheehan proponen en [Birkinshaw, 2002] un ciclo de vida del conocimiento que sigue una curva en forma de “S” (Figura 2.2) de forma que un determinado conocimiento está disponible a un grupo de personas cada vez mayor con el paso del tiempo.

La etapa de **creación** es caótica y se caracteriza por que nadie comprende por completo el cuerpo del conocimiento que está surgiendo, ni siquiera aquellos que lo están creando. El conocimiento que genera interés adquiere una forma más clara y pasa a la etapa de **movilización**, en la cual los creadores de este conocimiento lo comparten con personas dentro de un círculo de confianza para refinarlo y extraer valor de él. Posteriormente, el conocimiento pasa a la etapa de **difusión** en la cual dicho conocimiento se extiende fuera del círculo de confianza y es comprendido por una amplia audiencia. Finalmente, la etapa de **mercantilización** se centra en gestionar un conocimiento que ya es de sobra conocido y que pasa a ser un bien común.

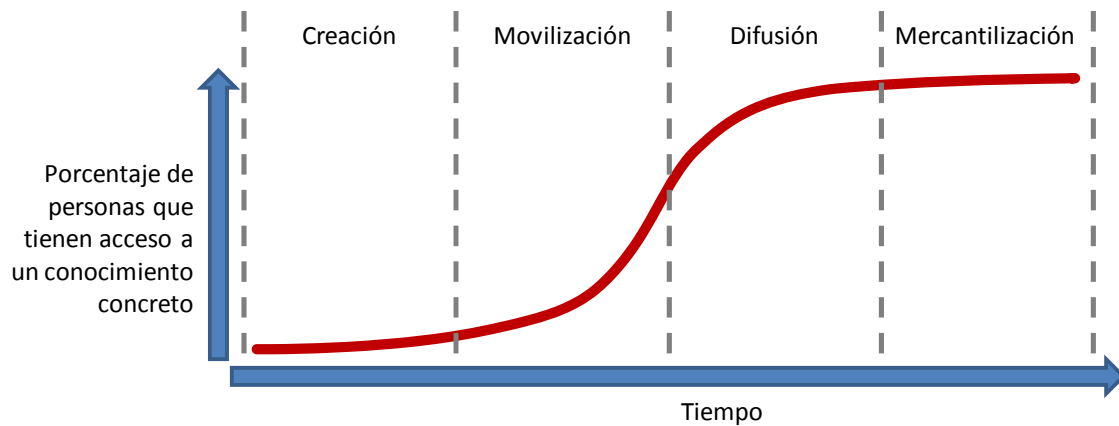


Figura 2.2. Ciclo de vida del conocimiento de Birkinshaw y Sheehan.

### Ciclo de evolución del conocimiento de Rus y Lindvall

Ioana Rus y Mikael Lindvall proponen en [Rus, 2002] un ciclo de evolución del conocimiento que define las distintas fases por las que pasa el conocimiento organizacional.

En la fase de **creación**, los miembros de la organización originan conocimiento por medio del aprendizaje, resolución de problemas, innovación, creatividad, e importación de otras fuentes. Después, en la fase de **captura**, las personas adquieren y capturan información acerca del conocimiento de forma explícita. En la fase de **organización**, la empresa transforma, organiza o incluye conocimiento en el material escrito y en las bases de conocimiento. Este conocimiento es distribuido en la fase de **despliegue** por medio de la formación de empleados, sistemas automáticos basados en el conocimiento o redes de expertos. Finalmente, la fase de **aplicación** es la más importante ya que el objetivo final de la empresa es hacer que este conocimiento esté disponible y sea aplicable cuando sea necesario.

#### 2.1.3. Sistemas de Gestión del Conocimiento

En las organizaciones con una cultura que promueve el aprendizaje continuo y fomenta el intercambio de experiencias [Bjørnson, 2008] se produce la creación de nuevo conocimiento organizacional durante el ciclo de vida del mismo. El propio Nonaka define la creación de conocimiento a nivel organizacional como el proceso por el cual el conocimiento creado por las personas se pone a disposición de la empresa, se amplifica y, además, se materializa y es conectado con un sistema de gestión del conocimiento de la organización [Nonaka, 2009]. En esta sección se presentan estos sistemas y se analizan las características de los mismos.

Un Sistema de Gestión del Conocimiento (*Knowledge Management System – KMS*) es una plataforma tecnológica de información y comunicaciones cuyo propósito no es únicamente almacenar información sino promover la gestión del conocimiento de forma efectiva y eficiente [Maier, 2007].

Para que el KMS sea efectivo y además se consiga la aceptación por parte de los miembros de la organización, el KMS debe estar integrado dentro de los procesos de la organización y dentro del entorno de trabajo, permitiendo recolectar y almacenar el conocimiento relevante al tiempo que va siendo generado con el trabajo [Natali, 2002].

Estos KMSs se pueden clasificar por funciones o por categorías. Según [Maier, 2007], las funciones que desempeña un KMS se pueden dividir principalmente en integradoras, las cuales se enfocan hacia la transferencia asincrónica de conocimiento explícito entre los participantes, y en interactivas, las cuales se enfocan hacia la comunicación y colaboración entre expertos y usuarios del conocimiento. De esta forma, atendiendo a la funcionalidad del KMS se puede diferenciar entre:

- Sistemas que proporcionan funcionalidades básicas integradoras (correo electrónico, listas de distribución) e interactivas (publicación, búsqueda y recuperación de documentos).
- Sistemas integradores, que soportan la codificación del conocimiento y la búsqueda y recuperación de conocimiento, así como la administración de repositorios de conocimiento y la organización de estructuras de conocimiento.
- Sistemas interactivos, que soportan la compartición directa o el desarrollo conjunto de conocimiento entre expertos y usuarios del conocimiento, o entre proveedores y buscadores de conocimiento.
- Sistemas de puente, que combinan funciones integradoras e interactivas para proporcionar repositorios de conocimiento altamente contextualizados, enfocados a conectar proveedores con buscadores de conocimiento, a unir participantes con perfiles similares, a hacer recomendaciones, y a filtrar y presentar elementos de conocimiento y enlaces de manera personalizada.

Por otro lado, clasificando los KMSs por categorías [Handzic, 2004] se puede distinguir:

- Sistemas de compartición de conocimiento: como portales, intranets, extranets y tecnologías colaborativas que facilitan la interacción entre personas con el propósito de compartir y realizar un aprendizaje colectivo.
- Herramientas de descubrimiento de conocimiento: incluye motores de búsqueda y la utilización de agentes inteligentes para encontrar información relevante.
- Repositorios de conocimiento: bases de datos que permiten capturar, codificar, organizar y almacenar el conocimiento relevante en la organización para su uso posterior.
- Sistemas de soporte al aprendizaje: fomentan la innovación empresarial por medio de juegos mentales y realidad virtual.

Una primera aproximación para lograr una plataforma de permita gestionar de manera efectiva el conocimiento acerca de los procesos de una organización es obtener una

herramienta que soporte las funcionalidades integradoras e interactivas básicas de gestión del conocimiento y las capacidades de las herramientas de colaboración, descubrimiento y búsqueda, y de repositorio de conocimiento.

En cuanto a las ventajas de los KMSs, algunas de ellas se indican a continuación:

- Compartición de información valiosa dentro de la organización, facilitando así su reutilización en diferentes proyectos [Petter, 2008].
- Reducción de trabajo redundante, especialmente evitando repetir errores gracias a la reutilización de experiencias obtenidas en proyectos anteriores [Kamsu-Foguem, 2008] [He, 2007].
- Reducción del tiempo de aprendizaje para nuevos empleados [Maier, 2007].
- Retención de la propiedad intelectual cuando un empleado deja de pertenecer a la organización (si dicho conocimiento puede ser codificado) [Petter, 2008].
- Ayuda en la toma de decisiones en base a experiencias exitosas previas [Rus, 2002].

Para explotar estas ventajas de los KMSs se pueden utilizar tecnologías Web 2.0. El uso de la Web 2.0 como plataforma tecnológica para el KM permite la construcción colaborativa del conocimiento acerca de los procesos de una organización, potenciando las competencias organizacionales, y fomentando las experiencias enriquecedoras de los usuarios que participan en dichos procesos. En numerosas ocasiones, los usuarios del proceso se sienten frustrados por las dificultades que encuentran a la hora de aportar sus conocimientos a un repositorio [Desouza, 2003]. El uso de las tecnologías Web 2.0 permitirá aplicar estrategias de implementación que reduzcan la resistencia al cambio y motiven al personal en la transición hacia la mejora de los procesos de la organización.

#### **2.1.4. Tecnologías Web 2.0 para gestionar el conocimiento**

La Web 2.0 es un conjunto de tendencias económicas, sociales y tecnológicas que colectivamente forman la base de la nueva generación de Internet, un medio distintivo más maduro caracterizado principalmente por la participación de los usuarios y la transparencia [Musser, 2006].

Entre las ventajas que ofrece la Web 2.0 destacan las siguientes [Murugesan, 2007]:

- Facilita el diseño flexible de la Web, la reutilización creativa de sus contenidos y las actualizaciones.
- Proporciona una interfaz de usuario enriquecida y más receptiva.
- Facilita la creación y modificación colaborativa de contenidos.
- Permite la creación de nuevas aplicaciones reutilizando y combinando diferentes aplicaciones o combinando información de diferentes fuentes.
- Facilita el establecimiento de relaciones entre personas con intereses comunes.
- Permite la colaboración y ayuda a recoger el conocimiento colectivo.

La Web 2.0 ofrece una infraestructura de tecnologías y servicios más dinámica, interactiva y colaborativa que su predecesora, la Web 1.0, haciendo énfasis en la interacción social. La Web 2.0 representa un cambio en la forma de usar la Web, pasando de la limitación de solo poder



ver de forma pasiva sitios web creados por unos pocos expertos a sitios web en los que cualquier persona puede contribuir de forma activa [Lin, 2007].

Algunas de las nuevas tecnologías que abarca la Web 2.0 se describen resumidamente a continuación:

- Weblogs o blogs: Páginas web que contienen anotaciones realizadas por los usuarios (conocidas como “entradas” o “posts”) en orden cronológico inverso (primero la más reciente) acerca de un tema concreto.
- Podcasts: Ficheros de sonido que se distribuyen mediante suscripción a los oyentes.
- Videoblogs: Ficheros de vídeo que se distribuyen mediante suscripción.
- Wikis: Sitios web cuyos contenidos son creados por los propios usuarios y que pueden ser posteriormente editados por cualquiera que tenga acceso a ellos.
- Aplicaciones en línea: Aplicaciones software que no necesitan instalación en una plataforma particular ya que se ejecutan a través de un navegador web.
- Redes sociales: Comunidades online que permiten a las personas relacionarse y comunicarse de forma directa y transparente con otros usuarios sobre temas tanto personales como profesionales.

Una de estas tecnologías Web 2.0 que puede apoyar la gestión del conocimiento acerca de los procesos de una organización es la wiki [Chau, 2005]. Una wiki es un software basado en web que permite a todos los lectores de una página web modificar los contenidos de dicha página usando el propio navegador web [Ebersbach, 2008].

Entre las principales características de las wikis se encuentran:

- Lenguaje propio de marcado: Proporciona una forma abreviada de dar formato al texto y de crear enlaces entre los documentos y los contenidos.
- Estructura y navegación sencillas: Los participantes pueden crear y enlazar páginas fácilmente. Puesto que la jerarquía y la estructura del sitio son bastante planas, la navegación es simple y se puede realizar de forma no lineal.
- Mecanismo simple de plantillas: Cuando se solicita una página de una wiki, el lenguaje de marcado se transforma en lenguaje HTML, utilizando una plantilla para ofrecer una imagen uniforme y consistente para todas las páginas de la wiki.
- Flujo de trabajo simple: Se pueden editar y publicar contenidos sin necesidad de una revisión y/o aprobación editorial exhaustiva. Los contenidos son gestionados por medio de la monitorización de cambios y el control de versiones.

Las wikis son plataformas de generación colaborativa de contenidos basados en la Web muy simples y sencillos de manejar. Cualquier persona que tenga acceso a la wiki puede crear nuevos contenidos o modificar los ya existentes. Por otro lado, por medio de un sencillo sistema de control de versiones, los administradores pueden rastrear los cambios hechos en las páginas de la wiki [Solís, 2009].

La wiki es por tanto una herramienta apropiada para gestionar conocimiento acerca de los procesos de una organización. Sin embargo, no todas las herramientas para la creación de wikis ofrecen las mismas funcionalidades y características, por lo que es conveniente

determinar qué software es el más adecuado. A priori no existe un software que sea mejor que el resto, por lo que se deberá elegir el que sea más apropiado para el propósito de esta tesis. Para ello se van a analizar cuatro de las wikis más populares: DokuWiki, MediaWiki, TWiki y Deki Wiki.

### ***DokuWiki***

Es una wiki sencilla de manejar orientada a la creación de documentos. Es apropiada para equipos de desarrollo, grupos de trabajo y pequeñas empresas. Su sintaxis es sencilla pero potente y no requiere de una base de datos puesto que almacena los datos en ficheros de texto plano. El software fue creado por Andreas Göhr en junio de 2004, y desde su inclusión en distribuciones Linux en 2005 ha experimentó un rápido crecimiento en popularidad.

DokuWiki cuenta con un control de acceso basado en usuarios y grupos de usuarios, y una lista de control de acceso en la que el administrador puede definir permisos a nivel de página o a nivel de grupos de páginas.

Esta wiki mantiene un control de revisiones de forma que almacena todas las versiones de cada página. Además, la edición en paralelo de una misma página por parte de varios usuarios se evita mediante un mecanismo de bloqueo.

### ***MediaWiki***

Es un motor para wikis creado y desarrollado por Magnus Manske en 2003 para Wikipedia y otros proyectos de la fundación Wikimedia (Wikcionario, Wikilibros, etc). Ha tenido una gran expansión desde el año 2005, existiendo un gran número de wikis basadas en este software que nada tienen que ver con dicha fundación, aunque sí comparten la idea de la generación de contenidos de manera colaborativa. Sus principales usos dentro de las empresas suelen ser como gestor de conocimiento y como sistema de gestión de contenidos.

Debido al énfasis en el multilingüismo dentro de los proyectos de Wikimedia, la internacionalización y la localización han recibido mucha atención por parte de los desarrolladores de esta wiki. Así, la interfaz de usuario ha sido traducida completa o parcialmente a más de 300 idiomas, y puede ser personalizada aún más por parte de los administradores (la interfaz entera es personalizable a través de la wiki).

Una de las principales diferencias de MediaWiki es que no usa enlaces de tipo CamelCase, un estilo de escritura que se aplica a frases o palabras compuestas y que consiste en poner en mayúscula la primera letra de cada palabra. MediaWiki utiliza dobles corchetes para marcar los enlaces, de forma que se evita el uso del CamelCase y se permite la inserción de espacios en dichos enlaces.

MediaWiki no proporciona una interfaz WYSIWYG ya que está actualmente en desarrollo, aunque proporciona una barra de herramientas para simplificar el proceso de edición de contenidos. Mientras se incorpora esta funcionalidad, se puede utilizar en su lugar un plugin que proporciona esta interfaz WYSIWYG.

### ***TWiki***

Es una wiki estructurada implementada en Perl, flexible, potente y fácil de manejar. Esta es una wiki empresarial que se suele utilizar como espacio de desarrollo de proyectos, sistema de gestión de documentos, base de conocimiento, o como cualquier otra herramienta de trabajo colaborativo. El proyecto TWiki fue fundado por Peter Thoeny en 1998, surgiendo TWiki inicialmente como una base de conocimiento para dar soporte a clientes.

TWiki proporciona un control de acceso de grano fino basado en grupos de usuarios, con restricciones de lectura, escritura o renombrado a nivel de sitio, nivel de web o nivel de página.

La edición de contenidos se realiza con un editor WYSIWYG, las páginas se almacenan en ficheros de texto plano, y la generación de contenidos se hace de forma dinámica utilizando variables.

### ***Deki Wiki***

Es una plataforma colaborativa de la empresa MindTouch que proporciona una forma sencilla de crear, organizar, agregar y compartir información. Esta herramienta permite desarrollar sitios web, comunidades online, intranets y sistemas de colaboración. La aplicación está formada por una interfaz y una API. La interfaz más habitual es web y está desarrollada en PHP, aunque existen también interfaces de escritorio para Windows, para publicar documentos desde Outlook, e incluso una interfaz para iPhone. Por otro lado, la API está desarrollada en C# y facilita además a los desarrolladores la personalización e integración con otras aplicaciones de empresa existentes y con servicios web. Aunque el software es gratuito y de código abierto, MindTouch también proporciona una versión comercial.

Toda la información de esta wiki se almacena en XML. Sus páginas pueden ser modificadas usando un editor WYSIWYG o pueden ser manipuladas como un servicio web XML. Deki Wiki soporta autenticación tanto interna como externa, y los permisos pueden ser aplicados tanto a páginas individuales como a categorías completas.

Por otro lado, esta wiki cuenta con uno de los sistemas de permisos más avanzados. Los administradores pueden hacer que la wiki sea privada o pública, anónimas o no. La gestión se puede hacer tanto a nivel de usuarios como de grupos de usuarios. Los permisos se pueden aplicar tanto a páginas individuales como a una jerarquía completa de páginas para crear espacios de trabajo privados o no editables.

### ***Comparativa de herramientas para la creación de wikis***

La Tabla 2.1 resume las principales características de las cuatro wikis comentadas anteriormente, lo cual permite realizar una comparación entre ellas.

Tabla 2.1. Comparativa de wikis.

<b>Características generales</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Coste	Gratuita	Gratuita	Gratuita	Gratuita y comercial
Lenguaje de programación	PHP	PHP	Perl	PHP, C#
Almacenamiento de datos	Ficheros de texto	Base de Datos (MySQL)	Ficheros de texto, control de versiones (opcional)	Base de Datos (MySQL)
Estado de desarrollo	Maduro	Maduro	Maduro	Maduro
Destinatarios principales	PYMES	Usuarios finales, educación	Grandes empresas, PYMEs	Desarrolladores, profesionales TIC
<b>Requisitos de sistema</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Sistemas Operativos	Linux, UNIX, Windows, Mac	Linux, UNIX, Windows, Mac	Linux, Windows, Mac	Linux, Windows, Mac
Acceso como root	NO	NO	NO	SI
Servidor web	Cualquiera con soporte PHP	Cualquiera con soporte PHP	Típicamente Apache	Apache, IIS
<b>Seguridad</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Permisos de página	SI	SI	SI	SI
Listas de Control de Acceso	SI	NO	SI	SI
<b>Características sintácticas</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Etiquetas HTML permitidas	Opcional	Algunas	Todas	Todas
Fórmulas matemáticas	Plugin	SI	Plugin	SI
Tablas	Simples	Simples y complejas	Simples y complejas	Simples y complejas
<b>Usabilidad</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Edición por secciones	SI	SI	Plugin	SI
Plantillas de página	SI	SI	SI	SI
Barra de herramientas	SI	SI	Plugin	SI
Edición WYSIWYG	Plugin	Plugin	SI	SI
Atajos de teclado	SI	SI	SI	SI
Auto-firma de páginas	SI	SI	SI	SI
<b>Estadísticas</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Cambios recientes	SI	SI	SI	SI
Páginas enlazadas no existentes	Plugin	SI	NO	NO
Páginas existentes no enlazadas	Plugin	SI	Plugin	NO
Páginas más/menos populares	NO	SI	SI	SI
Lista de visitas	NO	Plugin	NO	SI
<b>Salidas</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
HTML	XHTML 1.0 Transitional	XHTML 1.0 Transitional	XHTML 1.0 Transitional	XHTML 1.1
Hojas de estilo CSS	SI	SI	SI	SI
Impresión	Imprime CSS	Imprime CSS	Imprime vista	Imprime vista
Versión para móviles	NO	NO	Plugin	Plugin
Skins personalizables	SI	SI	SI	SI
RSS	SI	SI	SI	SI
Tabla de contenidos automática	SI	SI	SI	SI
Exportar como HTML	SI	SI	SI	SI
Exportar como XML	NO	SI	Plugin	SI
Exportar como PDF	Plugin	Plugin	Plugin	SI

<b>Multimedia y ficheros</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Ficheros adjuntos	SI	SI	SI	SI
Revisiones de multimedia	NO	SI	SI	SI
Flash embebido	SI	Plugin	Plugin	SI
Vídeo embebido	Plugin	Plugin	Plugin	SI
Búsqueda multimedia	Nombres de ficheros	Palabras clave	Contenidos	Contenidos
<b>Otras características</b>	<b>DokuWiki</b>	<b>MediaWiki</b>	<b>TWiki</b>	<b>Deki Wiki</b>
Historial de cambios	SI	SI	SI	SI
Comentarios de páginas	Plugin	Página de discusión aparte	Al final de la página	Directamente en la página
División por categorías	Plugin	SI	SI	SI
Búsqueda	SI	SI	SI	SI
Enlaces tipo CamelCase	Opcional	NO	SI	NO
Sistema de tickets	NO	NO	Plugin	Plugin

En primer lugar, MindTouch ofrece toda una plataforma colaborativa con una orientación principalmente comercial y profesional. Deki Wiki ofrece posibilidades de integración con otros entornos corporativos, aunque no existen muchas extensiones desarrolladas previamente. Por ello, aunque Deki Wiki parece ser una herramienta potente para la gestión del conocimiento, no parece adaptarse bien a las necesidades de esta tesis.

Por otro lado, las características de TWiki hacen que sea una herramienta atractiva y prometedora. Esta wiki escala razonablemente bien y cuenta con una gran cantidad de documentación detallada disponible. Su principal inconveniente es que la curva de aprendizaje es muy abrupta, con un proceso de instalación y configuración bastante complejo, aunque está disponible una versión empaquetada que puede ser ejecutada sobre una máquina virtual.

DokuWiki por el contrario es muy sencilla de instalar y de manejar, y cuenta con gran cantidad de documentación. Sin embargo, carece de algunas características que serían útiles para esta tesis, como la creación de tablas complejas, el control de versiones de ficheros o la posibilidad de exportar datos en formato XML.

Finalmente está MediaWiki, posiblemente la wiki más familiar para los usuarios gracias a la Wikipedia. Es una de las wikis más sencillas de instalar, y su configuración es sencilla a pesar de no disponer de un panel de control para su administración. La wiki ofrece buen rendimiento y escala bien, aunque su mayor inconveniente es su limitado control de acceso con respecto a otras wikis ya que está orientada a compartir conocimiento.

Una de las ventajas de MediaWiki es que cuenta con páginas de discusión aparte, de forma que se puede ocultar una discusión o permitir que los usuarios solo puedan editar estas páginas de discusión, así como incluir o no las discusiones dentro de las búsquedas.

MediaWiki cuenta también con una gran comunidad de desarrolladores, lo cual hace que haya una gran cantidad de extensiones disponibles o en desarrollo. Por ejemplo, existe una extensión que permite añadir contenido semántico a la wiki, haciendo que determinado texto de una página fuera como si formara parte de un campo en una base de datos, y permitiendo obtener nuevas páginas con determinados contenidos de la wiki mediante consultas.

Finalmente, MediaWiki proporciona un mecanismo para importar datos a la wiki por medio de un archivo XML. Esto podría facilitar la introducción del conocimiento inicial en la wiki de una forma sencilla si previamente se almacena dicho conocimiento pre-existente en un fichero XML con el formato adecuado.

Por los motivos antes mencionados y tras comparar las distintas wikis, las dos mejores opciones parecen ser MediaWiki y TWiki. Por sencillez y por la alta disponibilidad de extensiones, se ha optado por utilizar MediaWiki como herramienta para esta tesis.

## 2.2. Gestión del Conocimiento en Ingeniería del Software

El KM es una disciplina que se aplica en un amplio conjunto de campos tales como la gestión de negocios, los sistemas de información o la educación, y más recientemente en campos como la salud pública o la ingeniería del software. Es en este último en el cual se centra este apartado.

El software es una parte crítica dentro de nuestra sociedad actual y es el núcleo de cualquier producto o servicio moderno. Por este motivo, durante estos últimos años ha habido una gran cantidad de investigación y de desarrollos destinados al estudio y desarrollo de principios, conceptos, métodos y tecnologías para crear, desarrollar y mantener productos de software. Estas innovaciones han creado de forma incremental un conocimiento, unos métodos y unas tecnologías que reciben el nombre de Ingeniería del Software [Fuggetta, 2005].

El objetivo final de la Ingeniería del Software es ayudar a los desarrolladores a crear y trabajar con soluciones software de manera efectiva y eficiente. Esto quiere decir que un ingeniero de software debe tener el apoyo de un cierto número de métodos prácticos que guíen sus actividades, y de tecnologías específicas que implementen dichos métodos [SWEBOK, 2004].

Puesto que la Ingeniería del Software utiliza intensamente el conocimiento, cabe esperar un gran impacto del KM dentro de este campo [Dingsøyr, 2009]. De acuerdo con [Kneuper, 2002], el KM puede dar soporte a algunas tareas necesarias en el uso de procesos de software, tales como definición del proceso, introducción del proceso en el trabajo diario, institucionalización del proceso, mejora y adaptación del proceso, y ajuste e instanciación del proceso para un proyecto concreto.

El KM juega un papel tan importante en las organizaciones de software que muchos académicos y profesionales consideran esta disciplina como uno de los factores más importantes en la producción de software [Ivarsson, 2012] [Colomo-Palacios, 2011]. Los beneficios de la aplicación de los principios del KM en el desarrollo de software incluyen, entre otros, una reducción del tiempo y del coste de desarrollo, un incremento de la calidad, y un incremento de las capacidades de toma de decisiones [Dingsøyr, 2009].

Para sacar provecho del KM, las iniciativas específicas en este área deben ser adaptadas de manera adecuada al contexto de la ingeniería del software en las organizaciones, que habitualmente están formadas por varios equipos trabajando en diferentes áreas de negocio y con distintas tecnologías. Cada unidad organizativa está habitualmente especializada en un conjunto concreto de tecnologías, gestiona su propia infraestructura de desarrollo de software

y puede estar incluso separada geográficamente del resto. Sin embargo, también es habitual que distintas unidades organizativas compartan procesos comunes dentro de la organización.

En las organizaciones de ingeniería del software, el KM puede ser utilizado para capturar las experiencias generadas durante la ejecución de los procesos organizacionales [Natali, 2002], así como para la distribución del conocimiento tácito y explícito dentro de la organización [Aaen, 2003]. Aunque cada proceso es único en cierto sentido, experiencias similares pueden ayudar a los ingenieros de software a ejecutar mejor sus tareas y a capturar el conocimiento que estas personas adquieren durante su trabajo [Scorta, 2009]. Es más, el KM es considerado actualmente una parte esencial en el proceso de formación de los ingenieros de software en relación con el aprendizaje y adopción de procesos de software [Mathiassen, 2005].

Dado que la calidad de un software está directamente relacionada con la calidad del proceso por el cual es desarrollado [Humphrey, 1989], muchas organizaciones de ingeniería del software tratan de mejorar la calidad de sus productos mediante la mejora de sus procesos [Liou, 2011]. Para ello, las organizaciones de ingeniería de software deben esforzarse continuamente en maximizar el uso de su conocimiento organizacional y/o de su experiencia [Ivarsson, 2012]. Para afrontar este reto, las empresas aplican habitualmente los principios del KM con el objetivo de lograr una adaptación a los entornos cambiantes del negocio mediante la introducción de un conjunto específico de procesos y prácticas para identificar y capturar conocimiento, haciendo que los activos de conocimiento estén disponibles para su transferencia y reutilización a lo largo de la organización [Kamsu-Foguem, 2008].

Para hablar de las características y beneficios de la Mejora de Procesos de Software, así como de la aplicación del KM en este campo, es necesario describir primero qué son los procesos de software.

### 2.2.1. Proceso de software

El proceso de software es 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) [CMMI, 2010].

Los procesos de software pueden estar definidos explícitamente o pueden ser procesos implícitos que se usan sin tener consciencia del proceso como tal. Según [Kneuper, 2002], los motivos para definir los procesos de software son:

- Los procesos definidos proporcionan una estructura al desarrollo de software.
- Los buenos procesos de software dan soporte a los desarrolladores en su labor indicando cuáles son los pasos a realizar y proporcionando plantillas para los resultados que se deben obtener, de forma que los desarrolladores pueden concentrarse en resolver problemas técnicos.
- Los procesos definidos conforman la base para la mejora y el aprendizaje.
- Los procesos definidos hacen que el desarrollo sea más predecible, ayudando así a obtener estimaciones de tiempo y coste más reales, y dando como resultado productos de calidad.

Un proceso de software definido identifica roles y responsabilidades, y actividades junto con sus criterios de entrada, salida y medidas de proceso [Cuevas, 2002]. De esta manera, el proceso de software ofrece una descripción clara de las actividades y tareas que los agentes involucrados en el proceso (humanos o no) deben realizar para elaborar un determinado software [Derniame, 2004].

Para la descripción de los procesos se suelen utilizar representaciones creadas con el fin de dar soporte al entendimiento de procesos complejos, a la comunicación acerca de ellos, a su evaluación, y a las demás tareas aplicables [Kellner, 1998]. Un proceso que no esté adecuadamente representado será difícil de realizar, de controlar, y de mejorar.

Cada representación de un proceso captura los aspectos de un proceso que son relevantes para una tarea en particular, es decir, una representación de un proceso que esté orientada a la mejora de procesos puede ser diferente de una representación del mismo proceso orientada a otro propósito. Así, en la práctica existen varios modos de representación de procesos, los cuales se diferencian más por su forma y su uso que por sus contenidos. Entre ellos se pueden destacar los modelos de procesos, las plantillas y formularios de procesos, y las guías de procesos tanto en su formato tradicional como en su formato electrónico.

### 2.2.2. Mejora del Proceso de Software

En los últimos tiempos, el software se ha convertido en una parte imprescindible para la sociedad. Sin embargo, la Ingeniería del Software es una disciplina que todavía parece no haber alcanzado su madurez [Aurum, 2008]. Ver el desarrollo de software como un proceso ha ayudado significativamente a identificar las diferentes dimensiones del desarrollo de software y de los problemas que deben tratarse para establecer prácticas efectivas. Los investigadores y los profesionales se centran en el estudio y en la mejora del proceso por el cual el software es desarrollado dada la correlación directa entre la calidad del proceso y la calidad del software desarrollado [Humphrey, 1989].

Desde principios de los años 90, la Mejora de Procesos de Software (*Software Process Improvement – SPI*) ha tratado de afrontar los desafíos que presenta la mejora de la calidad y de la eficiencia en las prácticas de Ingeniería del Software [Pino, 2008], facilitando la identificación y aplicación de cambios en las actividades de gestión y desarrollo de software [Allison, 2007].

El SPI es una aproximación a la mejora sistemática y continua de la capacidad que una organización tiene para producir y entregar software de calidad cumpliendo unas restricciones de tiempo y presupuesto [Arent, 2000]. El SPI está orientado a madurar el proceso de software, es decir, a que el proceso de software esté mejor definido e implementado de forma más consistente en la organización [Aaen, 2001].

Los proyectos de SPI habitualmente comienzan con una evaluación de las prácticas que se llevan a cabo actualmente en la organización, identificando cuellos de botella, problemas u oportunidades [Capell, 2004], para luego identificar y priorizar las mejoras potenciales en el proceso de desarrollo de software existente. El objetivo a largo plazo del SPI es implementar e institucionalizar en la organización las prácticas mejoradas de desarrollo de software, es decir,



crear nuevo conocimiento a nivel organizacional [Arent, 2000]. Este conocimiento será adquirido a nivel individual por los participantes de los procesos y posteriormente expandido al nivel organizacional para ser aplicado en nuevos proyectos.

El interés existente en esta disciplina dentro de la industria del software es evidente por el gran número de artículos relacionados con SPI, tal y como muestra el análisis realizado por Hall et al. en [Hall, 2002]. Además, existe un gran número de iniciativas internacionales relacionadas con SPI, como son CMMI [CMMI, 2010], ISO/IEC 15504 [ISO, 2006], e ISO/IEC 12207 [ISO, 2008], entre otras.

El SPI se está convirtiendo en un estándar internacional y hoy en día es una estrategia adoptada por cada vez más organizaciones en el mundo entero [SEI, 2006]. Las organizaciones que no adopten este esquema de trabajo corren el riesgo de quedar descolgadas en un mercado cada vez más globalizado y competitivo.

### ***Factores para el éxito en la Mejora del Proceso de Software***

En esta sección se presentan algunos elementos que suelen estar presentes en los esfuerzos de mejora que tienen éxito.

En primer lugar, es necesario establecer una imagen clara de cómo se llevan a cabo las actividades actualmente en la organización, y desarrollar posteriormente un plan para acometer los cambios [Capell, 2004]. Este plan debería detallar el programa de acciones a tomar y se deberían asignar responsables para cada una de ellas. Este plan de acción tiene las siguientes ventajas: (1) Facilita el entendimiento común de objetivos, hitos y resultados esperados para acciones concretas de la mejora; (2) La mejora puede ser descompuesta en una secuencia de tareas limitadas, cada una de las cuales tendrá un conjunto de objetivos operacionales; (3) Las acciones de mejora pueden ser priorizadas y coordinadas para clarificar cómo se interrelacionan las tareas involucradas; (4) El plan ayuda a formar y mantener el compromiso de la directiva, de los miembros del proyecto de SPI y de otros profesionales; (5) El plan puede ser utilizado como un vehículo para comunicar el progreso y asegurar la visibilidad de los esfuerzos en SPI [Aaen, 2001].

También se debe establecer un entorno piloto en el cual se puedan aplicar los cambios de forma segura [Capell, 2004]. Organizar el SPI como un proyecto, por ejemplo, ofrece una serie de beneficios. En primer lugar, asegura que la adaptación del proceso puede tener lugar con la debida consideración de los aspectos prácticos experimentados. En segundo lugar, la asignación de recursos será parte integral del proceso de planificación del proyecto. En tercer lugar, organizar el esfuerzo como un proyecto con entregables concretos derivados de los objetivos del SPI incrementará el control del proyecto y de sus resultados [Aaen, 2001]. Por el contrario, algunos riesgos de organizar el SPI como un proyecto pueden ser que el esfuerzo se hace más vulnerable a la falta de recursos, que las soluciones producidas por el proyecto no sean consideradas relevantes, o que haya parte de la organización que vea con indiferencia los resultados obtenidos [Aaen, 2001].

Finalmente, una de las lecciones aprendidas de la práctica del SPI es que la mejora debe ser integral y coordinada para ser efectiva en la resolución de problemas a nivel de la organización

[Aaen, 2001]. Tener éxito en la mejora requiere tiempo y líderes comprometidos que proporcionen un entorno apropiado para el cambio cultural que implica el SPI, así que se debe tener en cuenta todo el entorno de la organización [Capell, 2004].

### ***Beneficios de la Mejora del Proceso de Software***

En [Capell, 2004] se hace una revisión de artículos, documentos, libros y sitios web para determinar el tipo y el alcance de los beneficios que los métodos de SPI pueden aportar, descubriendo los siguientes aspectos comunes:

- Todos los esfuerzos de mejora se basan en la eliminación de errores en las etapas iniciales del proceso.
- Mientras que los esfuerzos de mejora son dirigidos típicamente por el coste y medidos por el retorno de la inversión, los atributos propios de la calidad como son la seguridad y la estabilidad tienen a menudo mayor valor para el éxito del proyecto.
- La mayoría de esfuerzos de mejora proporcionarán beneficios siempre y cuando dichos esfuerzos sigan unas reglas generales y usen los métodos apropiados.

En general, el SPI conlleva beneficios significativos como son:

- La reducción de costes [Niazi, 2006].
- Un retorno de la inversión positivo [Capell, 2004].
- La reducción de riesgos [Dybå, 2005a].
- La mejora de la calidad del producto software [Allison, 2007] [Pino, 2008].
- La mejora de la moral, de la responsabilidad y de la comunicación en el equipo de trabajo [Capell, 2004].
- El aumento de la satisfacción del cliente [Mathiassen, 2005] [Rico, 2004].
- En definitiva, una mayor tasa de éxito de los proyectos [Capell, 2004].

### ***Modelo IDEAL***

IDEAL es un modelo para un programa de mejora continua de procesos de software propuesto por el SEI [McFeeley, 1996] que forma una infraestructura para guiar a las organizaciones en la planificación e implementación de un programa efectivo de SPI. Para que los cambios del programa de SPI funcionen no deben hacerse al azar, sino que es necesario planificarlos y poder supervisar sus resultados. Este modelo proporciona una descripción genérica de una secuencia recomendada de pasos que sirve de guía para la implementación del cambio.

El modelo IDEAL contempla los siguientes pasos (Figura 2.3), los cuales dan nombre al modelo:

- Iniciar (*Initiating*): Se buscan las bases para garantizar el proyecto de mejora a través del patrocinio, la infraestructura, y la asignación de recursos y establecimiento de compromisos.
- Diagnosticar (*Diagnosing*): Evaluar y diagnosticar mediante un método formal el estado actual de la organización en cuanto al uso de los procesos identificando sus fortalezas y debilidades.

- Establecer (*Establishing*): En base a los resultados y a los objetivos que la organización se ha propuesto alcanzar se genera un plan de acción para la mejora. En base a las necesidades se establecen prioridades, objetivos y equipos de acción de procesos.
- Actuar (*Acting*): Se implementa el plan de acción, se desarrollan las mejoras, se definen o ajustan los procesos y artefactos, se capacita, se implementan las iniciativas de mejora en pilotos, y se miden los avances.
- Aprender / Difundir (*Learning*): Se analizan los resultados de la implementación y se aprende del ciclo recién realizado para mejorar los procesos en forma continua.

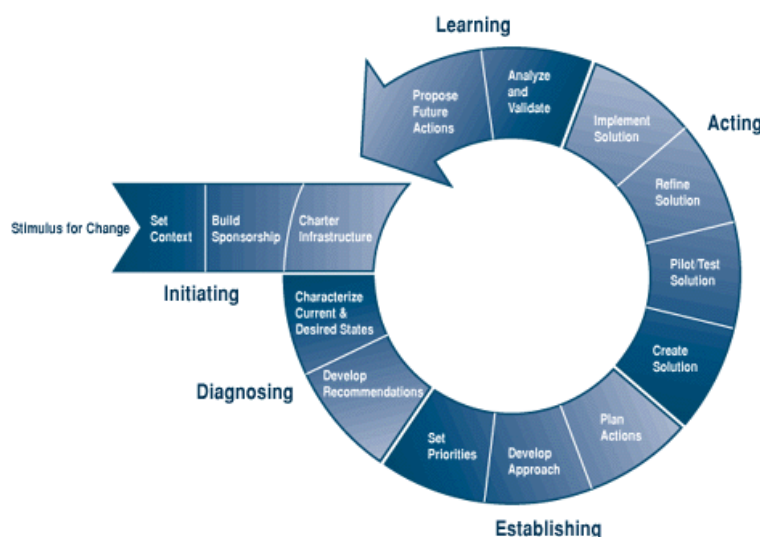


Figura 2.3. Modelo IDEAL.

A través de la ejecución y supervisión continua de los pasos anteriores se pueden implementar las mejoras hasta obtener los resultados buscados. Los tiempos del ciclo dependerán del tamaño y complejidad de los cambios.

Los principios del KM pueden ofrecer soporte para gestionar el conocimiento en una iniciativa de SPI que se base en este modelo. La Tabla 2.2 resume el tipo de soporte que es necesario en cada fase.

Tabla 2.2. Soporte del KM en las fases del modelo IDEAL.

Fase	Soporte de KM necesario
Iniciar	- Difusión de la visión, compromiso y objetivos del programa de SPI.
Diagnosticar	- Recogida de evidencias de la práctica actual de la organización y sus procesos. - Mapa o gráfico de la capacidad o de la madurez de la organización.
Establecer	- Planificación de las iniciativas específicas para la mejora de procesos. - Visualización y seguimiento del plan de acción.
Actuar	- Definición de los nuevos procesos y adquisición de nuevos activos. - Prueba piloto para la adquisición de conocimiento tácito del uso de los procesos. - Aprendizaje y adopción gracias a la publicación del conocimiento enriquecido y de la realimentación.
Aprender	- Medición y evaluación de la mejora mediante métricas del uso del conocimiento.

De acuerdo con la hipótesis y los objetivos establecidos para esta tesis doctoral, la solución propuesta deberá facilitar las actividades de KM relacionadas principalmente con las fases de

Actuar y de Aprender, lo cual a su vez también influirá de cierta manera en la fase de Diagnosticar ya que se facilitará la evaluación del estado actual de la organización en cuanto al uso de los procesos.

### 2.2.3. Gestión del Conocimiento en la Mejora de Procesos de Software

Para mejorar los procesos de desarrollo de software y obtener los beneficios citados anteriormente, las organizaciones deben aplicar los principios del KM para gestionar adecuadamente su conocimiento [Dingsøyr, 2009]. Así, el KM se convierte en una parte esencial de los esfuerzos en SPI ya que la identificación de nuevo conocimiento es considerada como una de las claves para el éxito en la mejora de procesos [Maier, 2007].

Sin embargo, el estudio llevado a cabo por [Komi-Sirviö, 2002] revela que el conocimiento existente no es sencillo de encontrar y que, cuando es descubierto, su reutilización es difícil de lograr en la práctica. Este problema se debe a que, dentro de las organizaciones, gran parte del conocimiento útil para la ejecución de los procesos de software es tácito [Assimakopoulos, 2006], y solamente una parte de todo el conocimiento relativo al proceso de software puede ser capturado y hacerse explícito [Rus, 2002]. Además, este conocimiento tácito solamente se hace presente por medio de la práctica y, para su captura, suele ser necesario que las personas formen parte de un equipo de trabajo [Goffin, 2011].

Por este motivo, es necesario construir repositorios que ayuden a recolectar y almacenar el conocimiento acerca de los procesos de software llevados a cabo por los equipos de trabajo, y que permitan posteriormente compartir y reusar este conocimiento en otros equipos de trabajo de la organización [Ivarsson, 2012]. La Biblioteca de Activos de Procesos es una de estas herramientas que permiten reunir en un mismo repositorio todo este conocimiento, además de todos los artefactos que pueden ser utilizados por los procesos, siendo así un elemento clave de la infraestructura tecnológica que permite utilizar las ideas del KM para dar soporte al SPI [Calvo-Manzano, 2008].

### 2.2.4. Bibliotecas de Activos de Procesos

Una Biblioteca de Activos de Proceso (*Process Asset Library* – PAL) es un repositorio organizado, correctamente indexado, con capacidad de búsqueda, y fácilmente accesible por cualquiera que necesite información orientativa sobre procesos, como pueden ser ejemplos, datos, plantillas, u otros materiales de soporte al proceso [García, 2004].

El término “activo de proceso” fue introducido por primera vez en el *Software Capability Maturity Model* (SW-CMM v1.1). Los activos de procesos son una colección de entidades consideradas útiles por la organización que son utilizadas en proyectos durante el desarrollo, adaptación, mantenimiento e implementación de los procesos de software [Calvo-Manzano, 2008]. Estos activos de procesos estarán recogidos en una PAL, y deben ser almacenados, gestionados, presentados y reutilizados de manera efectiva. Una PAL bien diseñada y mantenida facilita la estandarización y la mejora de procesos en la organización [Fogle, 2001].

Una PAL está formada por un repositorio de conocimiento y una interfaz basada en tecnologías Web 2.0 para acceder a dicho conocimiento, lo cual permite:

- Almacenar información (activos) acerca del proceso de software de una organización, permitiendo que la experiencia adquirida pueda ser utilizada en futuros proyectos de software.
- Facilitar la búsqueda y reutilización del conocimiento previo acerca de los procesos de la organización que mejor se adapte a cada proyecto de la organización.
- Registrar la experiencia acumulada, lecciones aprendidas y consejos sobre cómo aplicar los procesos en la organización, utilizar herramientas software e ejecutar los diferentes tipos de proyectos de la organización.
- Facilitar la gestión del cambio de conocimiento de la organización para su adaptación a las necesidades cambiantes del entorno.

Así, en lugar de seguir un modelo jerarquizado en el cual los procesos son definidos por sus autores y llevados a cabo por los equipos de proyecto, una PAL soportada por una tecnología Web 2.0 va a proporcionar un entorno colaborativo en el cual los equipos de trabajo están completamente involucrados en el desarrollo de procesos proporcionando realimentación y lecciones aprendidas acerca del uso de los procesos en sus proyectos.

Entre las principales características soportadas por una PAL están:

- Gestión del Conocimiento: soporte integral al repositorio con procesos de KM (adquisición, refinamiento, organización, almacenamiento, publicación, entrenamiento y aprendizaje) sobre los activos de proceso. La PAL soporta la colaboración y la generación compartida del conocimiento de activos de proceso.
- Mejora del Proceso de Software: además de ser un repositorio de elementos del proceso, la biblioteca evoluciona gracias al trabajo colaborativo de sus usuarios evaluando y mejorando sus elementos de conocimiento mediante anotaciones, comentarios y realimentación a partir de su uso.
- Gestión del repositorio: mecanismos de abstracción para extraer información de ejemplos concretos eliminando el contexto para una reutilización efectiva, incorporando una metodología propia orientada a la gestión del ciclo de vida de los activos de proceso (definición, modelado, relaciones, prácticas de uso, y criterios y procedimientos para añadir, eliminar y modificar activos en el repositorio).
- Ejemplos y estándares: incorporación de plantillas y material de soporte al proceso, ejemplos e instancias de dichos elementos, y aplicación de estándares para promover la interoperabilidad que ayuden a facilitar la diseminación eficiente de contenidos.
- Ayudas a los usuarios: implementación de agentes proactivos que actúan inmediatamente cuando es necesario para facilitar el trabajo a los usuarios del repositorio, personalización de la interfaz, y soporte acorde al nivel de experiencia del usuario.

A diferencia de otras tecnologías, la PAL favorece la gestión de activos de proceso a través de una infraestructura de KM para adquirir, definir y diseminar guías acerca de los procesos, además de facilitar la implementación y gestión de proyectos y de incrementar el rápido uso de activos utilizando recursos de la Web 2.0. También incorpora más recursos para buscar y reusar activos en proyectos.

### ***Propósitos de una Biblioteca de Activos de Proceso***

Los propósitos de una PAL incluyen, entre otros:

- Proporcionar una base central de conocimiento para adquirir, definir y difundir información orientativa acerca de los procesos dentro de la organización [Calvo-Manzano, 2008].
- Proporcionar mecanismos para compartir conocimiento acerca de los activos de proceso de la organización y de cómo se usan éstos [Cuevas, 2008].
- Reducir la duplicación innecesaria de activos de proceso dentro de la organización [García, 2004], mejorando así la efectividad y la eficiencia.
- Proporcionar un entorno de aprendizaje efectivo para los nuevos empleados que se espera usen los procesos de la organización [Calvo-Manzano, 2008], reduciendo así el tiempo de entrenamiento.
- Minimizar el riesgo de pérdida del conocimiento tácito debida a la rotación del personal [He, 2007] gracias a la aplicación de técnicas de KM individual y grupal sobre el proceso de software.
- Obtener innovación del proceso y del producto, motivando al personal para crear y utilizar el proceso, e incrementando la calidad de los productos desarrollados [García, 2004].

### ***Contenidos de una Biblioteca de Activos de Proceso***

Los contenidos de una PAL incluyen los siguientes elementos:

- Activos de definición del proceso: definiciones de procesos, modelos de procesos, métodos y técnicas de procesos, productos de trabajo, estándares, roles, políticas y criterios de selección y ajuste.
- Activos de soporte e implementación del proceso: plantillas de producto, ejemplos de productos de trabajo, recursos de entrenamiento e información sobre herramientas.
- Activos de soporte a la evaluación: descripciones de modelos de referencia de la industria, criterios de evaluación, métodos de evaluación y transformaciones entre modelos de referencia y el proceso de la organización.
- Datos históricos de proyectos: características y métricas de proyecto/producto, métricas de proyecto/producto, lecciones aprendidas y perfiles de calidad de proyecto/producto.
- Datos de mejora de la calidad del proyecto/producto: resultados de auditoría, planes de mejora del proceso, planes de acción, informes de estado y activos pilotos.
- Datos de realimentación del proceso: reportes de problemas, solicitudes de mejora y realimentación de los usuarios de los procesos.

Por medio de estos elementos, el proceso es establecido, documentado y medido [Kulpa, 2008]. El conocimiento almacenado en la PAL representa la experiencia de la definición del proceso en los proyectos realizados, indica el grado de despliegue del proceso en la organización, y ofrece mecanismos para plantear posibles mejoras del proceso ya que representa la memoria corporativa del proceso en la organización. En otras palabras, se puede entender que una PAL está compuesta por:

- Una parte de proceso (por ejemplo, una Guía Electrónica de Procesos), la cual contendrá todo el conocimiento relacionado con los procesos de la organización.
- Una parte de proyecto (por ejemplo, una wiki), la cual contendrá todos los activos que se hayan obtenido a partir del desarrollo de proyectos en la organización.

## A) Parte de proceso de una Biblioteca de Activos de Proceso

En la práctica, el modo más habitual de representar procesos en una PAL es la guía de procesos.

### *Guías de procesos*

Las guías de procesos y el KM se utilizan habitualmente en las organizaciones para mejorar la calidad del software que producen y para aumentar la productividad de su trabajo [Kurniawati, 2004a]. Estas guías de procesos son todos aquellos contenidos que definen el conjunto de procesos implantados dentro de la organización, y que por tanto deben ser seguidos por los equipos de trabajo de la misma.

Más formalmente, diversos autores [Dingsøyr, 2004] [Kellner, 1998] [Wangenheim, 2006] definen este concepto como documentos de referencia, estructurados y orientados al flujo de trabajo para un proceso particular que proporcionan soporte a las personas encargadas de llevar a cabo dicho proceso de forma efectiva y eficiente. Por tanto, una guía de procesos debe proporcionar una descripción explícita y concreta del proceso al que se refiere, sin dar lugar a ambigüedades, y debe ser fácil de entender, de comunicar y de seguir. Además, debe hacer referencia a su contexto operativo, es decir, a las condiciones bajo las que el proceso se efectuará y que incluyen al entorno y a las infraestructuras organizacionales que utilizarán el proceso, sus aptitudes y su nivel actual de entendimiento sobre el proceso.

Las guías de procesos proponen una serie de pautas sobre quién, cómo y cuándo realizar ciertas actividades para alcanzar el objetivo del proceso. Los elementos comunes que suelen incluir estas guías de procesos son texto narrativo acerca del proceso, tablas o árboles de decisión, asistencia gráfica, plantillas, listas de comprobación, etc. La guía de procesos también puede incluir elementos tales como descripciones detalladas de métodos concretos, advertencias sobre riesgos o dificultades, ejemplos de productos de trabajo, etc. [Kellner, 1998].

Una guía de procesos es un elemento importante para el SPI, ya que supone una primera aproximación para desplegar un KMS en una organización. Una vez que las iniciativas de mejora han sido acordadas y se han definido nuevos procesos –o se han modificado los ya existentes dentro de la organización–, es necesario comunicar y transferir esta información a través de la organización. Muchas organizaciones software están utilizando guías de procesos para realizar dicha transferencia de conocimiento entre sus empleados. Además, realizar de forma eficaz este proceso es crucial en cualquier iniciativa de mejora de software. Por tanto, el equipo de trabajo necesita una guía que sea efectiva para poder transferir el conocimiento de forma eficaz [Moe, 2006].

Diversos autores [Kellner, 1998] [Kurniawati, 2006] [Wangenheim, 2006] destacan los siguientes como los principales beneficios del uso de las guías de procesos:

- Proporcionan una excelente oportunidad de establecer una aproximación claramente definida, sistemática y repetible al desarrollo de software.
- Proporcionan madurez a los procesos de software, siendo estos aplicados de forma más consistente a lo largo de los proyectos.
- Proporcionan un flujo de trabajo claro y guiado que conlleva la liberación de los más expertos de la labor de guiar a los más inexpertos, consiguiendo así una mayor eficiencia en el trabajo.
- Permiten lograr una mejor gestión de riesgos al proporcionar procedimientos estandarizados y una documentación más consistente y completa.
- Permiten una planificación más sencilla y precisa, lo que se traduce en un aumento del porcentaje de proyectos que cumplen los plazos, así como en unos presupuestos más realistas.
- Los usuarios obtienen una mejor comprensión de los procesos y de sus elementos fundamentales, como son las actividades, productos de trabajo, roles, etc.
- Facilitan la comunicación tanto online como offline. La comunicación online podría darse entre las personas que forman parte de los equipos de trabajo participantes en el proceso y que llevan a cabo acciones relacionadas, mientras que la comunicación offline podría darse entre autores y lectores de documentos de procesos.
- Ayudan a los equipos de trabajo que participan en el proceso a trazar su trabajo en base a información sobre eventos del proceso (por ejemplo, tan sencillo como tachar tareas una vez que han sido completadas).
- Ayudan a los equipos de trabajo que participan en el proceso a realizar su trabajo eficientemente (por ejemplo, en base a guías que hagan referencia a experiencias que ayuden a elegir entre diferentes alternativas en una toma de decisiones).
- Ayudan a los equipos de trabajo que participan en el proceso a volver a encarar un proceso, completo o en parte (por ejemplo, la información de estado podría ayudarles a entender lo que necesitan para reiniciar un trabajo después de una interrupción).
- Entre los resultados observados tras la implantación de guías de procesos en PYMEs están las mejoras en la ejecución eficiente y efectiva de procesos, en la planificación de proyectos, en la estimación, monitorización y control, así como una mejora en la calidad de los productos y servicios, y mejora en la relación con el cliente y en la satisfacción del cliente.

Tradicionalmente, las guías de procesos se desarrollaban en formato papel, presentándose en forma de manuales de procesos y procedimientos, procedimientos operativos estandarizados y descripciones de ciclos de vida, entre otros. Con la llegada de las nuevas tecnologías, algunas organizaciones actualizaron sus guías de procesos tradicionales a formatos electrónicos muy básicos, como podían ser los documentos de texto, para poco después comenzar a usar tecnologías como la Web.



### ***Guías de procesos tradicionales***

Las guías de procesos tradicionales se caracterizan por que el modo de describir y comunicar los procesos a través de toda la organización se centra en guías impresas, estándares, e incluso libros [Moe, 2005]. Sin embargo, estas guías presentan un problema en cuanto a su usabilidad ya que, entre otras causas, son difíciles de mantener actualizadas y difíciles de personalizar en base a las necesidades específicas de su usuario [Moe, 2006]. Esto fomenta que la transferencia de conocimiento utilizando guías de procesos tradicionales sea una tarea ardua y compleja y que, por tanto, los equipos de trabajo no se vean motivados a utilizar este tipo de guías.

En [Kellner, 1998] se abordan los principales problemas de las guías tradicionales. Éstos pueden resumirse en los siguientes:

- Los lectores no pueden navegar fácilmente a través de las páginas si no siguen el flujo de información del documento.
- Las búsquedas de información en la guía es una tarea muy complicada.
- Falta de información clave, las actualizaciones son demasiado complejas.
- Las anotaciones y aclaraciones incluidas en las guías por un lector no pueden ser compartidas con otros.
- El control de versiones no está soportado de la mejor manera y, además, la distribución de una nueva versión de un proceso a sus participantes es poco formal.
- Las guías, o bien contienen información mezclada para diferentes lectores, o bien están compuestas de documentos orientados a necesidades específicas que sin embargo comparten una descripción de proceso común.
- Las guías no se encuentran diseñadas para almacenar información de estado del proyecto.
- No ofrecen facilidades de adaptación a las necesidades de los usuarios.
- Resulta difícil estructurarlas con el fin de presentar información diferente pero que se encuentra relacionada.

Tal y como indican estos problemas, las guías de procesos tradicionales presentan ciertas limitaciones por lo que no permiten alcanzar un completo aprovechamiento de todos los beneficios que las guías de procesos son capaces de ofrecer. Encontrar soluciones para estos problemas es posible gracias al desarrollo de nuevas tecnologías, principalmente la Web, surgiendo así las Guías Electrónicas de Procesos.

### ***Guías Electrónicas de Procesos***

Con la llegada de las nuevas tecnologías, algunas organizaciones actualizaron sus guías de procesos tradicionales a formatos electrónicos muy básicos, como podían ser los documentos de texto. Sin embargo, poco después surgió un nuevo concepto entorno a las mismas: la Guía Electrónica de Procesos (*Electronic Process Guide* – EPG). Según [Dingsøyr, 2008], una EPG puede ser vista como un documento de referencia para un proceso particular, disponible en línea, estructurado y orientado al flujo de trabajo, y que existe para dar soporte a sus usuarios para llevar a cabo dicho proceso.

Estas EPGs permiten implementar mejoras sobre las guías de procesos tradicionales haciendo uso de las nuevas tecnologías, como por ejemplo la Web, lo cual permite una transferencia del conocimiento más eficiente que con las guías tradicionales [Kellner, 1998]. Aparte de proporcionar información a los usuarios de los procesos, las EPGs pueden ser utilizadas como herramientas de soporte para los esfuerzos en SPI [Moe, 2006].

De acuerdo con [Dingsøyr, 2004], una EPG debería proporcionar todos los elementos informativos y relaciones contenidas en una buena guía de procesos tradicional. Además, debería sacar provecho de elementos tales como diagramas o tablas para proporcionar una interfaz de usuario efectiva. A partir de esto es fácil deducir que una EPG es algo más que los modelos de los procesos y que una guía de procesos tradicional. Una EPG añade cierta funcionalidad a la guía de procesos ya que va a estar soportada por una tecnología. Éste soporte tecnológico constituye la base para solucionar los problemas que presentan las guías tradicionales facilitando así su utilización.

Aunque no necesariamente tenga que utilizarse tecnología Web, la idea de EPG está asociada con la de una aplicación web estructurada contenedora de descripciones de procesos, que permite la navegación y la búsqueda rápida, y que ofrece información y funcionalidad adicionales. Esto incluye un uso extenso de hipervínculos, plantillas, ejemplos, herramientas, y bases de datos de proyectos [Scott, 2002b]. Los hipervínculos proporcionarán a los usuarios de la EPG una navegación flexible e intuitiva además de accesos directos a información adicional como productos de trabajo, plantillas, procedimientos, ejemplos o herramientas relacionadas.

El hecho de que habitualmente hagan uso de la tecnología Web y de que puedan ser instaladas en las redes internas de las organizaciones permite que:

- Los miembros de los equipos de trabajo que participan en el proceso puedan acceder, en cualquier momento, a la información actualizada de una manera muy rápida y sencilla.
- La navegación por la EPG y las búsquedas de información en la misma se puedan realizar de forma fácil.
- Se simplifiquen las actualizaciones, ya que sólo hay que actualizar en un único lugar y no en todos los manuscritos existentes. Por tanto, siempre se encontrará información actualizada a disposición de los usuarios. Además, el control de versiones no presentará problemas.
- Se estructure la información de forma adecuada teniendo en cuenta las necesidades de cada usuario.
- Se recojan anotaciones, aclaraciones o comentarios de los usuarios con el objetivo de poder compartirlas con otros usuarios.

Sin embargo, no todo son ventajas puesto que también existen algunos inconvenientes. Entre ellos, destacan la resistencia al cambio y la dificultad de codificación cuando los procesos son largos y complejos [Kurniawati, 2006]. Existen organizaciones donde hay equipos de trabajo que están acostumbrados a hacer uso de guías de procesos tradicionales, que se encuentran familiarizados con este tipo de guías y que no quieren cambiar. Esta resistencia al cambio puede suponer un problema grave si no se afronta de forma adecuada. Además, en algunas ocasiones las organizaciones tienen definidos procesos largos y complejos, cuya codificación

en una EPG puede llegar a resultar una tarea difícil, y su mantenimiento puede complicarse en gran medida si los procesos cambian frecuentemente.

Otro problema habitual que se puede encontrar respecto a las EPGs es que, una vez implantadas en una organización, estas no son utilizadas por los desarrolladores o los jefes de proyecto. Para evitar este tipo de situaciones, en [Dingsøy, 2008] se sugiere que involucrar a los empleados en la creación de la EPG tiene un efecto positivo a largo plazo respecto a la posterior utilización de dicha EPG.

A pesar de las desventajas encontradas, la mayoría de autores referenciados mantienen que las EPGs pueden llegar a ser una forma eficaz de transferir el conocimiento de los procesos y permiten desarrollar software de mejor calidad. Para ello, es de gran importancia que la EPG se desarrolle teniendo en cuenta un conjunto de objetivos y requisitos básicos. A continuación se indican algunos de ellos.

### ***Requisitos de una Guía Electrónica de Procesos***

Analizando la literatura existente a acerca de las EPGs, especialmente los trabajos realizados por Kellner [Armitage, 1994] [Kellner, 1998] y Dingsøy [Dingsøy, 2004] [Moe, 2005], se pueden obtener el conjunto de objetivos que debe cumplir una EPG. Éstos son los siguientes:

- Proporcionar una descripción del proceso, preferiblemente mediante gráficos, describiendo tanto actividades como productos, roles, agentes y recursos, así como las relaciones principales entre ellos.
- Hacer el mayor uso posible de diagramas, tablas y texto para proporcionar una interfaz de usuario efectiva.
- Hacer un uso extenso y efectivo de hipervínculos para proporcionar una navegación flexible a través de la información contenida en la guía de procesos.
- Proporcionar un acceso sencillo y rápido a la información deseada, en especial a la información que se requiera más habitualmente.
- Facilitar la orientación del usuario en el proceso usando estructuras que le sean familiares en cada página de la guía. Un diseño estructurado y bien organizado de la EPG facilitará la orientación y el uso general de la misma.
- Evitar el uso de un número elevado de ventanas que se solapen y que pueda provocar la desorientación del usuario. Los usuarios de la EPG deben tener control sobre la apertura, cierre, cambio de tamaño, posicionamiento y visualización de las ventanas.
- Ser implementada usando tecnología comercial genérica para aplicaciones Web. Esto proporcionará una funcionalidad conocida y unas interfaces familiares, simplificando a su vez el mantenimiento y las perspectivas de actualización de la EPG.

Como se puede ver, estos requisitos se centran principalmente en la interfaz de usuario. Una vez identificados estos requisitos básicos, el siguiente paso es centrarse en la funcionalidad adicional de la guía de procesos. Una cuestión importante que surge aquí es cuánta funcionalidad debe proporcionar la EPG. No existe una respuesta precisa para esta cuestión ya que dependerá de las necesidades de cada organización.

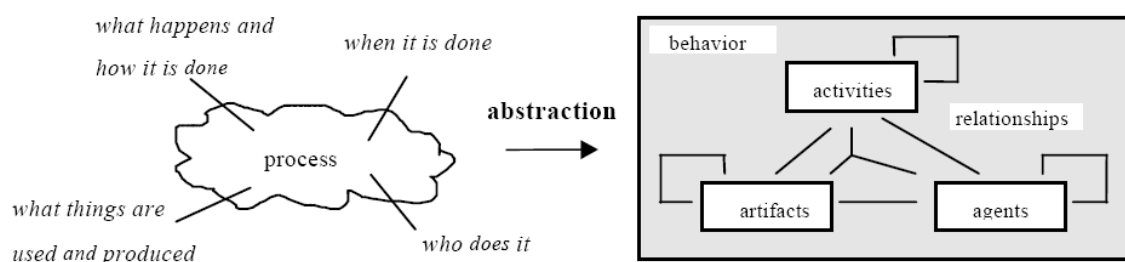
Kellner et al. [Kellner, 1998] proponen una serie de requisitos orientados al contenido y a la funcionalidad de la EPG que permiten mejorar la usabilidad y asegurar la fiabilidad del sistema. Estas propuestas adicionales son:

- Las páginas de la EPG deben ser fáciles de generar, extender, modificar y actualizar. Los procesos evolucionan con el tiempo y se debe asegurar la posibilidad de realizar estas modificaciones.
- Los usuarios de la EPG deben poder realizar búsquedas para extraer de la EPG la información que les interesa. Un ejemplo puede ser cuando un usuario de la EPG quiere obtener una lista de las actividades en las que participa un determinado rol.
- Se debe proporcionar un control de versiones de la EPG. Los cambios frecuentes en la EPG pueden llevar a que no se esté utilizando una versión actualizada de la guía de procesos en todos los proyectos, por lo que se hace necesario un control de versiones.

Por último, remarcar que todos estos requisitos se corresponden con el prototipo de EPG desarrollado por Kellner. Sin embargo, la mayor parte de estos requisitos pueden ser generalizados a la mayor parte de las EPGs.

### ***Elementos presentes en una Guía Electrónica de Procesos***

El trabajo de esta tesis está basado en el marco de trabajo conceptual para procesos de software propuesto en [Armitage, 1994] y representado en la Figura 2.4. A la hora de hablar de procesos de software, los aspectos que se suelen tener en cuenta son (1) *qué ocurre y cómo*, (2) *qué elementos son usados y producidos*, (3) *quién lo lleva a cabo*, y (4) *cuándo se lleva a cabo*. Haciendo una abstracción de estos aspectos, el proceso es conceptualizado como un conjunto de actividades (*qué ocurre*), artefactos (*elementos usados y producidos*) y agentes (*quién lo lleva a cabo*), las relaciones entre dichas entidades, y el comportamiento (*cuándo se lleva a cabo*) del conjunto completo formado por estas entidades y sus relaciones. La naturaleza dinámica de los procesos convierte al comportamiento en un aspecto crítico, quedando la información acerca de este comportamiento representada en forma de criterios de entrada o de salida, estados y transiciones, etc.



**Figura 2.4. Marco conceptual para procesos de software.**

Para la definición y modelado de procesos, Armitage y Kellner [Armitage, 1994] definen un esquema conceptual de los elementos más importantes que deben estar presentes en una EPG, y de las relaciones existentes entre los mismos. Posteriormente, Kellner en [Kellner, 1998] proporciona un esquema expandido, mostrado en la Figura 2.5, que proporcionará la base para almacenar la información que debe ser presentada en las guías de procesos. Los elementos básicos de este esquema son:

- Actividades (*Activities*): Descripciones de “cómo se hacen las cosas”, incluyendo una visión general y detalles de cada una de ellas.
- Productos de Trabajo (*Artifacts*): Descripciones de productos creados o modificados durante el proceso, tanto en su resultado intermedio o final como en su estado temporal interno.
- Agentes (*Agents*): Descripciones de entidades que pueden llevar a cabo actividades. Las descripciones son en términos de técnicas, costes y disponibilidad. Un agente puede ser un individuo o un grupo.
- Roles (*Roles*): Descripción de un conjunto de obligaciones y permisos relacionados con las actividades.
- Recursos (*Resources*): Descripciones de programas informáticos u otras ayudas que pueden ser utilizadas para dar soporte o automatizar el desarrollo de una actividad.

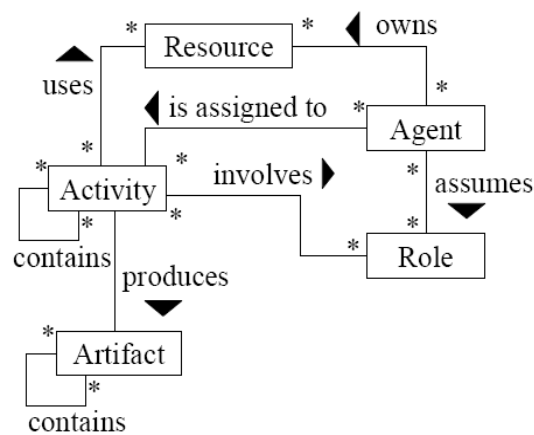


Figura 2.5. Esquema conceptual extendido.

Según el este esquema conceptual extendido, las actividades producirán artefactos utilizando los recursos que tienen los agentes. Estos agentes serán asignados a las actividades, asumiendo distintos roles para llevar a cabo las actividades en las que estén involucrados. Además, las actividades podrán dividirse en sub-actividades más pequeñas, de la misma forma que los artefactos podrán contener otros artefactos.

Además, en [Kellner, 1998] se define un conjunto de contenidos estructurados que deben incluirse en una EPG, estos son los siguientes:

- Una visión general de la actividad (propósito, diagrama de contexto y de descomposición, entradas, salidas, roles y agentes implicados, guías de esfuerzo, recomendaciones, políticas y estándares aplicables, conceptos y referencias, y advertencias).
- Detalles referentes a la propia actividad (objetivos, descomposición, diagrama de comportamiento, entradas/salidas, diagrama funcional, productos, y relaciones con los diferentes roles).
- Detalles referentes a los productos pertinentes de la actividad (descripciones y estados aplicables de entradas, salidas y productos internos, descomposición de productos, relaciones con otros productos, y localización de los productos).

- Detalles referentes a los roles y agentes implicados en la actividad (descripciones de participantes, roles y relaciones, responsabilidades).
- Detalles referentes a las tareas (objetivos, contexto, descripción de tareas, alarmas, diagrama de comportamiento y notas, criterios de entrada/salida, diagramas funcionales, productos de entrada y salidas, y roles y responsabilidades).

Dingsøyr [Dingsøyr, 2004] define una estructura genérica para las EPGs que es compatible con el esquema conceptual definido por Kellner [Armitage, 1994] [Kellner, 1998]. Así, la estructura de una EPG estará compuesta de los siguientes elementos:

- Entradas: Descripción de los elementos que deben estar disponibles para el desarrollo del proceso.
- Actividades: Descripción de "cómo se tienen que hacer las cosas", incluyendo una breve descripción de las actividades.
- Roles: Definición de los roles que se encuentran implicados en el desarrollo de las actividades.
- Documentos relacionados: Herramientas, plantillas y técnicas utilizadas para dar soporte o automatizar el desarrollo de una actividad.
- Salidas: Descripción de los elementos que se producen como salida.

## **B) Parte de proyecto de una Biblioteca de Activos de Proceso**

Existen diversas experiencias previas acerca de la utilización de wikis dentro del contexto de la Ingeniería del Software [Louridas, 2006], de forma que permiten gestionar el conocimiento interno de una organización.

La mayor parte de las aplicaciones prácticas de las wikis en la ingeniería del software están relacionadas con la gestión colaborativa de los elementos de trabajo, artefactos y activos generados a lo largo del desarrollo de un proyecto software.

Entre dichas referencias, Rech et al. [Rech, 2007] utilizan un marco de trabajo centrado en una wiki que permite a los equipos de desarrollo compartir experiencias de forma rápida y reutilizar conocimiento relacionado con los proyectos en PYMEs.

Aguiar y David presentan en [Aguiar, 2005] una wiki que sirve como repositorio para la documentación en un proyecto de desarrollo de software. La wiki ayuda a la obtención de un único documento a partir de un conjunto heterogéneo de contenidos (texto, código, modelos, imágenes) con el objetivo de reducir la documentación relacionada con el desarrollo, haciendo así que la documentación sea más apropiada y atractiva para los desarrolladores.

En [Fokaefs, 2009] se presenta una wiki como plataforma integradora para el desarrollo colaborativo de software en equipos de trabajo. La wiki permite mejorar la comunicación entre los miembros del equipo, mejorar la administración, evitar inconsistencias en la documentación compartida, y resolver los problemas que surgen a la hora de compartir artefactos.

Por otro lado, Solís et al. [Solís, 2009] describen la utilización de una wiki como herramienta para la gestión del conocimiento arquitectural, demostrando que dicha wiki puede ser usada para implementar espacios de trabajo ligeros para la compartición de conocimiento. En

[Schuster, 2007] utilizan una wiki como soporte para la toma de decisiones que deben realizar los arquitectos de software.

También hay referencias [Riechert, 2009] de la utilización de una wiki para que los *stakeholders* de un proyecto puedan elicitar, estructurar semánticamente y clasificar requisitos. Por último, en [Xiao, 2007] se presenta un entorno basado en una wiki en el cual los programadores pueden escribir código fuente, generar proyectos, ejecutar programas y depurar defectos de forma colaborativa. Existen muchos otros ejemplos de la utilización de wikis en el ámbito de la Ingeniería del Software [Phuwanartnurak, 2009] [Ras, 2009].

La utilización de wikis como herramientas para la gestión del conocimiento para la planificación y la documentación de proyectos software presenta las siguientes ventajas:

- Repositorio centralizado y compartido de conocimiento sobre el proceso de software de la organización, accesible desde cualquier lugar del mundo.
- Contribución asíncrona de los equipos de proyectos donde el contenido es generado, publicado y compartido de manera sencilla por los usuarios.
- Participación más activa, colectiva y colaborativa de los usuarios en todas las fases del ciclo de vida de los activos de procesos de software, aprovechando el potencial de los individuos para crear trabajos colaborativos.
- Medios para añadir anotaciones o discutir los contenidos publicados.
- Mayor eficiencia en la comunicación y mayor productividad, sobre todo cuando se compara con el intercambio de correos electrónicos entre los miembros del equipo de proyecto.
- Filtrado colaborativo donde, a partir de las preferencias del usuario o de usuarios con perfiles similares, se proporcionan recomendaciones personalizadas.
- Soporte para que el contenido evolucione, crezca y mejore con el tiempo.

Aunque la gestión del conocimiento en Ingeniería del Software se puede aplicar a nivel de proyectos, esta tesis doctoral se centra más en su aplicación a nivel organizacional puesto que es donde surgen las iniciativas para esta mejora de procesos.

## 2.3. Soluciones existentes para la gestión del conocimiento acerca de procesos de software

Diferentes autores han definido con anterioridad sus propias soluciones tecnológicas que aplican la gestión del conocimiento para la gestión y mejora de procesos de software, y en este apartado se resumen algunas de ellas. En base a las características de cada solución, en el apartado siguiente se hará un análisis comparativo de las mismas.

### 2.3.1. Spearmint

La primera definición de EPG y el primer prototipo fueron desarrollados conjuntamente por el Software Engineering Institute (SEI) y el Instituto Fraunhofer para la Ingeniería del Software Experimental (FhG IESE) [Kellner, 1998]. Dicho prototipo era generado codificando directamente las páginas HTML por medio de un editor WYSIWYG, por lo que realizar cambios en el aspecto o en los contenidos de la EPG requería de un considerable esfuerzo.

Poco después, Becker-Kornstaedt y Verlage en [Becker-Kornstaedt, 1999b] detallan las experiencias tenidas en la implementación como EPG del estándar alemán “Vorgehensmodell” (abreviado “V-Modell”) y en su posterior utilización. Esta EPG se basa en las mismas ideas que la EPG de Kellner, pero el FhG IESE decidió además implementar un generador de EPGs basado en Java con el objetivo de reducir el esfuerzo de crear la EPG. Esta herramienta se integra en el entorno Spearmint (Software Process Elicitation, Analysis, Review, and Measurement in an INTegrated environment), y permite exportar modelos de procesos del entorno Spearmint al formato de EPG en HTML.

Spearmint es un entorno integrado de modelado que utiliza un lenguaje visual de modelado específico de procesos para describir y definir procesos de software del mundo real [Becker-Kornstaedt, 1999a]. Este entorno proporciona soporte a los ingenieros de procesos durante la elicitación, la edición, la revisión y el análisis de modelos de procesos [Scott, 2001]. Además de soportar la generación de una EPG con la descripción de estos procesos, este entorno de modelado permite también comprobar la consistencia y la calidad de un modelo de proceso antes de generar la versión EPG de dicho proceso.

Spearmint proporciona 4 formas de representación (vistas), cada una de las cuales está diseñada para modelar una perspectiva distinta del proceso:

- Vista de flujo de producto: es una representación gráfica del modelo del proceso, donde unos iconos representan entidades que están conectadas mediante líneas que representan relaciones.
- Vista de propiedades: permite añadir detalles acerca de los distintos elementos del modelo del proceso mediante atributos y valores.
- Vista de descomposición: permite definir y navegar por una descomposición jerárquica de los elementos del modelo del proceso (artefactos, actividades y roles).
- Vista textual: permite ver online la información de los procesos modelados sin necesidad de generar la EPG.

En [Scott, 2001] se realiza una evaluación de Spearmint basada en dos casos de estudio. El primero de ellos probó la capacidad de Spearmint para la creación, edición, documentación y comunicación de procesos de desarrollo de aplicaciones web. El segundo de ellos probó la capacidad de expresividad del lenguaje de modelado de Spearmint a la hora de capturar toda la información del estándar ISO 12207, su capacidad para comunicar a los clientes información sobre un proceso, el funcionamiento del generador de EPGs, y la utilidad de la EPG generada. Los resultados obtenidos en dicha evaluación indican que el lenguaje de Spearmint soporta adecuadamente dichos proyectos, que la representación visual es de un valor inestimable para la comunicación con los clientes, que Spearmint puede gestionar modelos complejos, que el generador de EPGs permite ahorrar esfuerzos a la hora de producir una guía de procesos, y que la EPG generada fue bien aceptada por los participantes del proceso.

Entre las principales características de Spearmint descritas en [Scott, 2001], este análisis se centrará en el generador de EPGs. Este generador crea EPGs automáticamente a partir de modelos Spearmint. Estas guías son generadas de acuerdo a un formato predefinido con datos específicos del proyecto incluidos en el modelo. El propósito de este generador es



proporcionar guías de procesos de una forma rápida y sin errores, en oposición a la creación de dichas guías codificando las páginas web a mano.

Las EPGs creadas usando el generador de Spearmint se componen básicamente de:

- Un árbol de navegación disponible en cada página de la EPG.
- Una página principal que muestra un listado de todas las actividades, artefactos, roles y herramientas de la EPG.
- Una página individual para cada actividad, artefacto, rol y herramienta incluyendo una descripción, enlaces a otros elementos asociados, y enlaces a plantillas y ejemplos.

Uno de los principales inconvenientes de Spearmint es que no tiene una capacidad limitada de representación. Por ejemplo, para los roles solo permite describir los artefactos y actividades de los cuales es responsable. Además, esta herramienta no permite añadir elementos complementarios como pueden ser documentos técnicos, plantillas, ejemplos, listas de comprobación, etc. [Phongpaibul, 2007]. En este mismo artículo se indica que, al ser una de las primeras herramientas de generación de EPG, Spearmint presenta algunas limitaciones en cuanto a usabilidad, facilidad para realizar modificaciones y escalabilidad.

### 2.3.2. WAGNER

Kurniawati y Jeffery [Kurniawati, 2006] también se basan en los requisitos de Kellner para implementar una EPG de un modelo de proceso basado en el estándar ISO 12207. En este caso, la EPG se genera automáticamente a partir de unos diagramas de actividad en UML usando WAGNER (Web-based process Guide aNd Experience Repository) [Scott, 2002a].

La estructura y funcionamiento de este generador de EPGs están reflejados en la Figura 2.6. Usando una herramienta de modelado UML, los procesos son especificados gráficamente mediante diagramas de actividad UML. Estos diagramas son exportados a formato de intercambio XMI, y luego son transformados por un traductor XSLT usando plantillas XSL y JavaScript, lo cual permite definir el aspecto y la funcionalidad de la EPG generada.

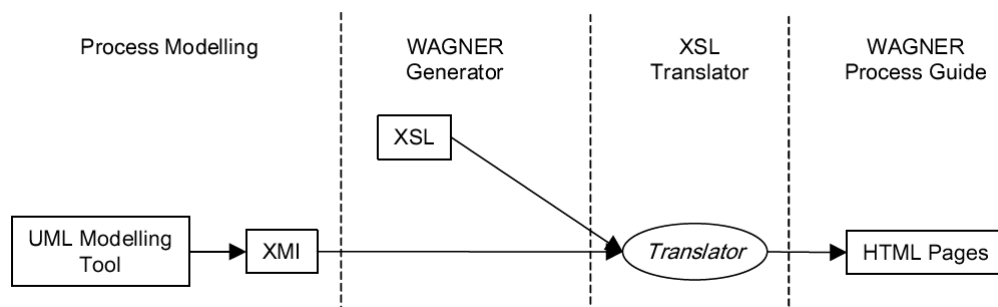


Figura 2.6. Framework de WAGNER.

El resultado es un conjunto de páginas HTML que presentan tres secciones:

- Una sección de visión general gráfica con el diagrama UML del proceso que proporciona navegabilidad mediante hipervínculos.
- Otra sección que muestra la descripción y las experiencias relacionadas con la entidad mostrada.

- Una tercera sección que presenta un árbol de navegación. Esta última sección fue eliminada de la EPG ya que su presencia era redundante al poderse navegar mediante los diagramas, de forma que además así se dejaba más espacio en pantalla para las otras dos secciones.

Como mejora, posteriormente se añadió a la EPG una página de inicio que proporciona el acceso a los contenidos de la EPG, un conjunto de accesos rápidos y un buscador.

En un artículo anterior [Kurniawati, 2004a], Kurniawati y Jeffery ya hicieron una comparación entre dos EPGs creadas usando diferentes herramientas de generación de EPGs. Primero se generó una EPG con Spearmint y más adelante una versión mejorada con WAGNER. La versión creada con Spearmint solo proporcionaba navegabilidad a través del modelo del proceso por medio de un árbol jerárquico, pero no proporcionaba una visión general gráfica del proceso. La versión creada con WAGNER proporcionaba además navegabilidad a través del modelo del proceso por medio de estos gráficos usando hipervínculos.

Como extensión a la EPG, en [Kurniawati, 2006] también se implementa un repositorio de experiencias (ER) usando PageSeeder, que es una herramienta desarrollada por la misma organización en la que se introdujo la EPG descrita en dicho artículo. PageSeeder introduce “semillas” en determinados lugares de las páginas de forma que los usuarios pueden añadir comentarios y realizar o responder preguntas. Usado como ER, los usuarios pueden adjuntar a las páginas de la EPG las plantillas, ejemplos, listas de comprobación, y experiencias genéricas (anécdotas, lecciones aprendidas, fragmentos de código y enlaces a más información) que consideren puedan ser de utilidad a otros usuarios de la EPG. La utilización de PageSeeder como complemento a la EPG permite que exista una realimentación por parte de los usuarios, de forma que la evolución del ER sería dinámica mientras que el resto de la EPG permanecería estática [Scott, 2002a].

Como alternativa a la aplicación propietaria PageSeeder, en [Kurniawati, 2004b] se propone el uso de una wiki de código libre (TikiWiki) para hacer las funciones de ER. Los autores indican los diversos problemas que encontraron a la hora de mostrar la EPG con un aspecto similar al obtenido usando PageSeeder. En concreto, problemas para mostrar el diagrama del proceso que está formado por varias imágenes, problemas para mostrar la EPG a un lado de la pantalla y el ER al otro, y problemas para modificar las páginas de la EPG debido a la ausencia de controles que faciliten la labor y a la necesidad de tener que escribir los contenidos de dichas páginas en HTML en vez de poder usar texto plano. Por estos motivos, los autores consideraron que la opción de usar una wiki como ER no era viable.

### 2.3.3. Microsoft Solutions Framework (MSF)

Microsoft Solutions Framework Process Guidance [MSF, 2012] es una EPG que recopila prácticas aprobadas por Microsoft, sus clientes, y la experiencia de la industria. Esta guía de procesos se integra dentro del Team Foundation Server (TFS), que es el núcleo de Visual Studio Team System (VSTS), una plataforma de Microsoft para la gestión del ciclo de vida de aplicaciones (*Application Lifecycle Management* – ALM). Además de la guía de procesos, el TFS también ofrece de manera integrada un portal para el equipo de proyecto, control de

versiones, seguimiento de elementos de trabajo, gestión de *builds*, y funcionalidades de negocio.

Esta EPG se presenta originalmente en dos formas: MSF for Agile y MSF for CMMI. También existen versiones alternativas desarrolladas por socios de Microsoft que se basan en el mismo sistema para generar EPGs basadas en otras metodologías, por ejemplo en SCRUM. MSF for Agile es un proceso ligero pensado para proyectos con ciclos de vida breves y equipos orientados a obtener resultados que puedan funcionar sin que intermedie mucha documentación. MSF for CMMI es un proceso orientado para facilitar el nivel 3 de CMMI y se basa en una planificación más formal que la versión ágil, más documentación y productos de trabajo, más puertas de cierre, y más tiempo de trazado. Ambas instancias tratan de aportar valor utilizando para ello un desarrollo iterativo e incremental.

Inicialmente, las versiones de la MSF Process Guidance para TFS 2005 se generaban utilizando una herramienta de Microsoft llamada MSFWinBuild, la cual generaba páginas HTML a partir de la información contenida en varios ficheros XML que se gestionaban usando Microsoft Office Infopath. Sin embargo, en TFS 2008 esta herramienta fue sustituida por otra llamada VSTS Process Guidance Generator, también desarrollada por el propio personal de Microsoft y de libre distribución, mucho más rápida y fácil de personalizar. El funcionamiento de esta nueva herramienta queda reflejado en la Figura 2.7.

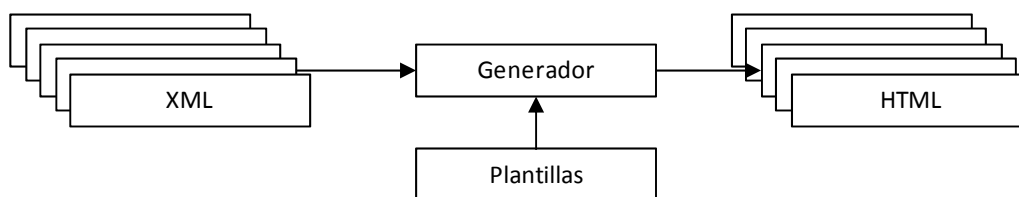


Figura 2.7: Framework de VSTS Process Guidance Generator.

A la entrada del generador se tiene un fichero en formato XML que contiene toda la información con la cual se generará la EPG. El generador utiliza además unas plantillas que son las que establecerán la estructura de cada una de las páginas que conforman la EPG, como son por ejemplo los menús o la distribución de los propios contenidos en distintas páginas.

La EPG que se obtiene a la salida del generador se compone de varias páginas HTML con información general de la guía de procesos, y enlaces a las páginas individuales de cada elemento. Los distintos elementos que contienen estas guías de procesos se dividen en: secuencias de trabajo, actividades, funciones de los miembros del equipo de trabajo, elementos de trabajo, productos de trabajo, informes, consultas y procedimientos.

Además, mediante una hoja de estilos se puede establecer la apariencia visual de dichas páginas. A esto hay que añadir también otros elementos que deberán quedar almacenados conjuntamente con las páginas HTML, como pueden ser imágenes que se muestren en la EPG.

### 2.3.4. ProKnowHow

ProKnowHow [Borges, 2002] es una herramienta basada en el KM para dar soporte al SPI mediante la creación de una EPG. Fue desarrollada para ayudar en la adaptación de procesos

para proyectos permitiendo que el conocimiento adquirido en dichos proyectos sea compartido.

Las principales funcionalidades de esta herramienta son [Falbo, 2004]:

- Personalizar los procesos estándar de la organización para los proyectos software.
- Almacenar y difundir el conocimiento adquirido durante la instanciación de los procesos estándar.
- Actualizar los procesos estándar en base a la realimentación de los proyectos.
- Ofrecer gestión de proyecto y algunas funcionalidades para acceder a la planificación de proyectos anteriores y a sus lecciones aprendidas.
- Ofrecer estimación de proyectos y métricas.

ProKnowHow almacena el conocimiento formal e informal en la base de datos de procesos de software de la organización utilizando dos ontologías. Por un lado se almacena el conocimiento formal, compuesto por activos de proceso y artefactos del proceso de software, y por otro lado se almacena el conocimiento informal, compuesto por las lecciones aprendidas.

Esta herramienta ayuda al jefe de proyecto a adaptar los procesos estándar de la organización a cada proyecto, sugiriendo modelos de ciclo de vida, actividades, procedimientos, etc. El procedimiento para confeccionar un proceso en ProKnowHow se describe a continuación. En primer lugar, el jefe de proyecto debe especificar las características del proyecto (tipo de personal, complejidad del problema, etc.) así como el perfil de calidad del mismo. Una vez caracterizado el proyecto, la herramienta sugiere un modelo de ciclo de vida adecuado que será aceptado o rechazado por el jefe de proyecto. Después de seleccionar el modelo de ciclo de vida, el jefe de proyecto puede añadir o quitar actividades al proceso. Tanto para seleccionar el ciclo de vida como las actividades, el jefe de proyecto se puede ayudar de las lecciones aprendidas almacenadas en el repositorio informal. Además, para cada actividad se deberían definir sub-actividades, artefactos de entrada y salida, procedimientos, recursos y herramientas. Una vez hecho esto, el proceso habrá quedado definido.

Respecto al conocimiento informal, existe una persona (gestor del conocimiento) encargada de filtrar las lecciones aprendidas y decidir si son útiles para convertirse en conocimiento a nivel organizacional. El procedimiento para que una lección aprendida pase a estar disponible en el repositorio de conocimiento informal se describe a continuación. Primero, el jefe de proyecto introduce una lección aprendida en el repositorio de conocimiento informal. Después, dicha lección aprendida debe ser evaluada, adaptada y aprobada por el gestor del conocimiento para estar disponible para otros jefes de proyecto.

### 2.3.5. OnSSPKR

OnSSPKR (Ontology Supported Software Process Knowledge Representation) [He, 2007] es una herramienta para la gestión de activos de proceso que se basa en ontologías para representar el conocimiento, tanto formal como informal. Los autores dividen el conocimiento en tres tipos (experiencias de procesos, artefactos de conocimiento, y habilidades personales) y construyen una ontología por cada tipo para que sirvan como vocabulario compartido y

controlado, eliminando así las posibles confusiones conceptuales y terminológicas. El repositorio de conocimiento de la organización estará entonces constituido por la acumulación de instancias en dichas ontologías.

Las principales funcionalidades de esta herramienta son por tanto:

- 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.

OnSSPKR está implementado con tecnología Java y está basado en un sistema cliente-servidor que usa un portal web. Las ontologías se construyen manualmente utilizando OWL (Ontology Web Language).

La herramienta presenta dos perfiles para su utilización: los usuarios del conocimiento, los cuales participan en las actividades de desarrollo de la organización y acceden al conocimiento, y los administradores de la organización, los cuales lideran el aprendizaje de la organización e instancian el proceso de software para un proyecto concreto, así como adaptar y aprobar el conocimiento introducido por los usuarios del conocimiento.

Más en detalle, por un lado, los usuarios del conocimiento pueden: seleccionar un concepto en el árbol de términos y almacenar conocimiento en una carpeta indexada, navegar por el árbol de términos, buscar términos concretos, y realimentar conocimiento. Por otro lado, los administradores del conocimiento pueden adicionalmente: editar las ontologías, e instanciar un proceso de software para un nuevo proyecto.

### **2.3.6. Marco Metodológico basado en Patrones de Producto (MMPP)**

En la tesis doctoral realizada por Fuensanta Medina [Medina, 2010] se plantea una solución práctica a la formalización de la definición y despliegue del proceso. Dicha formalización permite, por un lado, almacenar el conocimiento de las mejores prácticas de la Ingeniería del Software y la experiencia de los ingenieros software que han usado el proceso en proyectos anteriores y, por otro lado, permite la difusión, búsqueda, uso y reutilización de los activos de procesos en nuevos proyectos. Según Medina, el artefacto que permite almacenar, empaquetar, proporcionar y reutilizar este conocimiento es el denominado “Patrón de Producto”.

En esta tesis, aparte de definir el concepto de patrón de producto, también se definió una taxonomía de patrones de producto así como las reglas que gobiernan la recuperación de los mismos para su uso en proyectos software. Además se proporcionó un catálogo de patrones de producto que sirve como repositorio de conocimiento de los activos de procesos y proyectos, el cual puede constituir una base de conocimiento para casos en los que las organizaciones no dispongan de conocimiento acerca de su propio proceso de desarrollo.

Por otro lado, la solución propuesta en esa tesis doctoral también incluía una definición de la estrategia corporativa para el despliegue de la solución que describe los procesos para realizar

la difusión, uso y reutilización de los activos de proceso en una organización, lo que permite una mayor organización en el día a día del equipo de trabajo.

Finalmente, también se construyó una plataforma colaborativa para dar soporte a la solución propuesta que permite una mejora en la comunicación entre los miembros del equipo de desarrollo.

Las principales conclusiones alcanzadas por Medina fueron que:

- El uso de los patrones de producto mejora la eficiencia de uso de los procesos software, mejora la calidad de los productos software, y reduce el tiempo de rehacer dichos productos software.
- La productividad de las organizaciones se incrementa debido al aumento de la eficiencia de uso de los procesos software. Esta aportación es debida no solo a la utilización de los patrones de producto sino también al uso de la estrategia corporativa y a las funcionalidades colaborativas de la plataforma desplegada.

### 2.3.7. PAL-Wiki

En la tesis doctoral realizada por Leonardo Bermón [Bermón, 2010] se presenta PAL-Wiki, una solución que propone una PAL basada en una tecnología Web 2.0, la wiki, utilizando técnicas de gestión del conocimiento para definir y desplegar el proceso software con el objetivo de ayudar en el aprendizaje y uso de dichos procesos.

Según el autor, PAL-Wiki permite facilitar un ambiente de aprendizaje efectivo para ayudar a los ingenieros de software a conocer y aprender un proceso software definido en una organización con un mayor grado de independencia, dar soporte a los usuarios del proceso en la definición y despliegue de activos de procesos, proporcionar mecanismos para compartir conocimiento sobre procesos software, y ofrecer un repositorio de conocimiento con artefactos útiles y de fácil acceso para su reutilización en proyectos software.

En esa tesis doctoral se definió un modelo del proceso de gestión del conocimiento sobre los activos de proceso gestionados en una PAL. El proceso se implementa de forma gradual y está orientado a la adquisición, organización, distribución, utilización, preservación y medición del conocimiento de procesos genéricos de software. Además se propuso un modelo estructural del conocimiento que debe poseer una PAL que utiliza técnicas de gestión del conocimiento por medio de una guía de procesos que articula los usuarios, los procesos de gestión del conocimiento y los activos de proceso asociados. Ambos modelos (estructural y de procesos) sirven como guías para implementar futuras PAL-Wiki ya que los modelos de referencia sobre procesos software indican lo que una PAL debe incluir pero no cómo implementarlas.

Los resultados obtenidos por Bermón implican que aplicar un sistema wiki como repositorio de conocimiento para implementar una PAL tiene ventajas para el aprendizaje del proceso de software. La PAL-Wiki ofrece una infraestructura fácil de usar para almacenar, actualizar y acceder al conocimiento del proceso mejorando así el uso de los procesos y, además, motiva a los ingenieros a explorar e investigar conceptos autónomamente para interiorizar e institucionalizar el conocimiento del proceso software en la organización.

Por otro lado, el uso de la PAL-Wiki durante el desarrollo de proyectos influye en los niveles de calidad de los productos de trabajo desarrollados. Además, los resultados de los experimentos realizados por Bermón muestran que el aprendizaje de los procesos está relacionado con los puntos de acceso, la cantidad de accesos, el tiempo de permanencia y la cantidad de activos descargados.

### 2.3.8. Experience Factory (EF)

Victor Basili introdujo el concepto de “Experience Factory” [Basili, 1994] para referirse a aquellas infraestructuras destinadas a la reutilización de experiencias almacenadas en un repositorio. Más formalmente, los repositorios de experiencias se definen como organizaciones lógicas y/o físicas que dan soporte al desarrollo de proyectos analizando y sintetizando toda clase de experiencias recogidas durante la ejecución de proyectos [Basili, 2002].

Las factorías de experiencias actúan entonces como repositorios con el objetivo de comprender, planificar, predecir, controlar, verificar y mejorar los procesos [Visaggio, 2009]. Así, las factorías de experiencias posibilitan el aprendizaje en la organización a partir de sus propias experiencias.

Estos repositorios deben almacenar experiencias en un formato que facilite la reutilización por parte de los usuarios. Para conseguir esto, en [Schneider, 2003] se presentan 5 factores que a menudo limitan la calidad de estos repositorios:

- Los usuarios esperan el sistema proporcione apoyo directo a su trabajo o tarea, es decir, esperan ser guiados. Quieren comenzar a reutilizar de inmediato, sin tener que analizar las experiencias.
- El sistema debe ser usable para los usuarios para que puedan utilizarlo como se pretende y obtengan beneficio de ello.
- El sistema debe ser conforme al proceso al que se supone da soporte.
- Las experiencias necesitan ser actualizadas continuamente, así que el repositorio debe permitir y fomentar la realimentación.
- El mantenimiento debe ser frecuente para que nuevos contenidos sean analizados e integrados lo antes posible.

Varios son los autores que han propuesto recientemente soluciones basadas en la EF de Basili, como son [Ardimento, 2009] o [Ivarsson, 2012].

En el primer artículo, Ardimento et al. [Ardimento, 2009] presentan un KMS basado en una EF para formalizar y empaquetar conocimiento y experiencias fomentando la explicación gradual del conocimiento tácito adquirido por las personas para facilitar su transferencia. La validación de esta solución solamente ha sido realizada en entornos académicos, y los autores destacan la necesidad de que sea validada en otros entornos.

En el segundo artículo, Ivarsson y Gorschek [Ivarsson, 2012] presentan una solución basada en la EF para diseminar y mejorar las prácticas utilizadas en una organización. Las experiencias tenidas durante el desarrollo de los proyectos son capturadas mediante revisiones post-

mortem para dar soporte a la toma de decisiones, permitiendo identificar qué prácticas funcionan bien y qué prácticas necesitan mejoras. La evaluación inicial de esta solución realizada en dicho artículo muestra que los post-mortems no tienen mucho efecto en la mejora de las prácticas de la organización, aunque la compartición entre distintos equipos de trabajo de las distintas experiencias recogidas puede influenciar en las decisiones acerca de qué prácticas utilizar en cada proyecto.

### 2.3.9. Proceso de Realimentación de Experiencias (PRE)

Bernard Kamsu-Foguem et al. destacan en [Kamsu-Foguem, 2008] la importancia que tienen en la mejora continua el aprovechar y el compartir aquel conocimiento que resulta de la realimentación de experiencias anteriores. En su artículo, proponen la formalización de un proceso de realimentación de experiencias para transformar en conocimiento explícito la información ganada a por medio de la experiencia. Gracias al uso de la semántica, el conocimiento explícito se clarifica con el objetivo de adquirir descripciones de lecciones aprendidas que sean significantes, correctas y aplicables para mejorar los objetivos de una empresa.

El proceso de realimentación de experiencias propuesto está compuesto por tres capas:

- Capa del dominio: consistente en la enumeración de conceptos y sus relaciones entre ellos. Esta capa estará representada por una ontología que materializa el conocimiento del dominio.
- Capa de inferencia: vista como una biblioteca de métodos para la resolución de problemas y de procesos descritos de manera declarativa. Mediante el uso de gráficos conceptuales, los mecanismos de esta capa están orientados a la adquisición, recuperación y reutilización de experiencias.
- Capa de tareas: proporciona una interpretación procedimental de la capa de inferencia. Los principales sub-procesos de esta capa serían la adquisición de experiencias (descripción del evento, descripción del contexto, análisis y determinación de la solución) y la explotación de experiencias (recuperación, adaptación y generalización).

La metodología propuesta solo está validada mediante un ejemplo sencillo, por lo que necesita ser evaluada en contextos industriales reales y complejos.

### 2.3.10. KiWi Systems

Karsten Jahn y Peter Axel Nielsen presentan en [Jahn, 2011] los sistemas KiWi (Knowledge in a Wiki), un prototipo de KMS que aplica distintas estrategias de gestión del conocimiento en función del nivel organizativo sobre el que se aplican. Las estrategias contempladas son la codificación (enfocada a los documentos y a su disponibilidad) y la personalización (enfocada a las personas y a la colaboración entre ellas). Así, en el nivel de gestión predominará la estrategia de codificación sobre la personalización, y en el departamento de desarrollo predominará la estrategia de personalización sobre la codificación.



Los sistemas KiWi combinan una aplicación para la gestión de proyectos, una wiki y un agente para el intercambio de datos. Los desarrolladores tendrían acceso solamente a la wiki, la cual utilizarían para compartir información, comunicarse y cooperar. El jefe de cada proyecto utilizará el agente para intercambiar (manualmente) datos entre la wiki y la aplicación para la gestión de proyectos, manteniendo ambas actualizadas. La alta dirección podría acceder tanto a la wiki como a la aplicación para la gestión de proyectos.

KiWi no interfiere en las tareas a realizar por los componentes de los equipos de desarrollo, pero:

- Los miembros del equipo de proyecto ganan una herramienta (wiki) que les ayuda en la colaboración diaria y en la transmisión de conocimiento entre ellos.
- El jefe del proyecto tiene una nueva responsabilidad ya que debe intercambiar datos entre la wiki y la aplicación para la gestión de proyectos. Con este mecanismo, el jefe del proyecto tiene la posibilidad de ofrecer a otras personas el acceso a datos que anteriormente no estaban disponibles.
- La alta dirección (que trabaja con la aplicación para la gestión de proyectos) no notará ninguna diferencia, aunque la calidad de los datos que tendrán disponibles será mayor.

Aun cuando la wiki no mejore la comunicación entre los desarrolladores, sí que facilitará la comunicación entre distintos niveles organizacionales. Esto quiere decir que, aun en el caso de haber rechazo por parte de los desarrolladores, el sistema permite reducir parte del trabajo intelectual que requiere la gestión del proyecto al dar soporte a la comunicación entre la alta dirección y el departamento de desarrollo.

La evaluación de esta solución fue realizada en laboratorio simulando un entorno real y utilizando datos originales. La evaluación estuvo dividida en tres sesiones, al final de las cuales se realizaron entrevistas detalladas a las personas que probaron la solución con el objetivo de investigar la utilidad del sistema. Aunque los principios en los que se basa la solución parecen funcionar y pueden resultar beneficiosos, surgieron algunos problemas relacionados con la usabilidad y con la seguridad de los sistemas.

### **2.3.11. Eclipse Process Framework (EPF)**

Eclipse Process Framework [EPF, 2012] es un proyecto de código abierto que pretende ofrecer un marco de trabajo personalizable y herramientas de autor para la ingeniería de procesos de software (autoría de métodos y procesos, gestión de bibliotecas, configuración y publicación de procesos). A pesar de ser abierto, el 99% de la contribución al proyecto proviene de IBM Rational. El EPF incluye la herramienta EPF Composer basada en Eclipse y proporciona además procesos pre-configurados basados en desarrollos ágiles, incrementales e iterativos, como OpenUP, eXtreme Programming o Scrum.

El EPF Composer proporciona capacidades de ingeniería de procesos, así como una base de conocimiento del capital intelectual que permite examinar, gestionar y desplegar contenidos. Estos contenidos y los procesos creados con el EPF Composer pueden ser publicados y desplegados como EPG en forma de sitios web en HTML, o también pueden ser exportados a

una plantilla que se integre con una herramienta de gestión de proyectos como Microsoft Project [Haumer, 2006].

IBM ofrece además dentro de la suite Rational una versión comercial del EPF Composer llamada Rational Method Composer (RMC) [RMC, 2012]. Todas las características del EPF Composer están también disponibles en el RMC, aunque este último proporciona algunas capacidades empresariales adicionales, y proporciona como proceso el IBM Rational Unified Process (RUP). Por otro lado, el RMC permite también la publicación de sus contenidos en una EPG con formato Word o PDF, aunque el proceso es en general muy lento. Sin embargo, la carencia principal que presenta el EPF Composer respecto al RMC es la falta de integración con otras herramientas. Mientras la solución comercial RMC puede conectarse con otras herramientas de IBM Rational, las opciones del EPF Composer solo pueden exportarse a Microsoft Project. Así, el EPF Composer se centra más en únicamente documentar el proceso, haciendo que la gestión de un proyecto requiera de un mayor tiempo al tener que repetir contenidos en diversas herramientas, perdiendo también en parte la exactitud de lo definido en el EPF Composer y reutilizando en menor forma los contenidos que con el RMC. Esta sección se centra en exponer las características del EPF Composer, siendo también aplicables al RMC por su similitud.

El EPF hace distinción entre métodos y procesos. Los métodos describen lo que debe conseguirse, las habilidades que se requieren, y la explicación paso a paso de cómo lograr los objetivos específicos de desarrollo; estos contenidos son por tanto las descripciones de tareas, roles, productos de trabajo, guías, etc. involucrados en la realización de un trabajo. Los procesos en cambio describen el ciclo de vida del desarrollo, tomando los elementos de los métodos y relacionándolos en secuencias semi-ordenadas que son personalizadas para tipos concretos de proyectos. La unión de los métodos y de los procesos da lugar al método (por ejemplo, RUP), el cual proporciona así tanto la descripción del trabajo como el orden de dicho trabajo.

En el EPF Composer, los métodos están organizados en forma de árbol, permitiendo un rápido acceso a los elementos disponibles. Al seleccionar un método en el árbol, dentro del EPF Composer se muestra a través de formularios la información correspondiente al método seleccionado. Cada campo relacionado con un método puede ser editado en forma de texto, o incluso permite añadir tablas, imágenes o hipervínculos. Por ejemplo, para una tarea se muestra el propósito, los pasos necesarios para alcanzar dicho propósito, y las relaciones (en forma de hipervínculos) de dicha tarea con los roles que la llevan a cabo y con los productos de trabajo que sirven como entrada o salida para dicha tarea. Además, el EPF Composer permite añadir guías, que serán elementos complementarios como pueden ser documentos técnicos, plantillas, ejemplos, listas de comprobación, etc.

En cuanto a los procesos, el EPF Composer permite definir, configurar y publicar procesos de desarrollo de software propios utilizando un esquema predeterminado denominado UMA (Unified Method Architecture), que es una evolución de SPEM. Estos procesos están representados en forma de flujos de trabajo o en estructuras de desglose (como por ejemplo, un WBS). Para cada proceso, el EPF Composer muestra las actividades que lo componen y tareas que contiene cada actividad, así como los roles involucrados y la lista de entradas y

salidas, de forma que dichos elementos pueden ser personalizados dependiendo del punto dentro del ciclo de vida en el que dicha tarea se lleve a cabo.

Para definir un proceso, el EPF Composer permite varias opciones: partir de una metodología pre-configurada que se adapte a nuestras necesidades (OpenUP, eXtreme Programming o Scrum), confeccionar nuestro propio proceso reutilizando elementos de dichas metodologías, o crear un nuevo proceso desde cero. De la misma forma, también se pueden reutilizar o crear nuevos métodos.

En [Phongpaibul, 2007] se hace referencia a la utilización del EPF Composer para la creación de una EPG. De hecho, en el artículo se menciona la implementación de dos EPGs para una asignatura de Ingeniería del Software en la Universidad de California del Sur. Por un lado, generan una EPG usando Spearmint para modelar el conjunto de directrices conocido como MBASE (Model-Based Systems Architecting and Software Engineering). Por otro lado, generan otra EPG usando EPF Composer para modelar el conjunto de directrices conocido como LeanMBASE (versión simplificada de MBASE). El artículo realiza una comparación entre ambos generadores de EPG, y resalta que, al ser el EPF Composer una herramienta de generación de EPGs más reciente que Spearmint, proporciona una mayor capacidad para modelar procesos y es más sencilla de utilizar.

Recientemente se ha añadido la posibilidad de desplegar los contenidos creados con el EPF Composer también en forma de wiki, lo cual permite desplegar una PAL. La EPF Wiki permite a los usuarios navegar y colaborar en la publicación de contenidos, añadir comentarios, modificar los contenidos existentes, crear nuevas páginas, etc. De esta manera, se ofrece una infraestructura para la ingeniería de procesos que combina un método de construcción modular con la sencillez de uso propia de las wikis.

La EPF Wiki se puede instalar sobre Windows usando InstantRails (solución que combina Ruby, Rails, Apache, y MySQL ya pre-configurados y listos para ser ejecutados), aunque se recomienda usar Linux como entorno de producción para Rails. Como complemento a la EPF Wiki se puede integrar la herramienta TinyMCE, que es un editor WYSIWYG basado en JavaScript.

EPF Wiki ofrece también una gestión de usuarios sencilla mediante 3 roles: administrador principal, administrador (destinado a los ingenieros de procesos) y usuario. El rol de administrador tiene algunos permisos especiales, ya que por ejemplo puede hacer uso de una de las principales características que ofrece la EPF Wiki, la recolección (“harvesting”), que consiste básicamente en revisar los cambios, las nuevas páginas, la subida de ficheros y la actualización de la biblioteca de activos de procesos usando el EPF Composer.

### 2.3.12. IRIS Process Author (IPA)

IRIS Process Author [IPA, 2012] es una herramienta comercial que permite la creación, edición y publicación de procesos. Aunque está enfocada a la gestión de procesos, también permite exportar el contenido de los procesos a una herramienta de gestión de proyectos como Microsoft Project o Visual Studio Team System. Junto con esta herramienta también se

proporciona el IRIS Process Central, que actúa como repositorio para los activos de los procesos. Así, IPA es un sistema que permite crear una PAL para las organizaciones.

IPA utiliza SPEM para definir los procesos, los cuales quedan almacenados en paquetes. Dentro de estos paquetes estarán definidos los roles, productos de trabajo, tareas, guías, etc. relacionados con el proceso en cuestión, siendo posible importar contenidos de otros paquetes.

Los propietarios de los procesos serán los responsables de definir y crear el conjunto de procesos estándar de la organización. Posteriormente, para hacer un uso efectivo del proceso en un proyecto, el jefe de proyecto se encargará de confeccionar un proceso a medida según las características del proyecto a partir del conjunto de procesos estándar de la organización almacenados en el repositorio de activos de proceso.

Por otro lado, los usuarios de los procesos nunca pueden acceder directamente a la herramienta, sino que deben utilizar la versión publicada de la EPG. Esta versión puede ser HTML, pero también existen una versión wiki que proporciona un entorno colaborativo. Mediante la wiki, los usuarios de los procesos pueden proporcionar realimentación o sugerir mejoras mediante la modificación de la versión publicada, aunque estos cambios no se verán reflejados en los modelos almacenados en la herramienta.

Esta herramienta está completamente basada en Web, proporcionando un repositorio de procesos online para facilitar el trabajo colaborativo. Sin embargo, la herramienta no permite que dos usuarios modifiquen el mismo proceso simultáneamente. Además, parece presentar algunos problemas con navegadores diferentes de Internet Explorer.

## 2.4. Análisis de soluciones existentes y conclusiones

Anteriormente se han presentado varias soluciones tecnológicas que aplican la gestión del conocimiento para la gestión y mejora de procesos de software. En este apartado se van a analizar más en detalle cada una de estas soluciones, teniendo en cuenta los aspectos concretos que se indican a continuación clasificados según las distintas fases del modelo SECI:

- Socialización:
  - Posibilidad de gestionar el conocimiento de manera colaborativa, admitiendo contribuciones para el enriquecimiento del conocimiento organizacional provenientes de equipos de trabajo distribuidos a lo largo de la organización.
- Externalización:
  - Facilidad para actualizar el conocimiento almacenado, admitiéndose la realización de cambios en el conocimiento ya existente en el repositorio de conocimiento.
  - Capacidad de realimentación del sistema, permitiendo el enriquecimiento del conocimiento organizacional mediante contribuciones provenientes de la ejecución de los proyectos.

- **Combinación:**
  - Facilidad para introducir conocimiento en el sistema, valorando si proporcionan mecanismos sencillos para que cualquier persona tenga la posibilidad de contribuir con nuevo conocimiento o si por el contrario es necesario adquirir alguna capacidad concreta para su utilización.
  - Gestión de la configuración sobre los contenidos de la herramienta, analizando la posibilidad de establecer varias líneas base a partir de un mismo repositorio de conocimiento y de gestionar después los cambios provenientes de cada línea base para enriquecer conjuntamente el conocimiento organizacional.
- **Internalización:**
  - Presentación del conocimiento al usuario, analizando si la forma de presentar el conocimiento es comprensible y usable, permitiendo incluso la presentación del conocimiento en distintos formatos dependiendo de las necesidades de los usuarios.
  - Posibilidad de personalización de la interfaz para los usuarios del proceso, valorando si se permite o no la reorganización de los elementos de la interfaz o el cambio de aspecto visual de la misma.
  - Facilidad para reutilizar contenidos introducidos previamente en el sistema, distinguiendo si la solución permite de manera sencilla la reutilización del nuevo conocimiento capturado o si solo permite la reutilización del conocimiento inicial del sistema.
  - Facilidad de aprendizaje del uso del sistema, analizando si la solución ofrece unas funcionalidades sencillas e intuitivas o si se requiere un cierto aprendizaje para poder utilizarlas de manera efectiva.

La Tabla 2.3 resume los resultados del análisis realizado.

**Tabla 2.3. Resumen del análisis de soluciones existentes.**

		Spearmint	WAGNER	MSF	ProKnowHow	OnSSPKR	MMPP	PAL-Wiki	EF	PRE	Kiwi Systems	EPF	IPA
S	Colaboración	x	x	x	x	x	✓	-	✓	✓	✓	x	-
E	Actualización de contenidos	✓	✓	✓	-	-	-	✓	x	x	✓	✓	✓
	Realimentación	x	x	x	✓	-	✓	-	✓	✓	✓	✓	✓
C	Inserción de contenidos	x	x	✓	-	-	-	-	✓	-	✓	x	x
	Gestión de la Configuración	x	x	x	x	x	-	-	x	x	-	-	-
I	Presentación de contenidos	-	-	-	-	-	✓	✓	-	✓	✓	✓	✓
	Personalización de interfaz	-	✓	✓	-	-	✓	✓	✓	-	✓	x	x
	Reutilización de contenidos	x	x	x	x	✓	✓	✓	✓	✓	✓	-	-
	Aprendizaje del sistema	x	x	-	-	-	✓	✓	✓	✓	✓	x	-

Leyenda:      ✓ Satisfactorio                      x No satisfactorio                      - Mejorable

A continuación se describen en detalle los resultados del análisis en base a cada uno de los aspectos tenidos en cuenta.

### ***Posibilidad de gestionar el conocimiento de manera colaborativa***

Solo algunas de las soluciones mencionadas permiten la gestión del conocimiento de manera colaborativa entre distintos equipos de trabajo a lo largo de la organización.

IPA permite sincronizar la modificación de contenidos mediante un sistema de bloqueo-modificación-desbloqueo, el cual evita que un usuario pueda editar un elemento bloqueado hasta que el usuario que lo está modificando no lo desbloquee. El problema que presenta este sistema es que si el bloqueo abarca gran cantidad de contenidos, tan solo el usuario que lo bloquee primero podrá modificar dichos contenidos.

Varias soluciones, como KiWi, el PRE, las EFs, el MMPP y la PAL-Wiki, permiten la gestión del conocimiento de manera colaborativa ya que fueron diseñadas a tal efecto, aunque algunas presentan limitaciones. KiWi permite la colaboración entre los desarrolladores al utilizar la wiki, pero requiere de la intervención del jefe de proyecto para que los datos introducidos queden a disposición de la alta dirección a través de la aplicación de gestión de proyectos. De forma similar, el PRE hace que el jefe de proyecto sea el único responsable de realimentar nuevo conocimiento al sistema. Por último, la PAL-Wiki permite la gestión colaborativa del conocimiento pero solo dentro del colectivo de personas que utilice la misma wiki, no admitiendo el intercambio de conocimiento entre varias PAL-Wiki en el caso de haber más de una desplegada.

### ***Facilidad para actualizar el conocimiento almacenado***

En cuanto a la posibilidad de que los contenidos relativos a los procesos evolucionen, en Spearmint, WAGNER y EPF habría que modificar los modelos y generar una nueva versión de la EPG, o modificar directamente el HTML de la EPG (aunque esto no modificaría los modelos). En MSF ocurre lo mismo, aunque en este caso habría que modificar el fichero XML.

Las EFs ofrecen la posibilidad de registrar experiencias de proyectos a través de la propia web que permite la visualización del conocimiento, pero no mencionan ningún mecanismo para modificar las experiencias ya registradas con anterioridad.

En cuanto al PRE, una vez recogida la nueva información en forma de lecciones aprendidas, estas deben ser transformadas en gráficos conceptuales para obtener una especificación formal. Sin embargo, no se menciona ningún mecanismo para actualizar los gráficos conceptuales existentes.

Por otro lado, en ProKnowHow el gestor del conocimiento puede actualizar tanto el conocimiento formal como el informal, pero solo se indica que para el conocimiento informal el jefe de proyecto sugiere la inserción de una lección aprendida en el repositorio que tiene que ser aprobada por el gestor del conocimiento. En OnSSPKR también se puede actualizar el conocimiento, pero tampoco se detalla cómo.

En cuanto a PAL-Wiki, EPF Wiki, IPA y KiWi, al contar con versiones wiki, estas modificaciones serían muy sencillas a través de la propia wiki.

Finalmente, la modificación del conocimiento almacenado en el MMPP es sencilla, pero estos cambios implican la creación de un nuevo patrón de producto por cada cambio realizado, conservándose el patrón original.

### ***Capacidad de realimentación del sistema***

En general, las soluciones que generan una EPG no admiten por sí mismas la posibilidad de que los usuarios del proceso añadan ningún tipo de comentario (experiencias, lecciones aprendidas, etc.) puesto que están diseñadas solo para mostrar el conocimiento, no para capturar nuevo conocimiento, aunque en los artículos consultados sobre Spearmint y WAGNER se sugiere la herramienta Pageseeder para añadir comentarios. Por otro lado, EPF Wiki, IPA y KiWi sí que admiten una forma sencilla de insertar comentarios al contar con versiones wiki. Sin embargo, ninguna de estas soluciones permite que estos comentarios se propaguen automáticamente de la EPG o de la wiki al repositorio de conocimiento, sino que deben ser insertados manualmente.

Por otro lado, las EFs, al estar orientadas a la adquisición de nuevo conocimiento por medio de la realimentación, ofrecen la posibilidad de registrar experiencias de proyectos a través de la propia web que permite la visualización del conocimiento. Sin embargo, la EF propuesta por Ivarsson y Gorschek requiere de la realización de revisiones post-mortem, lo cual presenta una limitación ya que la realimentación del conocimiento tácito para convertirlo en explícito solo se puede realizar al final de una fase o al terminar el proyecto.

El resto de soluciones analizadas admiten realimentación del conocimiento de forma diversa: el PRE recoge nueva información que es transformada en gráficos conceptuales, el MMPP asocia el nuevo conocimiento a un patrón de producto, y la PAL-Wiki permite vincular archivos y páginas de discusión a los distintos activos de conocimiento.

### ***Facilidad para introducir conocimiento en el sistema***

En general, la mayoría de las soluciones mencionadas requieren de la formalización de los procesos de la organización mediante la creación de modelos de dichos procesos. En Spearmint estos modelos se realizan en un formato propio, para WAGNER se usan diagramas de actividad en UML, y para EPF e IPA se usa SPEM y texto, si bien este último es más personalizable que el EPF ya que permite por ejemplo añadir atributos personalizados a diferentes elementos del proceso.

Otras soluciones proponen el uso de ontologías, como OnSSPKR que utiliza OWL para construir las ontologías o el PRE que propone la combinación de una ontología y gráficos conceptuales.

Algunas soluciones utilizan descripciones textuales, como MSF que utiliza descripciones textuales que se almacenan en un único fichero XML (fichero .guidance), las distintas EFs que empaquetan conocimiento proveniente de informes técnicos, artículos y libros, o KiWi que utiliza los datos introducidos en una wiki. PAL-Wiki también permite introducir el conocimiento mediante texto pero proporciona además unos formularios para facilitar la inserción.

Finalmente, en los artículos consultados sobre ProKnowHow no se especifica el formato de los datos. Solo se dice que almacena el conjunto de actividades a realizar para el desarrollo de

software, los procedimientos para llevar a cabo dichas actividades, patrones para crear artefactos, herramientas para dar soporte a las actividades, y las políticas de la organización.

### ***Gestión de la configuración sobre los contenidos de la herramienta***

En ninguna de las referencias consultadas se menciona la posibilidad de hacer gestión de la configuración sobre los contenidos.

IPA incluye un sistema de control de versiones sobre los distintos elementos de conocimiento que almacena. Dicha herramienta implementa un sistema que vuelca los contenidos antiguos a formato XML y los almacena en un sistema de control de revisiones externo, de forma que estos XML pueden ser re-importados a la herramienta en caso de ser necesario.

Por otro lado, los sistemas basados en wiki (MMPP, PAL-Wiki, KiWi Systems, EPF Wiki e IPA) proporcionan de manera integrada en sus wikis un sistema de control de versiones sobre los contenidos de cada página.

### ***Presentación del conocimiento al usuario***

La mayoría de las soluciones presentan el conocimiento almacenado en formato HTML, con algunos matices. Por ejemplo, Spearmint, WAGNER, MSF y EPF permiten generar directamente una EPG en formato HTML. Sin embargo, OnSSPKR, ProKnowHow y las distintas EFs cuentan con una interfaz web para acceder directamente al conocimiento, pero no permiten generar una versión de la EPG como tal.

IPA puede presentar el conocimiento en otros formatos como Word o PDF, e incluso crear una wiki. EPF también tiene un formato wiki, pero para ello hay que desplegar primero la EPG Wiki y luego hay que importar la EPG a la wiki.

Por otro lado, KiWi combina la presentación de datos en forma de wiki para los desarrolladores con la presentación de datos para la alta dirección que proporcione la propia aplicación de gestión de proyectos.

Finalmente, el PRE utiliza gráficos conceptuales para visualizar el conocimiento almacenado.

### ***Posibilidad de personalización de la interfaz para los usuarios del proceso***

WAGNER y MSF permiten personalizar la forma en que se presenta a los usuarios el conocimiento almacenado acerca del proceso utilizando para ello plantillas y hojas de estilo que modifican el aspecto que tendrán las páginas HTML correspondientes a las EPGs que generan.

Por otro lado, en los sistemas basados en wiki (MMPP, PAL-Wiki, KiWi Systems, EPF Wiki e IPA) también cabe esperar la posibilidad de personalizar la interfaz, aunque en las referencias consultadas no se menciona esta posibilidad.



### ***Facilidad para reutilizar contenidos introducidos previamente en el sistema***

Algunas de las soluciones (las EPG, principalmente) solo permiten la utilización de los contenidos inicialmente introducidos en el sistema puesto que no están diseñadas para capturar nuevo conocimiento que pudiera ser reutilizado después.

Sin embargo, muchas de las soluciones fueron diseñadas precisamente para reutilizar los contenidos, como es el caso de las EFs, el PRE, el MMPP y PAL-Wiki. En cuanto a KiWi, el sistema está diseñado para que la aplicación de gestión de proyectos pueda obtener datos de distintos proyectos. Finalmente, IPA y EPF admiten la utilización de contenidos de otros procesos.

### ***Facilidad de aprendizaje del uso del sistema***

Muchas de las soluciones y los lenguajes de modelado necesarios para insertar el conocimiento no son triviales y requieren un cierto aprendizaje para poder ser usados de manera efectiva. Para el EPF Composer en concreto la curva de aprendizaje es abrupta y la adaptación de los contenidos a las necesidades específicas es difícil.

En cuanto a las EFs, las interfaces web que ofrecen para la visualización de conocimiento y para la introducción de experiencias parecen sencillas de utilizar ya que manejan información textual.

Finalmente, la herramienta que utiliza MSF para generar la EPG también es sencilla de manejar, pero tiene el inconveniente de que la estructura del fichero XML donde se almacena el conocimiento resulta a priori algo compleja y difícil de gestionar.

## **2.5. Conclusiones del estado de la cuestión**

Como conclusión de este capítulo, se puede afirmar que las soluciones mencionadas en el apartado anterior presentan ciertos problemas respecto a la gestión del conocimiento acerca de procesos de software puesto que:

- Su elaboración requiere de mucho tiempo y esfuerzo, principalmente debido a que todo el conocimiento relacionado con los procesos de la organización se debe definir previamente de manera formal mediante modelos para poder después insertar dicho conocimiento organizacional en el repositorio de conocimiento.
- En general requieren de la presencia de un experto en procesos para la formalización de todos los procesos de la organización. La mayoría de definiciones de los procesos se realiza mediante diagramas de actividad en UML o mediante diagramas SPEM, por lo que se necesitan unos conocimientos previos a la hora de poder modelar los procesos de la organización.
- Algunas de las soluciones mencionadas no proporcionan un mecanismo sencillo para actualizar el conocimiento almacenado en el repositorio de conocimiento, ya que para realizar alguna modificación la mayoría requieren hacer cambios en los modelos de los procesos.

- Ninguna de las soluciones mencionadas no incluyen ningún mecanismo de gestión de la configuración que permita llevar un control de la evolución del conocimiento almacenado en el repositorio de conocimiento, ni tampoco la posibilidad de establecer varias líneas base a partir de un mismo repositorio de conocimiento y de gestionar después los cambios provenientes de cada línea base para enriquecer conjuntamente el conocimiento organizacional.
- En general, no permiten la inserción de conocimiento en el sistema de manera colaborativa, principalmente debido a que la mayoría de las soluciones no permiten el acceso simultáneo de varios usuarios a los diferentes contenidos almacenados. Sin embargo, todos los involucrados en un proceso deberían tener la posibilidad de contribuir a su mejora en cualquier momento. Es más, sería recomendable que la evolución de un proceso no se produjera de manera aislada en un proyecto sino de manera colaborativa con aportaciones provenientes de todos los proyectos en los que dicho proceso se lleve a cabo.
- Algunas soluciones (como las EPG) no contemplan por sí solas la adquisición de nuevo conocimiento ya que están diseñadas para que un equipo de ingenieros de proceso gestione el conocimiento acerca de los procesos y lo ponga luego a disposición de los participantes del proceso para su utilización. Este tipo de comunicación unidireccional impide la realimentación puesto que no facilita ningún mecanismo para que los ingenieros de proceso, que son los que realmente llevan a cabo los procesos y saben qué funciona y qué no funciona en cada proceso, puedan aportar sus experiencias y lecciones aprendidas para la mejora de dichos procesos.
- Aunque algunas soluciones sí están diseñadas para realimentar conocimiento a partir de los proyectos, en ocasiones esta realimentación presenta algunos inconvenientes. En ocasiones el conocimiento no se registra adecuadamente, haciendo difícil su localización; otras veces es necesaria la realización de una revisión post-mortem para realizar la realimentación; y en muchos casos, la realimentación solo se ciñe al proyecto en cuestión cuando lo deseable sería poder realimentar conocimiento entre los distintos equipos de trabajo de la organización.
- Sería recomendable además que tanto el conocimiento explícito como el conocimiento tácito formalizado y convertido en explícito fueran almacenados en un mismo repositorio organizacional y de una manera homogénea, algo que no suele ocurrir ya que habitualmente se despliegan distintos repositorios para los distintos tipos de conocimiento de la organización.
- En ocasiones no se permite personalizar la apariencia externa con la cual se presenta el conocimiento almacenado a los usuarios. Esto impide que la organización que utiliza dicha solución pueda caracterizar la herramienta con su imagen corporativa.

# Capítulo 3: Solución Propuesta

3.1. Visión General.....	67
3.2. Estructura del Conocimiento .....	68
3.3. Modelo de Procesos .....	70



En este capítulo se define ITAKA (*Interactive Tacit Knowledge Administration*), un marco de trabajo –incluyendo la especificación de roles, procesos y herramientas– para facilitar la elicitación, preservación y enriquecimiento de los activos de conocimiento de la organización que permita dar soporte a la implementación de iniciativas específicas y continuas para el aprendizaje y adopción de procesos de software. Este marco de trabajo está orientado a garantizar la alta calidad del nuevo conocimiento adquirido, asegurando la credibilidad del mismo y enriqueciendo así el conocimiento de la organización.

### 3.1. Visión General

Para alcanzar el objetivo marcado se propone un sistema para adquirir y preservar el conocimiento útil y relevante para una organización, añadiéndolo al conocimiento pre-existente de forma asistida y semi-automática, y difundir después este conocimiento enriquecido a través de distintas plataformas. La visión conceptual de este sistema se muestra en la Figura 3.1.

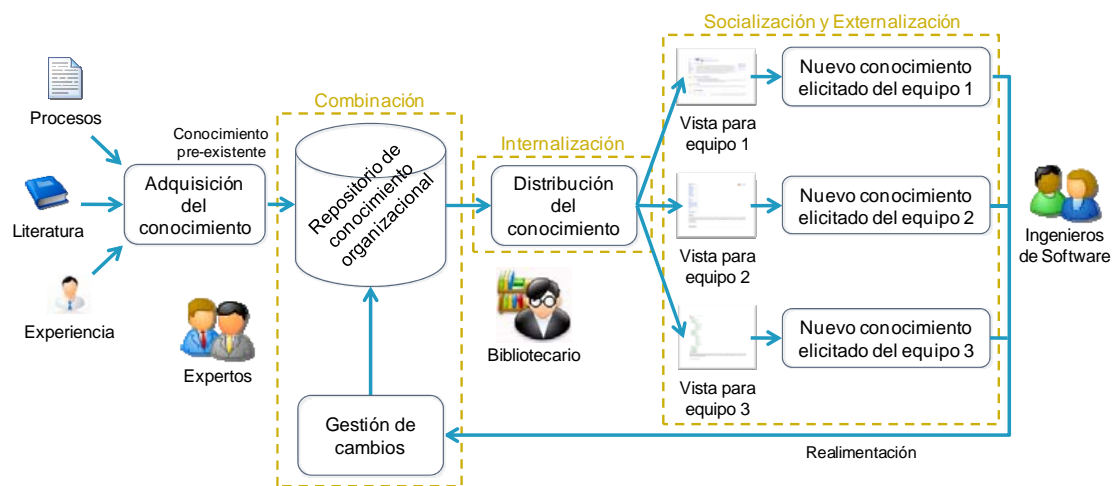


Figura 3.1. Visión conceptual de ITAKA.

El conocimiento fluye a través de ITAKA siguiendo las mismas fases que propone el modelo SECI [Nonaka, 2006]. Primero, el conocimiento pre-existente es adquirido y almacenado en el repositorio de conocimiento de la organización, permitiendo así una primera distribución de distintas vistas del conocimiento para distintos equipos de trabajo de la organización. Después, el conocimiento tácito que existe en cada una de las personas que forman parte de estos equipos de trabajo es intercambiado entre ellos (socialización) para alcanzar un mejor entendimiento de los procesos software de la organización, resultando en la creación de nuevo conocimiento. Este nuevo conocimiento tácito tiene que ser preservado, así que se formaliza y se transforma en conocimiento explícito (externalización). Una vez hecho explícito, el nuevo conocimiento se combina con el conocimiento explícito que ya estaba almacenado en el repositorio de conocimiento de la organización (combinación) y es de nuevo distribuido por las distintas vistas. Este conocimiento enriquecido puede ser reutilizado más adelante por los usuarios de las vistas, los cuales llegan a comprender y a asimilar el nuevo conocimiento (internalización). El ciclo continuaría de nuevo a la socialización cuando los individuos compartan este nuevo conocimiento tácito que acaban de adquirir.

Respecto a los roles involucrados, el Experto identifica y formaliza el conocimiento pre-existente, dándole la estructura apropiada para ser registrado en el repositorio de conocimiento de la organización; además, cuando el conocimiento tácito es realimentado, el experto también es el encargado de gestionar los cambios en el conocimiento de la organización de forma que solo el conocimiento útil y relevante puede ser incluido en el repositorio de conocimiento de la organización. Por otro lado, el Bibliotecario es el encargado de registrar el conocimiento en el repositorio, tanto el conocimiento pre-existente como los cambios que se realicen en el mismo gracias a la realimentación, y también es el encargado de distribuir a las diferentes vistas una versión de todo o parte del conocimiento de la organización almacenado en el repositorio en un instante concreto; otra responsabilidad del Bibliotecario es monitorizar el sistema para evaluar si el conocimiento de la organización es gestionado de manera adecuada y eficiente. Por último, los Ingenieros de Software aplican el conocimiento almacenado y comparten experiencias durante la realización de los distintos proyectos, generando así nuevo conocimiento tácito que es transformado en explícito para ser preservado.

Para definir ITAKA se han seguido las pautas para la construcción de una PAL basada en la gestión del conocimiento propuestas por García et al. [García, 2011]. Según afirman los autores de dichas pautas, la implementación de la PAL en una organización es un proceso gradual que se compone de tres tipos de funciones –esenciales, de apoyo y avanzadas–, las cuales se basan en características integradoras e interactivas de los sistemas de gestión del conocimiento. En los siguientes párrafos se revisan estas funciones describiendo su implementación en ITAKA.

### 3.2. Estructura del Conocimiento

El conocimiento almacenado en el repositorio de conocimiento de la organización debe estar organizado siguiendo alguna estructura determinada de manera que cualquier activo (elemento de conocimiento) pueda ser localizado dentro de dicha estructura.

El modelo que representa la estructura del repositorio de conocimiento de la organización que forma parte de ITAKA se muestra en la Figura 3.2 en forma de diagrama de clases en UML. Las ideas más importantes que subyacen en este modelo se pueden resumir como se indica a continuación.

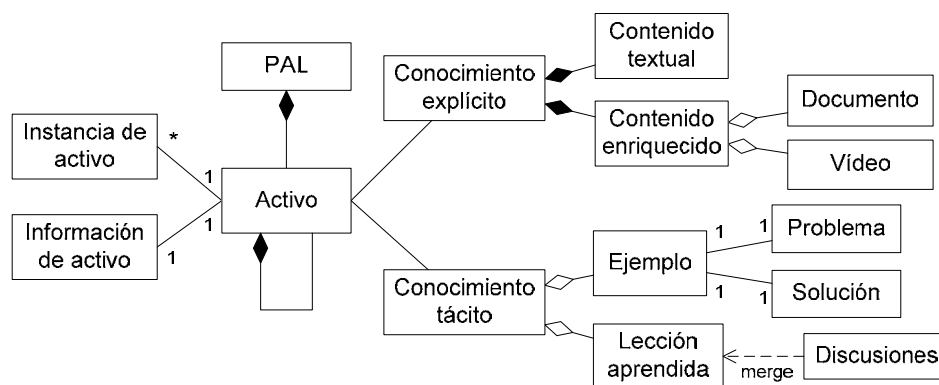


Figura 3.2. Modelo de la estructura del repositorio de conocimiento organizacional de ITAKA.

La PAL de ITAKA está formada por un conjunto de **activos**, que son aquellas entidades que son consideradas útiles por la organización. Todos los activos tienen asociada **información** acerca de su versión y de su historial de cambios. Además, un activo puede estar compuesto a su vez de otros activos. Un mismo activo puede ser **instanciado** varias veces para su reutilización en distintos proyectos, esto es, los contenidos del activo pueden ser adaptados en cada instancia para su aplicación en cada proyecto concreto. Los activos pueden contener **conocimiento explícito** (es decir, formal) y **conocimiento tácito** (es decir, personal). Por un lado, el conocimiento explícito podrá ser **textual** o **enriquecido** mediante la utilización de elementos multimedia (principalmente **documentos** y **vídeos**). Por otro lado, el conocimiento tácito se presenta en forma de **ejemplos** (obtenidos a partir de la aplicación de los activos en los proyectos y que ofrecen una posible **solución** a un **problema** concreto), y en forma de **lecciones aprendidas** durante la ejecución de cada proyecto (obtenidas a partir de las **discusiones**, es decir, a partir del intercambio de ideas y opiniones entre los miembros de un mismo equipo de trabajo).

Para almacenar estos activos, ITAKA cuenta con un único repositorio que actúa como una plataforma de control de versiones cuyo objetivo es organizar y preservar todo el conocimiento relativo a los procesos software de la organización. La Figura 3.3 muestra cómo se clasifican los activos dentro del repositorio de conocimiento de la organización, de manera que un activo puede referirse a un proceso, una actividad, un rol, un elemento de trabajo, un producto de trabajo o una instrucción técnica. Tal y como se mostró en la Figura 3.2, estos activos habitualmente contienen una descripción textual y contenido enriquecido, y algunos de ellos –actividades y procesos, mayormente– pueden también contener conocimiento tácito que ha sido formalizado y almacenado en forma de ejemplos y lecciones aprendidas. Todo el conocimiento de la organización se encuentra almacenado en este repositorio en forma de ficheros de texto y otros ficheros multimedia tales como imágenes, documentos o vídeos.

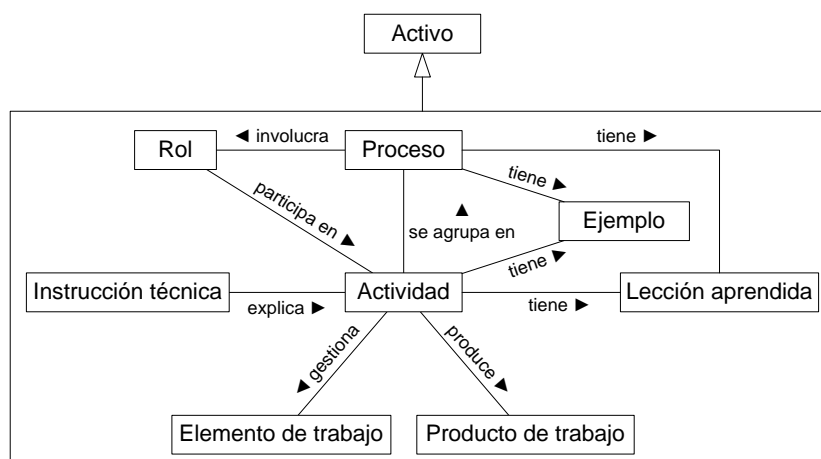


Figura 3.3. Clasificación de activos dentro del repositorio de conocimiento organizacional de ITAKA.

La estructura del repositorio de conocimiento propuesto para ITAKA hace énfasis en un conocimiento procedimental, es decir, en tareas que las personas saben cómo hacer en base a unas destrezas adquiridas gradualmente a través de la práctica. Los contenidos de dicha estructura permiten almacenar conocimiento acerca de qué acciones realizar, quién debe realizarlas, cuándo se deben realizar, y cómo pueden ser realizadas.

Así, dentro de esta estructura una actividad es un cierto trabajo que se realiza con un único propósito. Cada actividad está definida por una visión general, los roles participantes, los criterios de entrada y de salida, cuándo es realizada, las tareas que la constituyen, los elementos de trabajo que son gestionados, los productos de trabajo que son producidos, y las instrucciones técnicas que explican cómo debe llevarse a cabo la actividad.

Las actividades se agrupan en procesos. Un proceso está definido por una visión general, los roles participantes, los criterios de entrada y de salida, cuándo es realizada, y las actividades que constituyen dicho proceso.

Cada proceso involucra algunos roles. Un rol es un conjunto de competencias y responsabilidades de un individuo o de un grupo que toma parte en las actividades de un proceso. Un rol está definido por una descripción del mismo y por la lista de actividades y sus correspondientes procesos en los cuales participa dicho rol.

Por otro lado, un elemento de trabajo es un registro que se utiliza para seguir el rastro de la asignación del trabajo y del estado de dicho trabajo. Un elemento de trabajo está definido por una visión general, una descripción de sus estados y las transiciones entre ellos que describe el flujo que trabajo, un conjunto de campos relacionados con el elemento de trabajo, y la lista de actividades y sus correspondientes procesos que gestionan dicho elemento de trabajo.

Las actividades gestionan elementos de trabajo para realizar tareas que resultan en entregable o artefactos denominados productos de trabajo. Estos son ficheros, documentos, especificaciones, ficheros binarios y otros elementos tangibles que resultan necesarios para completar la realización de las actividades.

Finalmente, las instrucciones técnicas simplemente presentan información adicional acerca de cómo llevar a cabo las tareas relacionadas con las actividades.







### 3.3. Modelo de Procesos

El modelo de los procesos definidos dentro de ITAKA incluye una descripción de los distintos elementos que conforman cada uno de estos procesos. Para ello se ha utilizado la notación SPEM (Software & Systems Process Engineering Metamodel) en su versión 2.0. SPEM es un lenguaje formal promovido por el OMG (Object Management Group) que proporciona los conceptos para modelar, documentar, presentar, gestionar, intercambiar, y ejecutar métodos y procesos de desarrollo [OMG, 2008].

Los elementos SPEM que se utilizarán en los diagramas de esta sección son los que se indican en la Tabla 3.1.



Tabla 3.1. Descripción de elementos SPEM 2.0.

Notación	Descripción
	<b>Proceso:</b> componente desglosable que representa una relación entre las instancias de actividades y el uso de los roles en las instancias.
	<b>Actividad:</b> unidad general de trabajo asignable a ejecutores específicos representados por un rol.
	<b>Tarea:</b> unidad concreta de trabajo asignable y gestionable que define cómo los roles deben ejecutar el trabajo de una actividad.
	<b>Producto de trabajo:</b> elemento que es usado, modificado o producido por un proceso o actividad.
	<b>Rol:</b> conjunto de habilidades, competencias y responsabilidades de un individuo o conjunto de ellos que lleva a cabo un proceso o actividad.
	<b>Herramienta:</b> instrumento que es utilizado durante la ejecución de un proceso o actividad.

Partiendo de la visión conceptual presentada con anterioridad en la Figura 3.1 se han definido un conjunto de procesos dentro de ITAKA para facilitar la administración del conocimiento útil y relevante para una organización. Estos procesos se muestran en la Figura 3.4. Primero tiene lugar la adquisición del conocimiento, definida mediante los procesos de *identificación* y de *formalización* del conocimiento, y dando como resultado un conocimiento pre-existente que queda registrado en el repositorio de conocimiento organizacional. Este conocimiento inicial pasa entonces por un proceso de *distribución* mediante el cual cada se pone a disposición de cada uno de los equipos de trabajo dentro de la organización una vista de dicho conocimiento adaptada a sus necesidades concretas. La *aplicación* de este conocimiento en el ámbito de cada uno de los proyectos de la organización puede dar lugar a la creación de nuevo conocimiento tácito, el cual será elicitado mediante un proceso de *preservación*. Más adelante se produce una realimentación por la cual un proceso de *gestión de cambios* permite que todo el nuevo conocimiento preservado por cada uno de los equipos de trabajo se fusione con el conocimiento pre-existente almacenado en el repositorio de conocimiento organizacional. En paralelo a estos procesos se lleva a cabo además una *medición* que permite evaluar si el conocimiento de la organización es gestionado de manera adecuada y eficiente. En los siguientes apartados se describen en detalle cada uno de estos procesos.

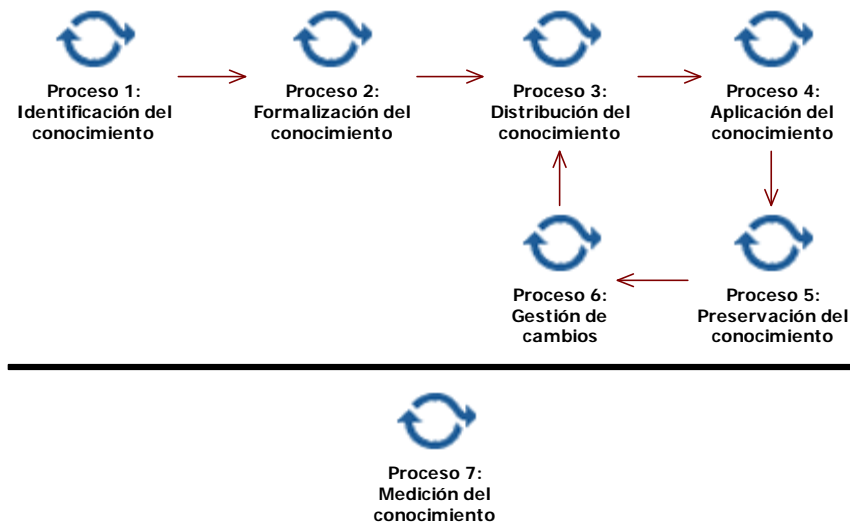





Figura 3.4. Procesos definidos dentro de ITAKA.

Para poder llevar a cabo todos estos procesos es necesario involucrar a determinados roles que participen en las distintas actividades que conforman dichos procesos. En concreto, los roles y responsabilidades definidos dentro de ITAKA son los que se indican en la Tabla 3.2.

Tabla 3.2. Roles y responsabilidades dentro de ITAKA.

Roles	Responsabilidades
 Experto	<ul style="list-style-type: none"> <li>- Recopilar procesos para la identificación del conocimiento</li> <li>- Recopilar bibliografía para la identificación del conocimiento</li> <li>- Recopilar experiencias para la identificación del conocimiento</li> <li>- Crear activos de conocimiento para su formalización</li> <li>- Relacionar activos de conocimiento para su formalización</li> <li>- Revisar cambios en los activos durante la gestión de cambios del conocimiento</li> </ul>
 Bibliotecario	<ul style="list-style-type: none"> <li>- Registrar activos de conocimiento para su formalización</li> <li>- Seleccionar activos de conocimiento para su distribución</li> <li>- Integrar activos de conocimiento para su distribución</li> <li>- Generar vistas para la distribución del conocimiento</li> <li>- Identificar cambios en los activos para la gestión de cambios del conocimiento</li> <li>- Incorporar cambios en los activos durante la gestión de cambios del conocimiento</li> <li>- Medir el conocimiento para determinar la efectividad del sistema</li> <li>- Medir el conocimiento para determinar la calidad del conocimiento</li> <li>- Medir el conocimiento para determinar la satisfacción de los usuarios</li> </ul>
 Ingeniero de software	<ul style="list-style-type: none"> <li>- Consultar activos de conocimiento para su aplicación en los proyectos</li> <li>- Instanciar activos de conocimiento para su aplicación en los proyectos</li> <li>- Registrar ejemplos para la preservación del conocimiento</li> <li>- Participar en discusiones para la preservación del conocimiento</li> <li>- Editar activos para la preservación del conocimiento</li> </ul>

## Proceso 1: Identificación del Conocimiento

Uno de los problemas clave que aparece repetidamente en la literatura relacionada con la gestión del conocimiento es la dificultad que existe para estimular a las personas a utilizar los repositorios de conocimiento y a participar en el enriquecimiento de los mismos mediante nuevas contribuciones. En primer lugar, no existe ningún motivo por el cual una persona quiera utilizar un repositorio de conocimiento si este no dispone de contenidos relevantes almacenados. Por otro lado, es difícil obtener contenidos relevantes si no se cuenta con la participación de los usuarios en el repositorio. Así, queda de manifiesto la importancia de alimentar el repositorio de conocimiento con un contenido inicial. Para ello, una organización puede utilizar distintos métodos, uno de los cuales se basa en introducir en el repositorio un conocimiento pre-existente [Petter, 2008].

### Objetivo del proceso

La identificación de conocimiento tiene por tanto como objetivo la obtención y caracterización de este conocimiento pre-existente, el cual es necesario para llevar a cabo los procesos de la organización. El conocimiento identificado será posteriormente almacenado en el repositorio de conocimiento de la organización de manera que pueda ser comunicado entre las personas para su uso continuo en la organización.

El alcance de este proceso se puede apreciar en la Figura 3.5.



Figura 3.5. Proceso 1: Identificación del Conocimiento.

### Entradas requeridas

El conocimiento pre-existente se obtiene de los procesos de desarrollo de software de la organización, de las prácticas más efectivas tomadas de la literatura existente acerca de la ingeniería del software, y de la experiencia de los expertos más relevantes dentro de la organización.

### Roles participantes

El Experto es el encargado de identificar el conocimiento pre-existente.

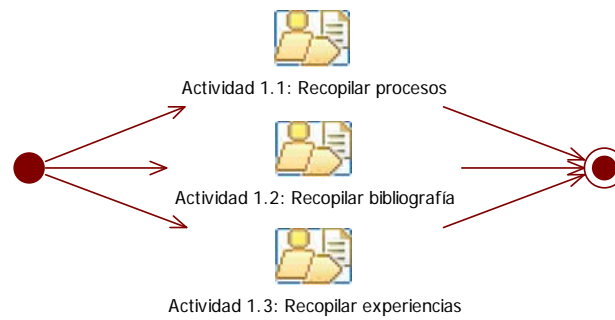
**Actividades a realizar**

Figura 3.6. Actividades del Proceso 1: Identificación del Conocimiento.

**Salidas esperadas**

Como resultado de la identificación del conocimiento se obtiene el conjunto de activos que conforman el conocimiento pre-existente de la organización.

## Actividad 1.1: Recopilar procesos

### Diagrama de la actividad

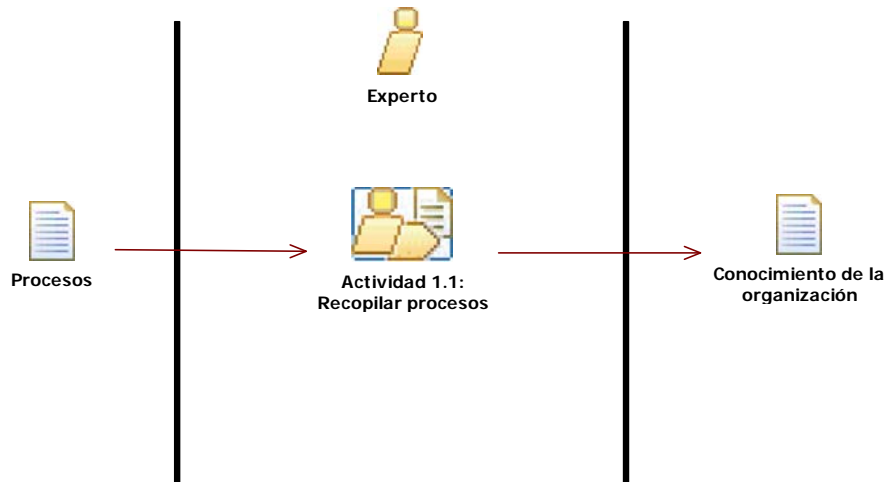


Figura 3.7. Actividad 1.1: Recopilar procesos.

### Entradas requeridas

Procesos de desarrollo de software de la organización.

### Roles participantes

Experto.

### Tareas a realizar

Recopilar información que permita definir los procesos que se llevan a cabo en la organización, para lo cual es necesario identificar qué procesos, técnicas y herramientas hay disponibles, cuáles de ellos resultan de utilidad y dónde está localizada toda la información referente a los mismos.

### Salidas esperadas

Conjunto de activos que representa el conocimiento acerca de los procesos de la organización.

## Actividad 1.2: Recopilar bibliografía

### Diagrama de la actividad

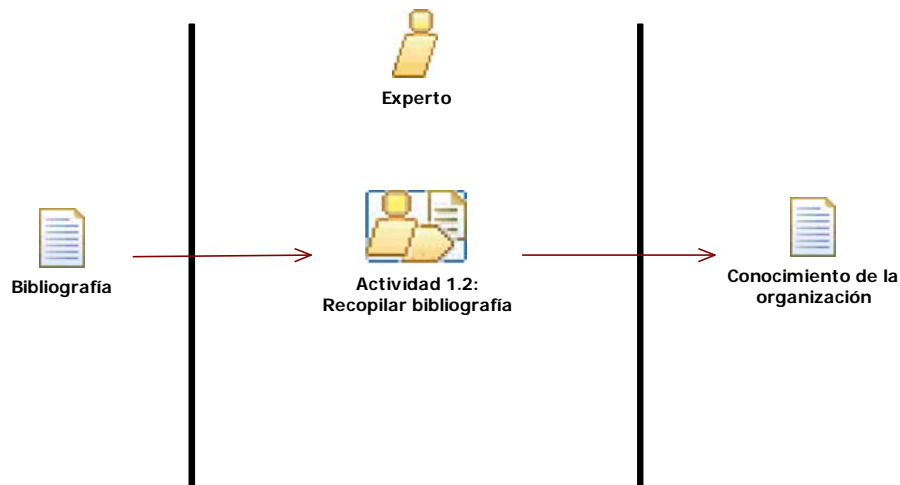


Figura 3.8. Actividad 1.2: Recopilar bibliografía.

### Entradas requeridas

Información obtenida de la bibliografía existente que guarde relación con los procesos de la organización.

### Roles participantes

Experto.

### Tareas a realizar

Recopilar referencias a libros, artículos, guías y otros elementos bibliográficos relacionados con los procesos que se llevan a cabo en la organización que permitan incrementar el conocimiento respecto a estos procesos y faciliten el aprendizaje de los mismos.

### Salidas esperadas

Conjunto de activos que representan las referencias bibliográficas relacionadas con los procesos de la organización.

### Actividad 1.3: Recopilar experiencias

#### Diagrama de la actividad

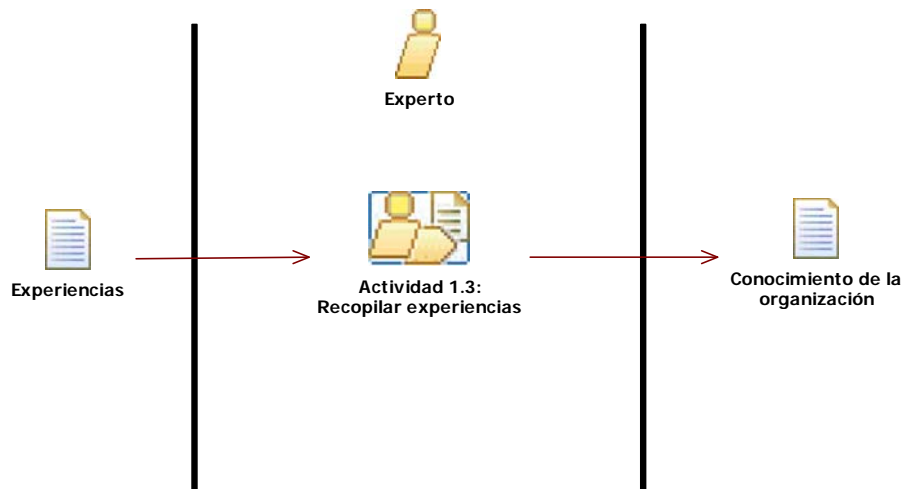


Figura 3.9. Actividad 1.3: Recopilar experiencias.

#### Entradas requeridas

Experiencias previas obtenidas en proyectos anteriores dentro de la organización.

#### Roles participantes

Experto.

#### Tareas a realizar

Recopilar experiencias obtenidas en proyectos realizados hasta la fecha en la organización con el objetivo de complementar el conocimiento explícito recopilado en relación con los procesos de la organización. Estas experiencias pueden ser recopiladas por medio de entrevistas individuales a personal experimentado o mediante grupos de enfoque donde los participantes debatan sobre la ejecución de distintos proyectos en los que estuvieron involucrados.

#### Salidas esperadas

Conjunto de activos que representan las experiencias obtenidas en proyectos previos de la organización.

## Proceso 2: Formalización del Conocimiento

El conocimiento de un dominio concreto puede representarse utilizando de distintos niveles de formalización, desde texto en lenguaje natural hasta reglas o modelos lógicos [Baumeister, 2011]. Cada uno de estos niveles de formalización tiene sus ventajas y sus inconvenientes. Así, el conocimiento en forma textual es muy habitual y puede ser elicitado de manera sencilla ya que no requiere conocimientos previos para su representación; sin embargo, la recuperación de conocimiento textual solo es posible mediante la comparación de cadenas de caracteres y, además, con los métodos actuales no resulta posible la automatización del razonamiento utilizando conocimiento textual. Por el contrario, las reglas lógicas son muy adecuadas para el razonamiento automatizado y, además, permiten procesar consultas semánticas; sin embargo, la obtención de reglas o modelos suele ser una tarea compleja y que requiere mucho tiempo.

Es por tanto muy importante seleccionar un nivel adecuado de formalización. Puesto que el conocimiento de una organización se encuentra habitualmente en forma textual (conocimiento explícito) o reside en los individuos (conocimiento tácito), en esta tesis doctoral se ha optado por una representación del conocimiento en forma textual estructurada mediante anotaciones semánticas, y complementada con algunos elementos multimedia (imágenes y vídeos). Para estructurar los datos se propone la utilización de plantillas y formularios de manera que se pueda introducir el conocimiento a un nivel básico utilizando la mayor cantidad posible de lenguaje natural.

### Objetivo del proceso

Una vez el conocimiento pre-existente ha sido identificado, este debe ser convertido en conocimiento explícito y luego ser almacenado en el repositorio de la organización. Este proceso de formalización del conocimiento se realiza mediante la utilización de un editor de conocimiento pre-existente que permite estructurar y estandarizar el conocimiento, definiendo así una guía de procesos que quedará registrada en el repositorio de conocimiento de la organización lista para ser distribuida posteriormente.

El alcance de este proceso se puede apreciar en la Figura 3.10.

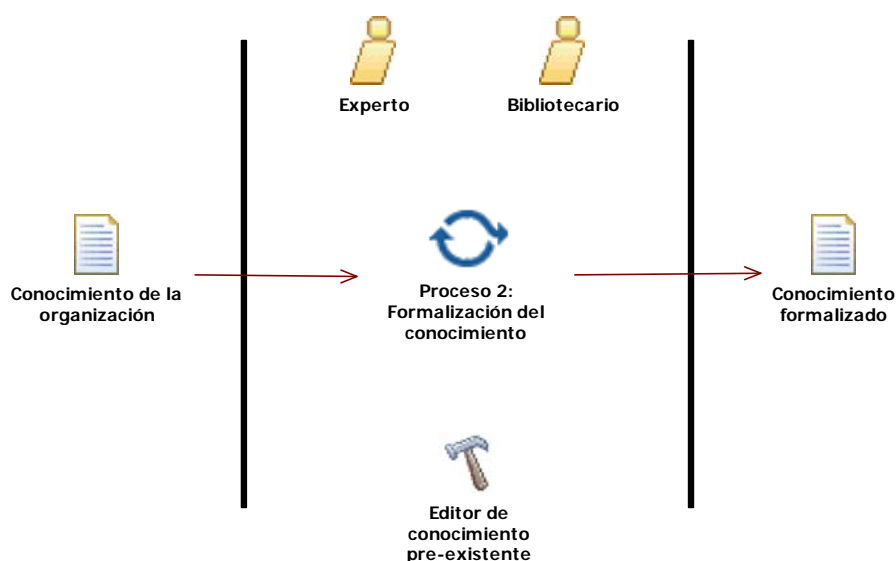


Figura 3.10. Proceso 2: Formalización del Conocimiento.



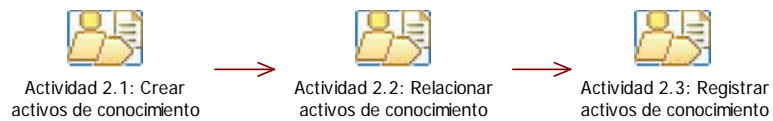
### **Entradas requeridas**

El proceso de formalización tomará como entrada el conocimiento de la organización identificado como conocimiento pre-existente.

### **Roles participantes**

El Experto es el responsable de la formalización del conocimiento inicial utilizando el editor de conocimiento pre-existente. El Bibliotecario es el responsable de registrar dicho conocimiento en el repositorio.

### **Actividades a realizar**



**Figura 3.11. Actividades del Proceso 2: Formalización del Conocimiento.**

### **Salidas esperadas**

La formalización del conocimiento pre-existente da lugar a la definición de una guía de procesos cuyos elementos quedan almacenados en el repositorio de conocimiento organizacional.

## Actividad 2.1: Crear activos de conocimiento

### Diagrama de la actividad



Figura 3.12. Actividad 2.1: Crear activos de conocimiento.

### Entradas requeridas

Conocimiento de la organización identificado como conocimiento pre-existente.

### Roles participantes

Experto.

### Tareas a realizar

Las tareas a realizar dentro de esta actividad se muestran en la Figura 3.13.

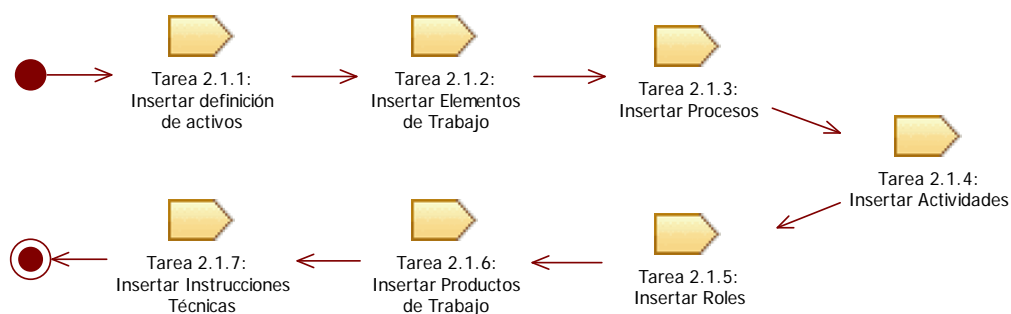


Figura 3.13. Tareas de la Actividad 2.1: Crear activos de conocimiento.

Utilizando el editor de conocimiento pre-existente, se deben insertar primero las definiciones de los activos que forman parte de conocimiento inicial de la organización, así como otras descripciones generales como pueden ser los principios y la cultura organizativa, el gobierno de la organización software, o el modelo de equipo a seguir en los proyectos. Posteriormente se deben insertar las descripciones, estados, transiciones entre estados y campos de los distintos elementos de trabajo. El siguiente paso será insertar las descripciones y los criterios de entrada y de salida de cada uno de los procesos de la organización. Después se insertan las descripciones y los criterios de entrada y de salida de cada una de las actividades que se llevan

a cabo dentro de los procesos de la organización. Más adelante se insertan las descripciones de los distintos roles involucrados en las actividades y procesos de la organización. A continuación se deben insertar los productos de trabajo necesarios para desarrollar las tareas durante la ejecución de los distintos procesos de la organización. Finalmente se introducen las instrucciones técnicas que explican cómo debe llevarse a cabo una actividad concreta.

### **Salidas esperadas**

Todos los contenidos de los activos de conocimiento han sido insertados por medio del editor de conocimiento pre-existente, quedando así creados los activos de conocimiento.

## Actividad 2.2: Relacionar activos de conocimiento

### Diagrama de la actividad



Figura 3.14. Actividad 2.2: Relacionar activos de conocimiento.

### Entradas requeridas

Activos de conocimiento creados mediante el editor de conocimiento pre-existente.

### Roles participantes

Experto.

### Tareas a realizar

Establecer vínculos entre los distintos activos de conocimiento creados previamente utilizando para ello el editor de conocimiento pre-existente. Dichos vínculos facilitarán el acceso desde el activo de conocimiento que se esté visualizando hacia otros activos de conocimiento que guarden relación con este. Así se deberá relacionar cada actividad con el proceso correspondiente dentro del cual se lleva a cabo, cada rol con las actividades y procesos en los cuales participe, cada producto de trabajo con las actividades y procesos en los que esté involucrado, etc.

### Salidas esperadas

Activos de conocimiento relacionados entre sí mediante el editor de conocimiento pre-existente.

## Actividad 2.3: Registrar activos de conocimiento

### Diagrama de la actividad

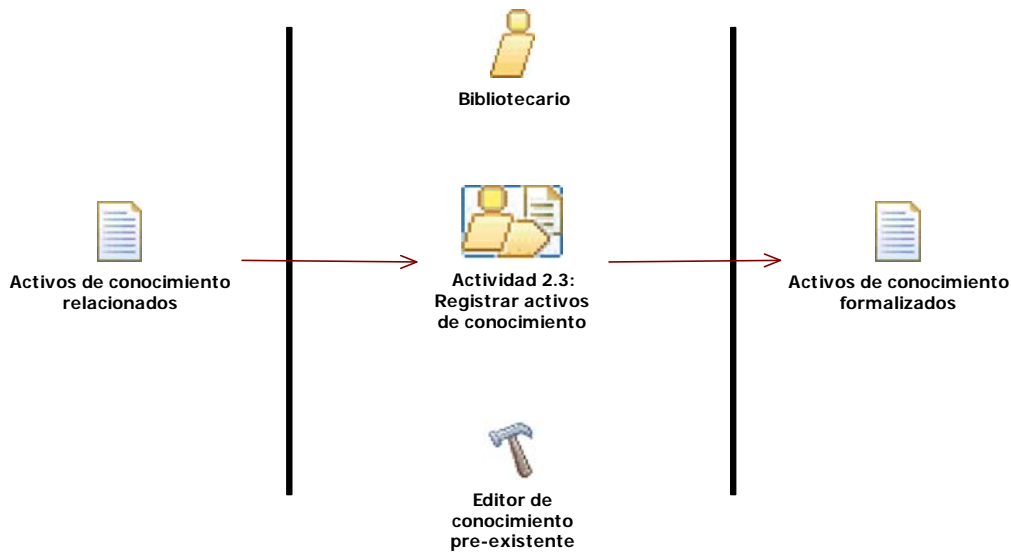


Figura 3.15. Actividad 2.3: Registrar activos de conocimiento.

### Entradas requeridas

Activos de conocimiento relacionados mediante el editor de conocimiento pre-existente.

### Roles participantes

Bibliotecario.

### Tareas a realizar

Los activos de conocimiento que fueron previamente creados y relacionados entre sí deben pasar a ser almacenados en el repositorio de conocimiento organizacional. Dichos activos quedan de esta manera formalizados y pasan a ser parte del conocimiento de la organización, estando desde este momento disponibles para su distribución entre los equipos de proyecto dentro de la organización.

### Salidas esperadas

Activos de conocimiento formalizados y almacenados en el repositorio de conocimiento organizacional en forma de guía de procesos.

### Proceso 3: Distribución del Conocimiento

Durante la realización de un proyecto, durante una jornada de aprendizaje o simplemente con el objetivo de estar al día, los equipos de trabajo de una organización pueden necesitar acceder a cualquier tipo de conocimiento acerca de los distintos procesos de la organización. Para satisfacer esta necesidad de acceso al conocimiento de la organización, dicho conocimiento debe ser previamente distribuido a quien lo necesite.

#### Objetivo del proceso

El objetivo de la distribución del conocimiento de la organización es conectar el conocimiento almacenado en el repositorio con los usuarios potenciales. Este proceso permite el despliegue de una vista que facilita el acceso a la versión del conocimiento organizacional distribuida.

Puesto que no todos los equipos de trabajo tienen las mismas necesidades, se debe ofrecer la posibilidad de desplegar distintos tipos de vistas adaptadas a cada necesidad concreta. Por ello, se proponen tres tipos de vistas:

- **Vista estática offline:** ofrece acceso a una versión del conocimiento organizacional sin necesidad de disponer de conexión a Internet; no incluye funcionalidades para recoger nuevo conocimiento tácito para el enriquecimiento del conocimiento organizacional.
- **Vista estática online:** ofrece acceso a una versión del conocimiento organizacional a través de Internet; no incluye funcionalidades para recoger nuevo conocimiento tácito para el enriquecimiento del conocimiento organizacional.
- **Vista dinámica online:** ofrece acceso a una versión del conocimiento organizacional a través de Internet; incluye funcionalidades para recoger nuevo conocimiento tácito para el enriquecimiento del conocimiento organizacional.

El alcance de este proceso se puede apreciar en la Figura 3.16.

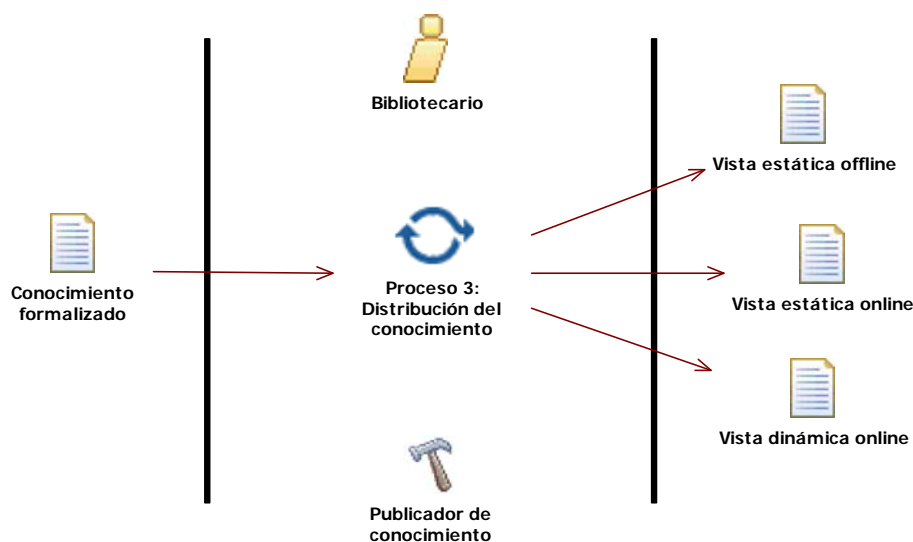


Figura 3.16. Proceso 3: Distribución del Conocimiento.

### **Entradas requeridas**

Como entrada de este proceso se utiliza el conocimiento que fue previamente formalizado y registrado en el repositorio organizacional.

### **Roles participantes**

El Bibliotecario es el encargado de distribuir a las diferentes vistas una versión de todo o parte del conocimiento de la organización almacenado en el repositorio en un instante concreto.

### **Actividades a realizar**



**Figura 3.17. Actividades del Proceso 3: Distribución del Conocimiento.**

### **Salidas esperadas**

El publicador de conocimiento permite la distribución de una versión de aquella parte del conocimiento organizacional que haya sido previamente seleccionado por resultar de utilidad para el equipo de trabajo durante la ejecución del proyecto. Dicho conocimiento es distribuido mediante la generación de una vista en el formato más conveniente para el equipo de trabajo al que vaya destinada: vista estática offline, vista estática online o vista dinámica online.

### Actividad 3.1: Seleccionar activos

#### Diagrama de la actividad

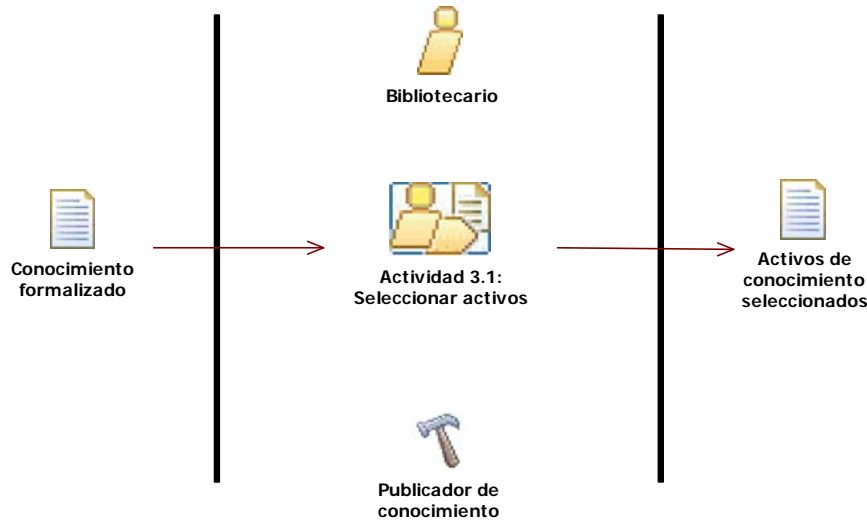


Figura 3.18. Actividad 3.1: Seleccionar activos.

#### Entradas requeridas

Conocimiento formalizado y registrado en el repositorio organizacional.

#### Roles participantes

Bibliotecario.

#### Tareas a realizar

Seleccionar los activos que se pretenden distribuir entre los equipos de trabajo a partir de todos los activos registrados en el repositorio de conocimiento organizacional. Los activos de conocimiento se seleccionarán en función de la metodología de desarrollo que se utilice en el proyecto y de las tareas que deban realizar los miembros del equipo de trabajo. Todo activo de conocimiento que pueda resultar de utilidad en el ámbito de dicho proyecto es susceptible de ser seleccionado.

#### Salidas esperadas

Conjunto de activos de conocimiento seleccionados para su distribución.



## Actividad 3.2: Integrar activos

### Diagrama de la actividad

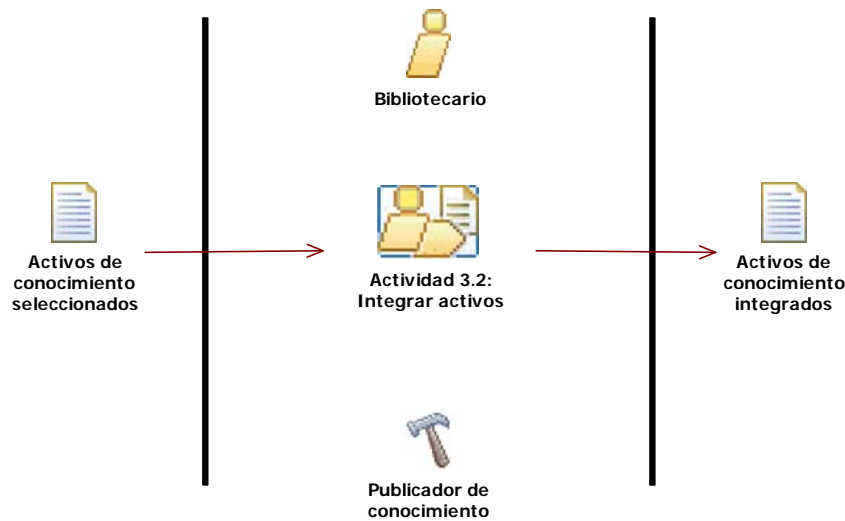


Figura 3.19. Actividad 3.2: Integrar activos.

### Entradas requeridas

Conjunto de activos de conocimiento seleccionados para su distribución.

### Roles participantes

Bibliotecario.

### Tareas a realizar

Se partirá de un esqueleto básico que definirá la estructura jerarquizada en la que se ensamblarán los elementos de conocimiento que formarán parte de la vista. Los activos de conocimiento seleccionados para formar parte de la vista se irán integrando por partes sobre dicho esqueleto. Igualmente, partiendo de las relaciones (unidireccionales) que se hubieran creado entre distintos activos durante la Actividad 2.2: Relacionar activos de conocimiento, se añadirán referencias cruzadas (bidireccionales) de manera que, cuando un activo esté vinculado con otro activo, se añada un vínculo que enlace el segundo con el primero. Así se simplifica la Actividad 2.2: Relacionar activos de conocimiento ya que solo requerirá introducir en los formularios un sentido de la relación entre dos activos, encargándose la herramienta de publicación de introducir automáticamente la relación en sentido contrario.

### Salidas esperadas

Activos de conocimiento seleccionados para su distribución e integrados en un único paquete de conocimiento.

### Actividad 3.3: Generar vistas

#### Diagrama de la actividad

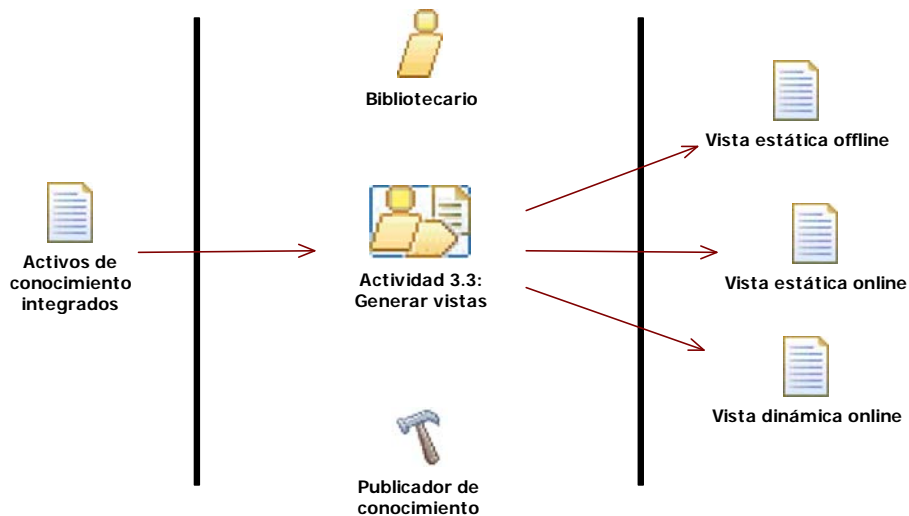


Figura 3.20. Actividad 3.3: Generar vistas.

#### Entradas requeridas

Activos de conocimiento seleccionados para su distribución e integrados en un único paquete de conocimiento.

#### Roles participantes

Bibliotecario.

#### Tareas a realizar

Crear y publicar una vista que permita el acceso a una versión del conocimiento de la organización generada a partir de los activos de conocimiento previamente seleccionados e integrados. Distintos mecanismos de generación permitirán obtener distintos tipos de vista en función de las necesidades del equipo de trabajo que la vaya a utilizar. Así, los contenidos de los distintos activos seleccionados serán convertidos a un formato estático disponible offline, a un formato estático disponible online, o a un formato modificable disponible online.

#### Salidas esperadas

Vista que contiene una versión de aquella parte del conocimiento organizacional que resulte útil para el equipo de trabajo durante la ejecución del proyecto. El tipo de vista podrá ser: vista estática offline, vista estática online o vista dinámica online.

## Proceso 4: Aplicación del Conocimiento

El conocimiento almacenado acerca de los procesos de la organización debe ser realmente conocido por los equipos de trabajo y aplicado efectivamente durante la realización de proyectos siempre que este sea útil.

Con la distribución del conocimiento y su posterior aplicación se completa la fase de internalización definida por Nonaka en el modelo SECI [Nonaka, 2006], de forma que los equipos de trabajo adquieren nuevo conocimiento tácito a partir del conocimiento explícito almacenado en el repositorio de la organización.

### Objetivo del proceso

Una vez que el conocimiento ha sido distribuido a través de las vistas, este podrá ser accedido y aplicado por las personas que participen en los procesos de futuros proyectos. Un activo puede ser consultado navegando a través de la vista, o puede ser adaptado a otro proyecto creando así una instancia de ese activo (estas instancias suelen ser documentos de texto, hojas de cálculo, tablas, figuras, etc. que son generadas con herramientas externas).

El alcance de este proceso se puede apreciar en la Figura 3.21.

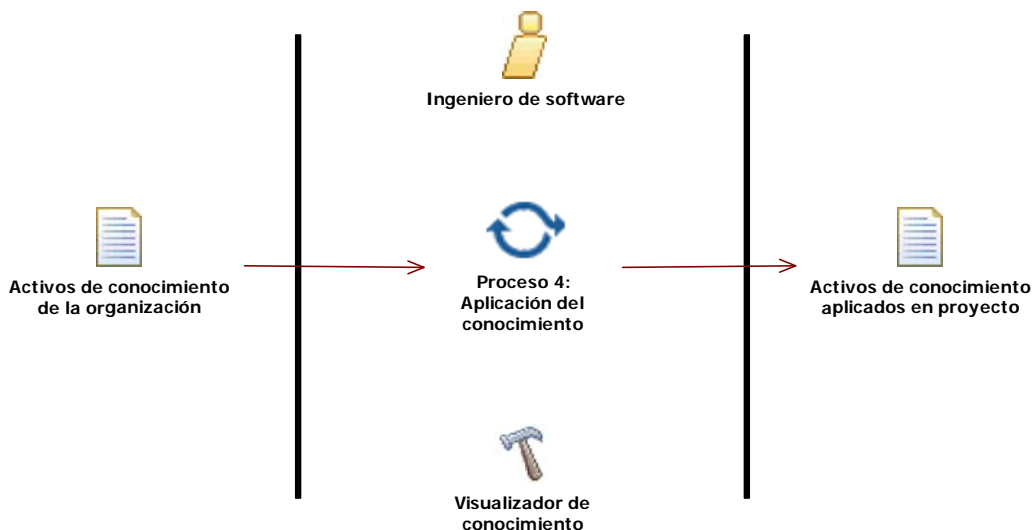


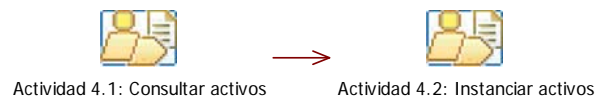
Figura 3.21. Proceso 4: Aplicación del Conocimiento.

### Entradas requeridas

Los activos de conocimiento distribuidos en la organización están disponibles para los equipos de trabajo.

### Roles participantes

Los Ingenieros de Software aplican el conocimiento que fue distribuido en la organización.

**Actividades a realizar**

**Figura 3.22. Actividades del Proceso 4: Aplicación del Conocimiento.**

**Salidas esperadas**

Los activos de conocimiento distribuidos han sido aplicados en la ejecución de algún proyecto de la organización.

## Actividad 4.1: Consultar activos

### Diagrama de la actividad

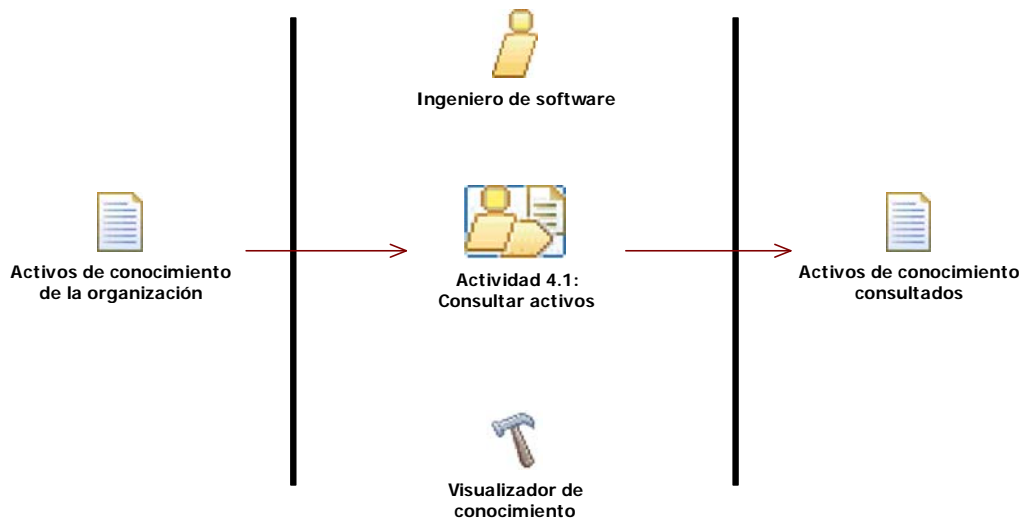


Figura 3.23. Actividad 4.1: Consultar activos.

### Entradas requeridas

Activos de conocimiento de la organización previamente distribuidos mediante una vista.

### Roles participantes

Ingeniero de software.

### Tareas a realizar

Navegar por la vista a través de hiperenlaces para localizar el activo que se quiere consultar.

En las vistas existen dos formas de navegación: en la primera, el usuario selecciona un proceso o actividad a partir del cual accede, por ejemplo, a un elemento de trabajo o a un producto de trabajo relacionado; en la segunda, el usuario selecciona un rol para después acceder a un proceso o actividad en el cual participa dicho rol. El primer tipo de navegación se recomienda cuando se necesita información específica acerca de un activo, mientras que el segundo tipo de navegación se recomienda cuando se quiere conocer el siguiente proceso o actividad a realizar.

Adicionalmente, las vistas proporcionan una caja de búsqueda en la cual el usuario puede introducir palabras clave de manera que la vista presentará una lista de aquellos activos en los cuales figuren esas palabras clave.

### Salidas esperadas

La vista presenta el activo consultado utilizando texto con formato e iconos, incluyendo en ocasiones vídeos y documentos asociados.

## Actividad 4.2: Instanciar activos

### Diagrama de la actividad

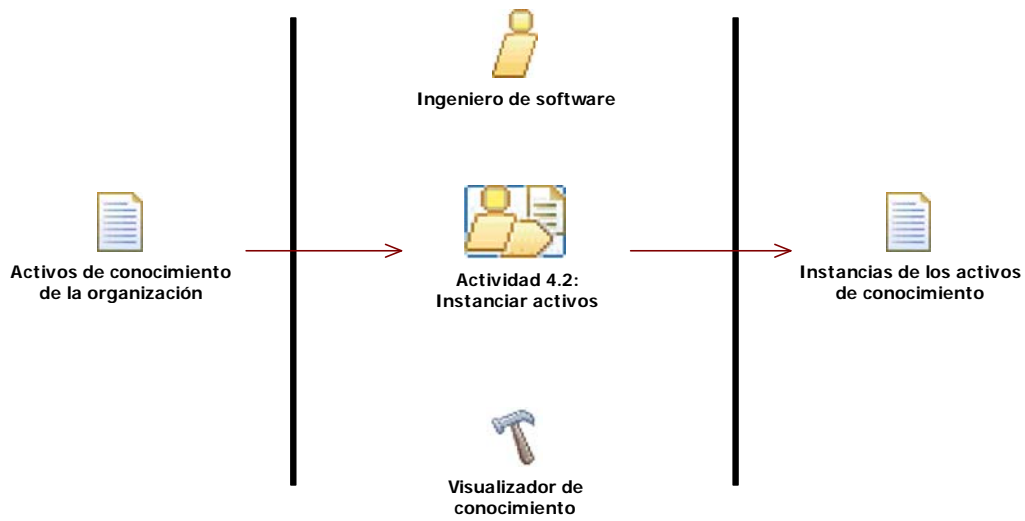


Figura 3.24. Actividad 4.2: Instanciar activos.

### Entradas requeridas

Activos de conocimiento de la organización previamente distribuidos mediante una vista.

### Roles participantes

Ingeniero de software.

### Tareas a realizar

Adaptar un activo de conocimiento de acuerdo con las características y necesidades de un proyecto, creando así una instancia de dicho activo.

### Salidas esperadas

Instancias de los activos de conocimiento.

## Proceso 5: Preservación del Conocimiento

La aplicación del conocimiento organizacional existente (explícito) puede dar lugar a la generación de nuevo conocimiento (tácito) proveniente de la experiencia diaria adquirida al llevar a cabo los procesos de la organización en distintos tipos de proyectos. Este nuevo conocimiento puede resultar relevante para la organización, por lo que debe ser capturado para su preservación.

Según [Borges, 2002], la interacción entre los proyectos y el conocimiento de la organización establece dos bucles que pueden dar lugar a esta generación de nuevo conocimiento tácito. El primero es a nivel de proyecto, cuando se obtiene nuevo conocimiento durante la realización de los procesos, el cual queda reflejado en este proceso. El segundo es a nivel de organización, representado mediante el Proceso 6: Gestión de Cambios.

Con la preservación del conocimiento se completan las fases de socialización y externalización definidas por Nonaka en el modelo SECI [Nonaka, 2006], de manera que los usuarios comparten sus experiencias adquiridas durante la realización de los proyectos, convirtiendo para ello ese conocimiento tácito en explícito.

### Objetivo del proceso

Recoger el nuevo conocimiento tácito generado al aplicar dichos activos de conocimiento en la realización de los distintos proyectos de la organización. Este nuevo conocimiento será recogido por medio del registrador de conocimiento tácito.

El alcance de este proceso se puede apreciar en la Figura 3.25.

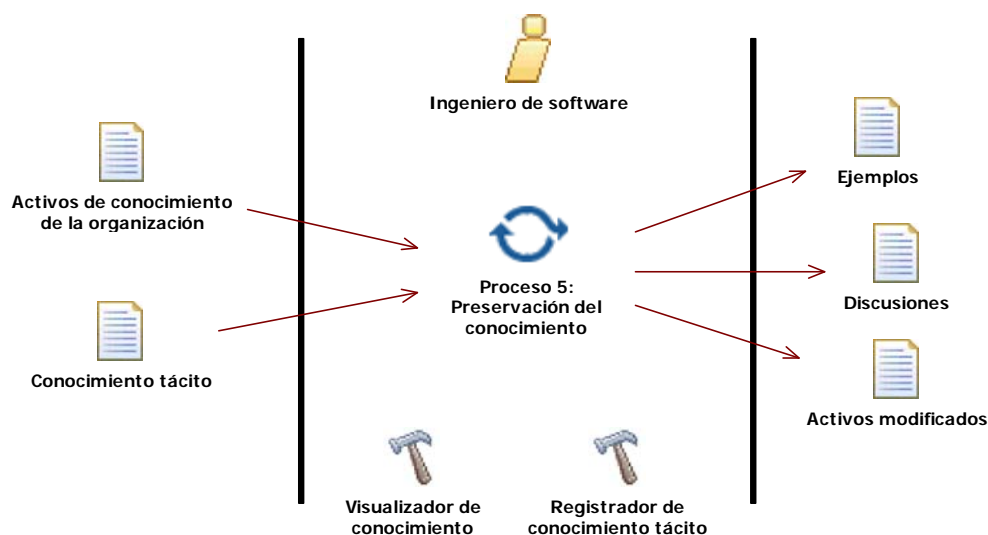


Figura 3.25. Proceso 5: Preservación del Conocimiento.

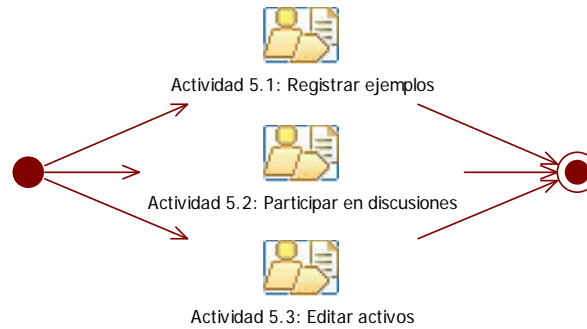
### Entradas requeridas

El elemento a preservar será el conocimiento tácito generado a partir de la aplicación del conocimiento almacenado, lo cual permitirá enriquecer los activos de conocimiento de la organización.

### **Roles participantes**

Los Ingenieros de Software pertenecientes a los distintos equipos de trabajo son los encargados de preservar el nuevo conocimiento tácito que se genera durante la realización de los proyectos.

### **Actividades a realizar**



**Figura 3.26. Actividades del Proceso 5: Preservación del Conocimiento.**

### **Salidas esperadas**

Contribuciones realizadas por los ingenieros de software a la vista disponible para el equipo de trabajo al cual pertenecen (vista dinámica online, principalmente). Estas contribuciones pueden ser en forma de ejemplos, de discusiones o cambios en los activos de conocimiento.



### Actividad 5.1: Registrar ejemplos

El nuevo conocimiento tácito generado durante la ejecución de los proyectos puede ser capturado en forma de ejemplos a partir de los artefactos producidos al llevar a cabo los procesos en la organización. Estos ejemplos pueden ser de utilidad para el equipo de proyecto o para otros equipos dentro de la organización ya que ofrecen posibles soluciones a problemas determinados. Por lo tanto, durante la ejecución de los procesos de la organización es conveniente registrar ejemplos que puedan ser de utilidad en un futuro. Para registrar un ejemplo bastará con introducir en el registrador de conocimiento tácito la información correspondiente al ejemplo, es decir, el contexto, el problema que aborda y la solución que se propone.

#### Diagrama de la actividad

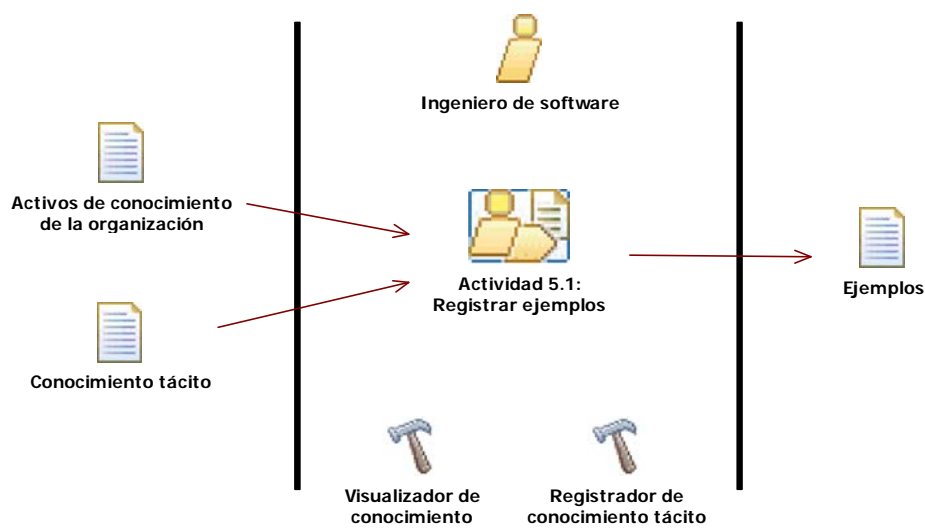


Figura 3.27. Actividad 5.1: Registrar ejemplos.

#### Entradas requeridas

Conocimiento tácito generado y activos de conocimiento de la organización.

#### Roles participantes

Ingeniero de Software.

#### Tareas a realizar

Las tareas a realizar dentro de esta actividad se muestran en la Figura 3.28.

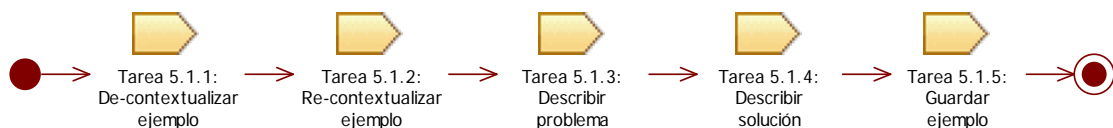


Figura 3.28. Tareas de la Actividad 5.1: Registrar ejemplos.

Para una reutilización más efectiva de los ejemplos, este tipo de contribución requiere primero una de-contextualización y luego una re-contextualización. En la de-contextualización, el usuario debe eliminar aquellos detalles del contexto que no sean relevantes para su

comprensión con el objetivo de aislar dicho ejemplo del proyecto del cual proviene y obtener así material limpio. Después, en la re-contextualización el usuario compone un nuevo contexto de manera que el ejemplo tenga suficiente significado como para poder ser reutilizado. El proceso de de-contextualización y re-contextualización de un ejemplo ayuda a la identificación de las condiciones bajo las cuales el ejemplo es aplicable sin depender del proyecto.

### **Salidas esperadas**

Contribuciones realizadas en forma de ejemplos. Todo ejemplo tiene un texto que describe el contexto al cual pertenece el ejemplo, una descripción del problema que resuelve el ejemplo, los artefactos asociados al problema, una descripción de la solución al problema, y los artefactos asociados a la solución.

### Actividad 5.2: Participar en discusiones

El nuevo conocimiento tácito generado durante la ejecución de los proyectos puede ser capturado en forma de discusiones a partir del intercambio de ideas y opiniones entre los miembros de un mismo equipo de trabajo al llevar a cabo los procesos en la organización. Estas discusiones pueden ser de utilidad para el equipo de proyecto o para otros equipos dentro de la organización ya que ofrecen distintas opiniones sobre los procesos o aportan nuevas ideas para su ejecución. Por lo tanto, durante la ejecución de los procesos de la organización es conveniente participar en discusiones que puedan ser de utilidad en un futuro. Las discusiones se pueden obtener de diferentes maneras dependiendo de la plataforma utilizada por la organización: discusiones por correo electrónico, comentarios en una wiki o mensajes en un foro, etc.

#### Diagrama de la actividad

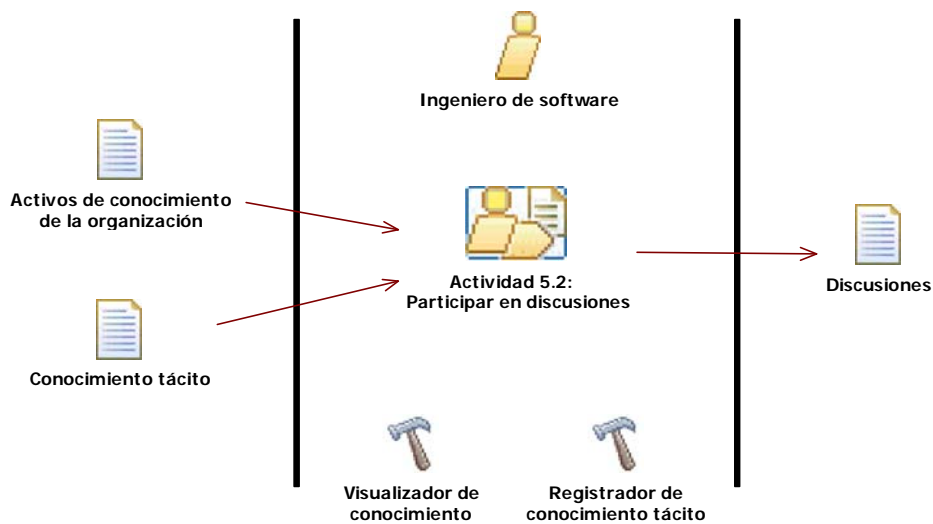


Figura 3.29. Actividad 5.2: Participar en discusiones.

#### Entradas requeridas

Conocimiento tácito generado y activos de conocimiento de la organización.

#### Roles participantes

Ingeniero de Software.

#### Tareas a realizar

Las tareas a realizar dentro de esta actividad se muestran en la Figura 3.30.

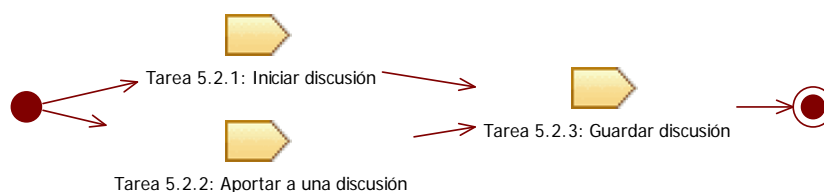


Figura 3.30. Tareas de la Actividad 5.2: Participar en discusiones.

Un ingeniero de software podrá empezar una nueva discusión relacionada con un activo de conocimiento o podrá realizar aportaciones a las discusiones ya comenzadas por él mismo o por alguno de los otros ingenieros de software. Después de iniciar una nueva discusión o de realizar una aportación a una discusión ya creada, la discusión debe ser guardada.

**Salidas esperadas**

Contribuciones realizadas en forma de discusiones. Toda discusión tiene un encabezado y su texto correspondiente.

### Actividad 5.3: Editar activos

El nuevo conocimiento tácito generado durante la ejecución de los proyectos puede ser capturado mediante la edición de los activos de conocimiento de la organización. La edición de activos puede ser de utilidad para mantener el conocimiento de la organización actualizado y libre de erratas. Por lo tanto, durante la ejecución de los procesos de la organización es conveniente editar aquellos activos que contengan información incorrecta. Para editar un activo bastará con modificar su contenido por medio del registrador de conocimiento tácito.

#### Diagrama de la actividad

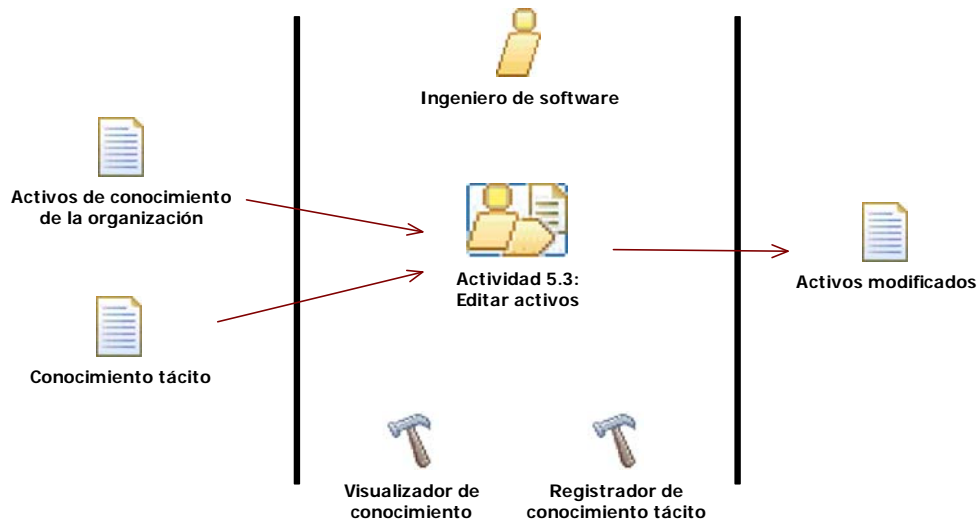


Figura 3.31. Actividad 5.3: Editar activos.

#### Entradas requeridas

Conocimiento tácito generado y activos de conocimiento de la organización.

#### Roles participantes

Ingeniero de Software.

#### Tareas a realizar

Las tareas a realizar dentro de esta actividad se muestran en la Figura 3.32.

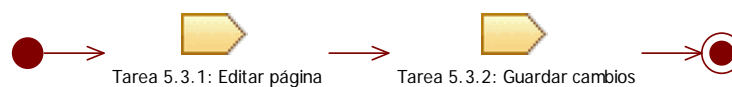


Figura 3.32. Tareas de la Actividad 5.3: Editar activos.

Los activos de conocimiento de la organización también pueden ser modificados para ser corregidos o ampliados.

#### Salidas esperadas

Contribuciones realizadas en forma de modificaciones de los activos de conocimiento.

### Proceso 6: Gestión de Cambios

Anteriormente se mencionó que la interacción entre los proyectos y el conocimiento de la organización establece dos bucles que pueden dar lugar a esta generación de nuevo conocimiento tácito [Borges, 2002]. El primero es a nivel de proyecto, representado mediante el Proceso 5: Preservación del Conocimiento. El segundo es a nivel de organización, cuando el nuevo conocimiento obtenido en los distintos proyectos es recogido en el repositorio de la organización y puesto a disposición de todos para su reutilización en otros proyectos, lo cual queda reflejado en este proceso.

Con la gestión de los cambios del conocimiento en el repositorio de la organización se completa la fase de combinación definida por Nonaka en el modelo SECI [Nonaka, 2006], de forma que el nuevo conocimiento que se hace explícito durante la externalización pasa ahora a unirse al conocimiento ya existente en el repositorio de conocimiento de la organización. Así, a partir de este conocimiento mejorado se puede ofrecer una nueva versión de las distintas vistas, de manera que el nuevo conocimiento mejorado estaría ahora disponible a todos los equipos de trabajo independientemente del quién aportó ese nuevo conocimiento.

#### Objetivo del proceso

Mejorar el conocimiento almacenado en el repositorio de conocimiento de la organización mediante la inclusión de las contribuciones realizadas por los ingenieros de software durante la realización de los distintos proyectos. Para ello se propone un mecanismo de realimentación que permite la gestión de los cambios en el conocimiento de la organización utilizando dos alternativas: con revisión de cambios (supervisada) o sin revisión de cambios (no supervisada).

El alcance de este proceso se puede apreciar en la Figura 3.33.

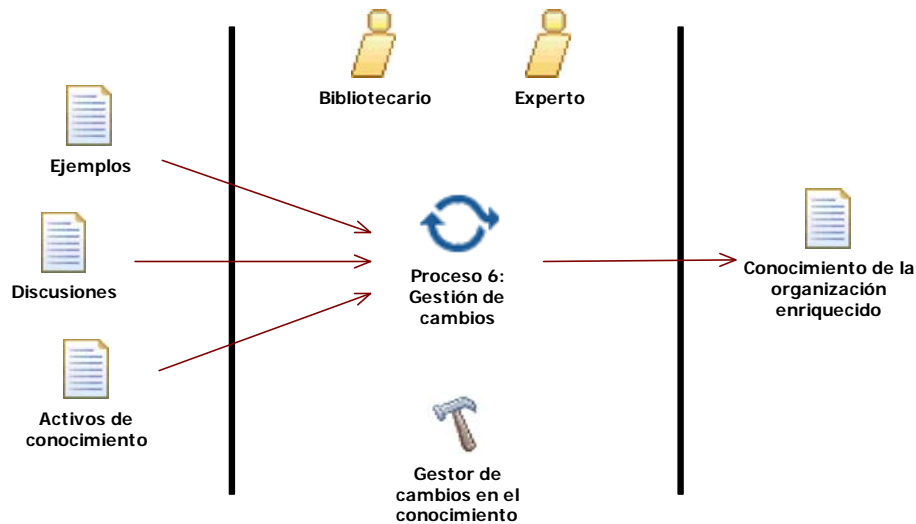


Figura 3.33. Proceso 6: Gestión de Cambios.

**Entradas requeridas**

Conocimiento actualmente proporcionado por la vista cuyos cambios se deseen gestionar. Parte de dicho conocimiento corresponderá al conocimiento tácito que haya sido preservado, es decir, a aquellas contribuciones realizadas por los ingenieros de software en forma de ejemplos, discusiones o cambios en los activos de conocimiento.

**Roles participantes**

El Experto es el encargado de gestionar los cambios en el conocimiento de la organización cuando el conocimiento tácito es realimentado. El Bibliotecario es responsable de registrar en el repositorio los cambios en el conocimiento debidos a la realimentación.

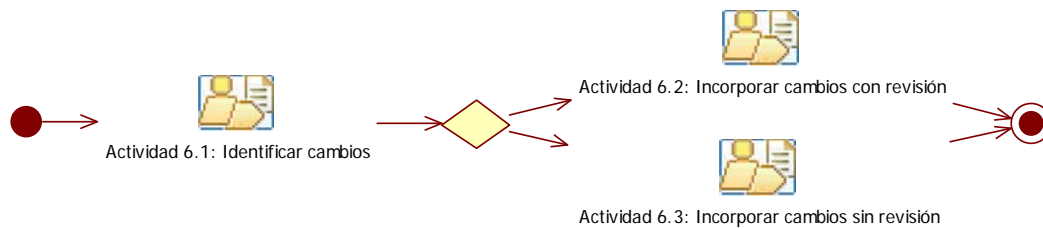
**Actividades a realizar**

Figura 3.34. Actividades del Proceso 6: Gestión de Cambios.

**Salidas esperadas**

El conocimiento de la organización queda enriquecido al añadir al repositorio organizacional el nuevo conocimiento tácito proveniente de las distintas contribuciones realizadas durante la ejecución del proyecto.

## Actividad 6.1: Identificar cambios

### Diagrama de la actividad

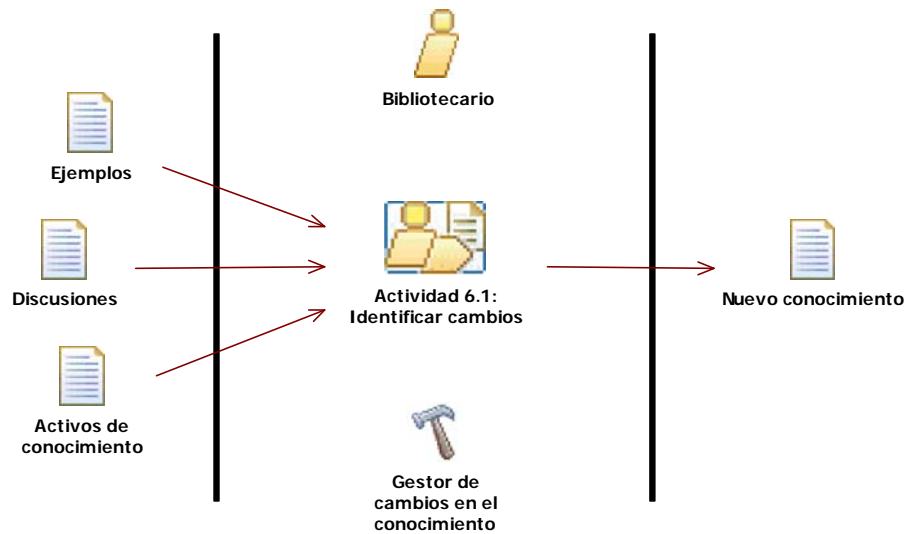


Figura 3.35. Actividad 6.1: Identificar cambios.

### Entradas requeridas

Conocimiento actualmente proporcionado por la vista cuyos cambios se deseen gestionar.

### Roles participantes

Bibliotecario.

### Tareas a realizar

Las tareas a realizar dentro de esta actividad se muestran en la Figura 3.36.

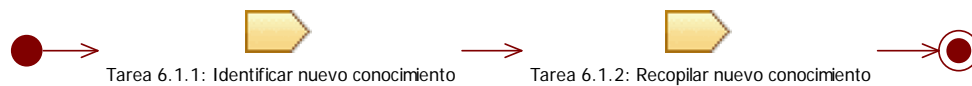


Figura 3.36. Tareas de la Actividad 6.1: Identificar cambios.

Primero se deben identificar cuáles son los activos de conocimiento que han sido modificados o que tienen asociadas nuevas contribuciones en forma de ejemplos o discusiones. Para facilitar la ejecución de esta tarea se puede configurar el registrador de conocimiento tácito de manera que envíe automáticamente al bibliotecario una notificación (un correo electrónico, por ejemplo) cada vez que un activo sea editado, cada vez que alguien inicie o participe en una discusión, o cada vez que se registre un nuevo ejemplo, es decir, cuando se realice alguna de las actividades del Proceso 5: Preservación del Conocimiento.

Una vez identificados los activos que fueron modificados, se debe recopilar todo el nuevo conocimiento preservado en dichos activos, quedando disponibles estos cambios en los activos para su incorporación al repositorio de conocimiento organizacional.

### Salidas esperadas

Nuevo conocimiento tácito identificado y recopilado.



### Actividad 6.2: Incorporar cambios con revisión

#### Diagrama de la actividad

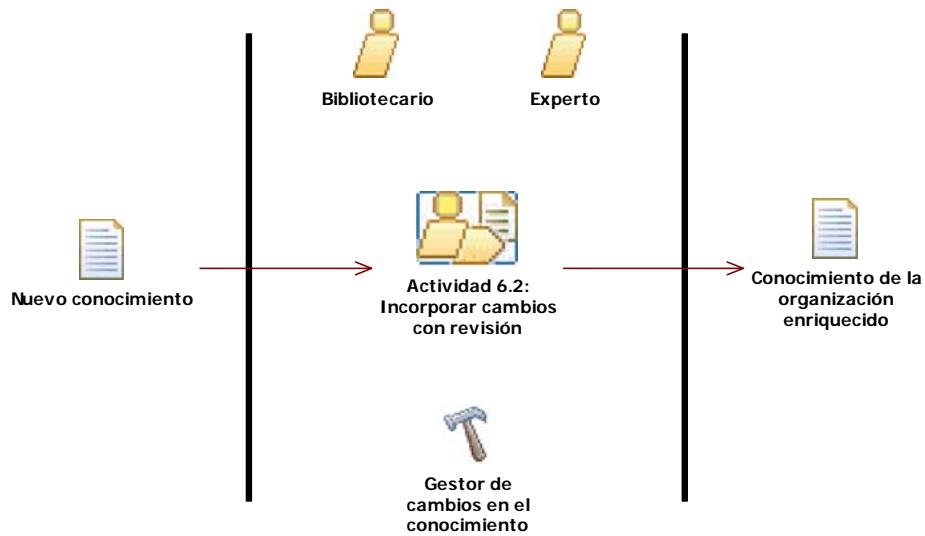


Figura 3.37. Actividad 6.2: Incorporar cambios con revisión.

#### Entradas requeridas

Nuevo conocimiento tácito identificado y recopilado.

#### Roles participantes

Bibliotecario y Experto.

#### Tareas a realizar

Las tareas a realizar dentro de esta actividad se muestran en la Figura 3.38.

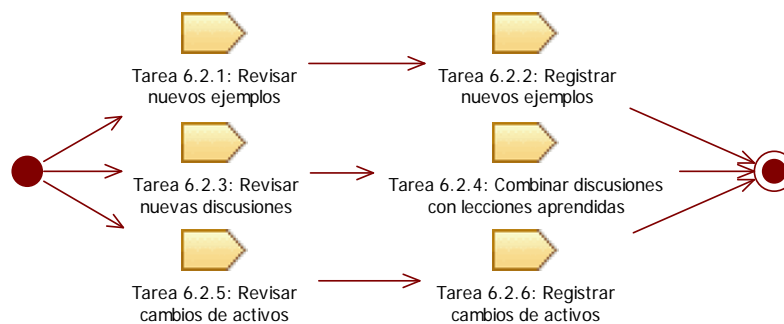


Figura 3.38. Tareas de la Actividad 6.2: Incorporar cambios con revisión.

Cada contribución es supervisada por un experto, el cual puede decidir descartarla, editarla o aceptarla. Si la contribución no es descartada, esta es incorporada al repositorio de conocimiento de la organización.

Las tareas implicadas en esta actividad dependen del tipo de contribución:

- En el caso de los nuevos ejemplos, estos son revisados por el experto que decidirá cuáles pueden ser útiles y relevantes para la organización y cuáles no. Además, el experto deberá revisar que su de-contextualización y re-contextualización se hayan hecho de tal manera que se puedan identificar fácilmente las condiciones bajo las cuales cada ejemplo es aplicable sin depender del proyecto. Por último, también se debe revisar que cada ejemplo está asociado al activo adecuado, ya que en caso contrario el experto deberá reubicarlos. Una vez superada la revisión, los ejemplos aceptados se registran en el repositorio de conocimiento organizacional siendo adjuntados a la lista de ejemplos del activo con el cual estén relacionados.
- En cuanto a las discusiones, estas son revisadas por el experto de manera que pueden ser modificadas para facilitar su comprensión. Además, las discusiones referentes a un mismo tema pueden ser unificadas para que este conocimiento sea más fácilmente localizable. Por último, también se debe revisar que las discusiones están asociadas al activo adecuado, ya que en caso contrario el experto deberá reubicarlas. Una vez revisadas, las discusiones aceptadas pasan a ser lecciones aprendidas y son combinadas con el resto de lecciones aprendidas del activo con el cual estén relacionadas, quedando todas almacenadas en el repositorio organizacional.
- Finalmente, los cambios en los activos de conocimiento son revisados por el experto que deberá comprobar si dichas modificaciones permiten el enriquecimiento del conocimiento organizacional. Los cambios que superan la revisión son incorporados al activo correspondiente dentro del repositorio de conocimiento organizacional.

### **Salidas esperadas**

Conocimiento de la organización enriquecido con el conocimiento tácito generado y preservado durante la ejecución del proyecto.

### Actividad 6.3: Incorporar cambios sin revisión

#### Diagrama de la actividad

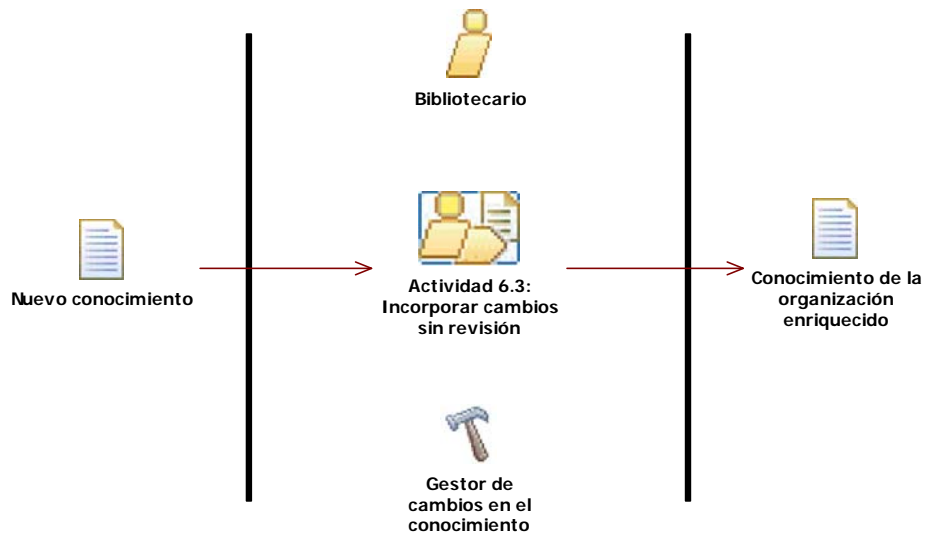


Figura 3.39. Actividad 6.2: Incorporar cambios sin revisión.

#### Entradas requeridas

Nuevo conocimiento tácito identificado y recopilado.

#### Roles participantes

Bibliotecario.

#### Tareas a realizar

Las tareas a realizar dentro de esta actividad se muestran en la Figura 3.40.



Figura 3.40. Tareas de la Actividad 6.2: Incorporar cambios sin revisión.

Todas las contribuciones son directamente aceptadas sin necesidad de que un experto revise los cambios, de manera que todas las contribuciones siempre pasan a formar parte del repositorio de conocimiento de la organización.

Las tareas implicadas en esta actividad dependen del tipo de contribución:

- En el caso de los nuevos ejemplos, esto se registran en el repositorio de conocimiento organizacional siendo adjuntados a la lista de ejemplos del activo con el cual estén relacionados.
- En cuanto a las discusiones, estas pasan a ser lecciones aprendidas y son combinadas con el resto de lecciones aprendidas del activo con el cual estén relacionadas, quedando todas almacenadas en el repositorio organizacional.
- Finalmente, los cambios en los activos de conocimiento son incorporados al activo correspondiente dentro del repositorio de conocimiento organizacional.

**Salidas esperadas**

Conocimiento de la organización enriquecido con el conocimiento tácito generado y preservado durante la ejecución del proyecto.

## Proceso 7: Medición del Conocimiento

García et al. [García, 2011] sugieren que el uso de los activos almacenados debe ser monitorizado para evaluar si el conocimiento de la organización es gestionado de manera adecuada y eficiente. Además, el éxito en el aprendizaje de los procesos utilizando ITAKA dependerá en gran medida de la satisfacción de los usuarios y de otros factores que puedan incrementar las intenciones de los usuarios respecto a la continuidad en el uso de ITAKA [Chiu, 2007], como por ejemplo la calidad del conocimiento almacenado.

Para poder medir y evaluar todos estos factores será necesario establecer previamente unos objetivos de medición, plantear un conjunto de cuestiones relacionadas con los objetivos establecidos, y especificar un conjunto de métricas (indicadores) para tratar de responder de una forma medible a las cuestiones planteadas.

### Objetivo del proceso

Analizar el uso de ITAKA a partir de las medidas tomadas con el objetivo de identificar oportunidades de mejora mediante la introducción de cambios correctores en el futuro. Los datos recogidos serán analizados para establecer qué aspectos de ITAKA requieren ajustes.

El alcance de este proceso se puede apreciar en la Figura 3.41.

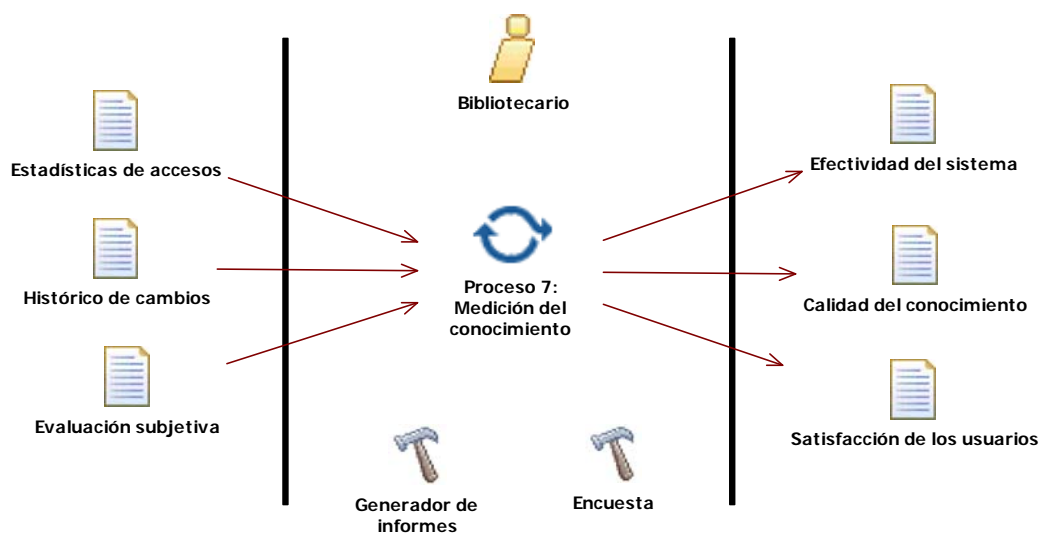


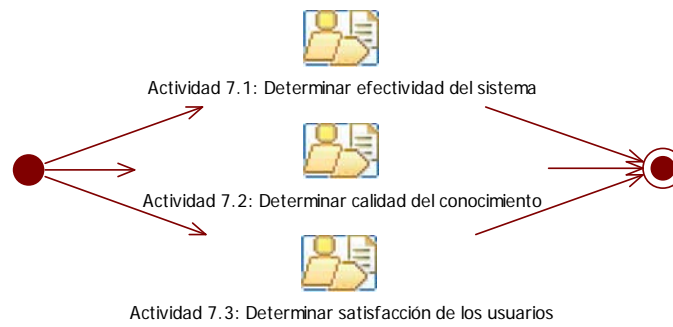
Figura 3.41. Proceso 7: Medición del Conocimiento.

### Entradas requeridas

Se deben obtener estadísticas acerca del comportamiento de los usuarios, medidas relacionadas con las nuevas contribuciones al repositorio de conocimiento de la organización, y datos acerca de la evaluación subjetiva de la utilidad y de la calidad del nuevo conocimiento y de los productos obtenidos en los proyectos realizados.

### Roles participantes

El Bibliotecario es el responsable de monitorizar el sistema para evaluar si el conocimiento de la organización es gestionado de manera adecuada y eficiente.

**Actividades a realizar**

**Figura 3.42. Actividades del Proceso 7: Medición del Conocimiento.**

**Salidas esperadas**

Como resultado se obtienen estadísticas que permiten analizar los niveles de efectividad del sistema en la captura de nuevo conocimiento, de calidad del conocimiento almacenado y de satisfacción de los usuarios.

## Actividad 7.1: Determinar efectividad del sistema

### Diagrama de la actividad

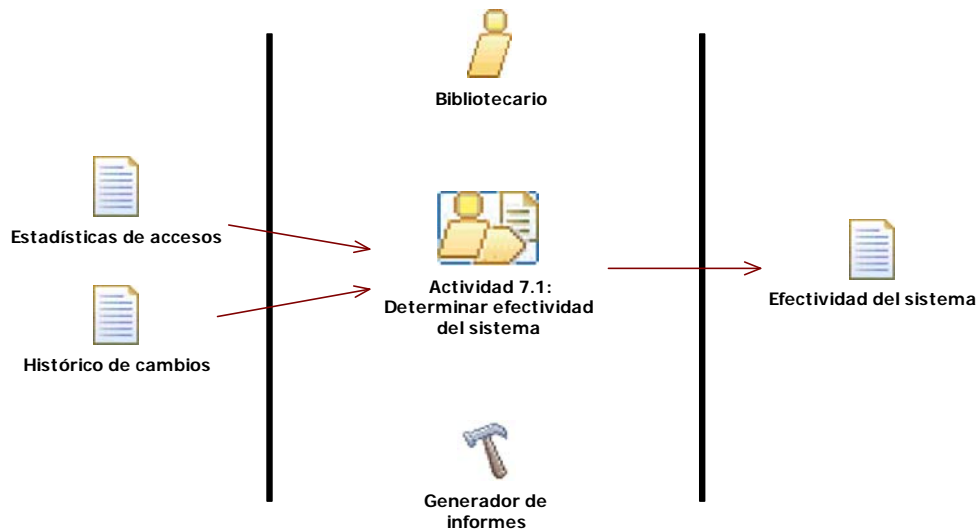


Figura 3.43. Actividad 7.1: Determinar efectividad del sistema.

### Entradas requeridas

Estadísticas acerca del número de usuarios que acceden al sistema, número de activos a los que se ha accedido, tiempo que los usuarios pasan conectados al sistema, número de cambios en el conocimiento almacenado, participación en discusiones, número de lecciones aprendidas, publicación de ejemplos, y número de activos mejorados en consecuencia.

### Roles participantes

Bibliotecario.

### Tareas a realizar

Recopilar y analizar los datos recogidos para determinar si los mecanismos de adquisición de conocimiento son efectivos.

### Salidas esperadas

Nivel de efectividad del sistema en la captura de nuevo conocimiento.

## Actividad 7.2: Determinar calidad del conocimiento

### Diagrama de la actividad



Figura 3.44. Actividad 7.2: Determinar calidad del conocimiento.

### Entradas requeridas

Estadísticas acerca del número de cambios en el conocimiento almacenado, participación en discusiones, número de lecciones aprendidas, publicación de ejemplos, número de activos mejorados en consecuencia, calidad de las contribuciones, número de contribuciones aceptadas y descartadas, y evaluación subjetiva de la utilidad y de la calidad del nuevo conocimiento y de los productos obtenidos en los proyectos realizados.

### Roles participantes

Bibliotecario.

### Tareas a realizar

Recopilar y analizar los datos recogidos para determinar cuál es la calidad del nuevo conocimiento elicitado.

### Salidas esperadas

Nivel de calidad del conocimiento almacenado.



## Actividad 7.3: Determinar satisfacción de los usuarios

### Diagrama de la actividad

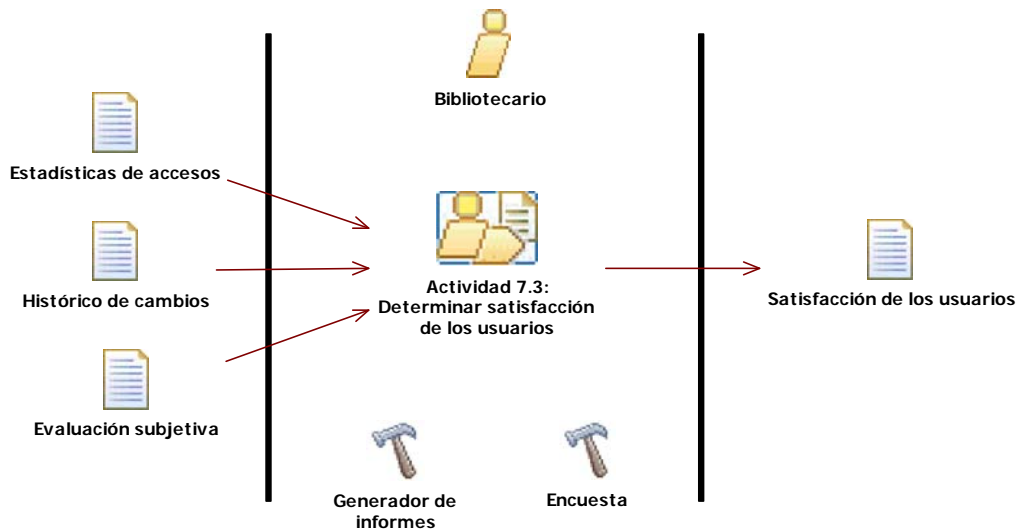


Figura 3.45. Actividad 7.3: Determinar satisfacción de los usuarios.

### Entradas requeridas

Estadísticas acerca del número de cambios en el conocimiento almacenado, participación en discusiones, número de lecciones aprendidas, publicación de ejemplos, número de activos mejorados en consecuencia, calidad de las contribuciones, número de contribuciones aceptadas y descartadas, y evaluación subjetiva de la utilidad y de la calidad del nuevo conocimiento y de los productos obtenidos en los proyectos realizados.

### Roles participantes

Bibliotecario.

### Tareas a realizar

Recopilar y analizar los datos recogidos para determinar si las funcionalidades ofrecidas por el sistema satisfacen las necesidades de los usuarios.

### Salidas esperadas

Nivel de satisfacción de los usuarios.



# Capítulo 4: Implementación de la Solución Propuesta

4.1. Arquitectura del sistema.....	115
4.2. Componentes para la persistencia del conocimiento organizacional .....	116
4.3. Componentes para la formalización del conocimiento pre-existente.....	118
4.4. Componentes para la distribución del conocimiento organizacional .....	118
4.4.1. Integrador .....	119
4.4.2. Generador.....	121
4.4.3. Conversor.....	122
4.5. Componentes para la aplicación y preservación de conocimiento .....	123
4.6. Componentes para la gestión de cambios en el conocimiento organizacional.....	126
4.6.1. Editor de cambios .....	126
4.6.2. Editor de lecciones aprendidas.....	127
4.6.3. Extractor de ejemplos.....	129
4.7. Componentes para la medición de conocimiento .....	129



El objetivo de este capítulo es mostrar los distintos componentes que han sido utilizados en la presente tesis doctoral para realizar una implementación de ITAKA que permite llevar a cabo la validación de la misma.

### 4.1. Arquitectura del sistema

A continuación se muestra un esquema de tres capas que representa la arquitectura del marco de trabajo propuesto en esta tesis doctoral y el intercambio de conocimiento que se produce entre los distintos módulos involucrados. El conocimiento fluye a través de ITAKA alternando entre sus formas tácita y explícita, y viceversa, siguiendo el ciclo continuo de procesos presentado en el capítulo anterior (ver Figura 3.4). Como se puede observar en la Figura 4.1, el proceso Identificación del Conocimiento no está contemplado en esta arquitectura puesto que, según la definición realizada en el capítulo anterior, su ejecución no requiere la utilización de ninguna herramienta software.

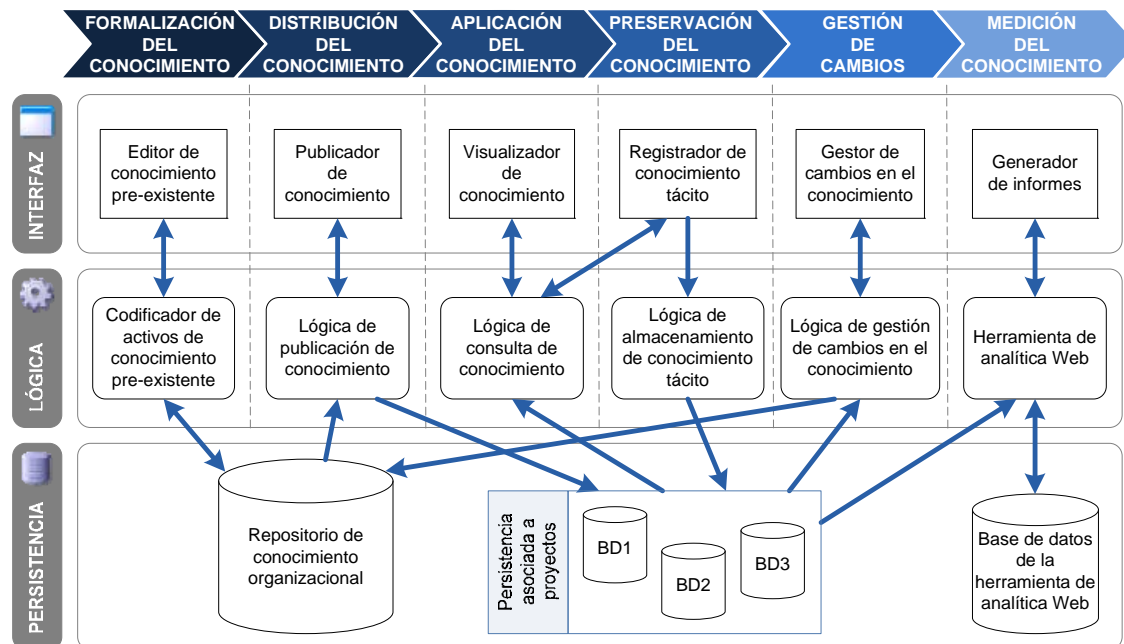


Figura 4.1. Arquitectura de ITAKA.

En primer lugar, el *Editor de conocimiento pre-existente* permite realizar el proceso Formalización del Conocimiento, estructurando y estandarizando el conocimiento por medio de la definición de los distintos activos de conocimiento de la organización y de las relaciones existentes entre ellos. Estos activos quedan registrados en el *Repositorio de conocimiento organizacional* gracias al *Codificador de activos de conocimiento pre-existente*. Los activos almacenados pueden ser seleccionados como entradas del proceso Distribución del Conocimiento por medio del *Publicador de conocimiento*, resultando en la creación de distintas vistas del conocimiento (estática offline, estática online y dinámica online). Cada *Visualizador de conocimiento* permite al equipo de trabajo realizar consultas sobre el conocimiento organizacional para llevar a cabo el proceso Aplicación del Conocimiento dentro del proyecto correspondiente. Durante la ejecución de los distintos proyectos de la organización se puede generar nuevo conocimiento tácito, el cual puede ser recogido a través del *Registrador de conocimiento tácito* durante el proceso Preservación del Conocimiento,

quedando temporalmente almacenado en la *Base de datos asociada al proyecto* en el cual haya sido generado dicho conocimiento. Este nuevo conocimiento tácito es exportado desde las bases de datos de los distintos proyectos de forma que puede pasar a formar parte del conocimiento de la organización mediante el *Gestor de cambios en el conocimiento* durante el proceso Gestión de Cambios. Finalmente, una *Herramienta de analítica web* permite realizar el proceso Medición del Conocimiento para monitorizar y analizar el uso del sistema y la evolución del conocimiento de la organización a través de un *Generador de informes*.

En las siguientes secciones se describen en detalle cada uno de los elementos que conforman la arquitectura del sistema representada en la Figura 4.1. Para cada elemento se indican las características necesarias para soportar los procesos y las actividades descritas en el capítulo anterior, así como las herramientas y tecnologías que han sido utilizadas en el ámbito de esta tesis doctoral para su implementación. Para tener una visión global de la implementación realizada en el ámbito de esta tesis doctoral, la Figura 4.2 muestra un diagrama que recoge todos los componentes software del sistema propuesto y sus dependencias, además de los distintos artefactos que se producen al utilizar cada uno de ellos.

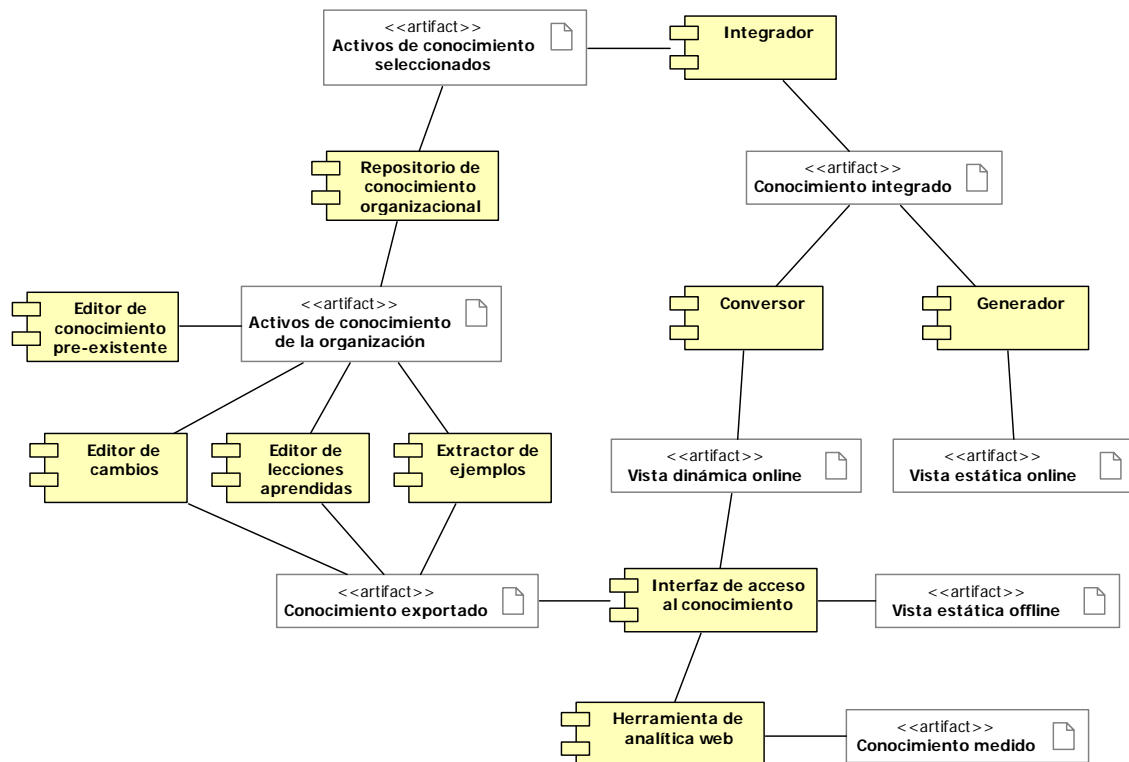


Figura 4.2. Diagrama de componentes del sistema propuesto.

## 4.2. Componentes para la persistencia del conocimiento organizacional

El conocimiento de la organización se encontrará almacenado en un repositorio común y centralizado, principalmente en forma de ficheros con formato XML (eXtensible Markup Language). La utilización de etiquetas en este tipo de ficheros permite transformar simples datos en información ya que estas etiquetas añaden un significado concreto y se establece un contexto. Esto da una mayor flexibilidad para estructurar documentos, siendo siempre dicha

estructura fácil de entender y de procesar. Además, para facilitar la gestión del conocimiento de la organización, cada activo de conocimiento estará almacenado en un fichero XML distinto.

## Funcionalidades

La herramienta que actúe como repositorio del conocimiento organizacional deberá proporcionar:

- Un mecanismo de almacenamiento centralizado de archivos en formato XML y archivos multimedia (imágenes, vídeos, documentos, etc.).
- La posibilidad de realizar cambios sobre los elementos almacenados, tales como modificaciones parciales, añadir, borrar, renombrar o mover elementos.
- Un registro histórico de las acciones realizadas sobre cada elemento, pudiendo además recuperar cualquier versión anterior de un elemento concreto.
- La posibilidad de generar informes con los archivos que han sido modificados en un periodo de tiempo concreto o con los cambios introducidos entre dos versiones de un mismo archivo.
- El marcado del estado de un conjunto de archivos en un instante determinado del tiempo mediante un identificador, pudiendo así recuperar en cualquier momento un estado concreto de dicho conjunto de archivos (versión).

## Implementación

En esta tesis doctoral se ha utilizado como repositorio del conocimiento organizacional una herramienta de control de versiones, concretamente la aplicación Microsoft Visual SourceSafe (VSS), aunque se podrían utilizar otras herramientas similares como por ejemplo Subversion. VSS es una solución orientada a pequeños proyectos de desarrollo de software. Inicialmente está destinada para desarrolladores que buscan una forma sencilla de gestionar los cambios en su código fuente (control de versiones), aunque puede ser utilizada para gestionar otros tipos de ficheros, como es este caso.

El acceso a los ficheros almacenados en VSS se realiza a través de una herramienta cliente, la cual puede ser la propia aplicación Windows proporcionada por VSS, la herramienta de línea de comandos, o cualquier otra aplicación que se integre con estos clientes o emule sus características.

El funcionamiento de este tipo de herramientas se basa principalmente en dos acciones. La primera, conocida como "Check In", guarda en VSS una copia de los ficheros locales seleccionados. La segunda, conocida como "Check Out", guarda una copia local de los ficheros seleccionados en VSS. Mediante el uso de estas dos acciones, los administradores del repositorio del conocimiento organizacional pueden fácilmente actualizar el conocimiento almacenado en el repositorio u obtener una copia local de parte o de todo el conocimiento de la organización.

### 4.3. Componentes para la formalización del conocimiento pre-existente

La gestión del contenido de un fichero XML se puede realizar por medio de un simple editor de texto plano. Sin embargo, para facilitar la edición del conocimiento en los ficheros XML, se recomienda utilizar una aplicación que se encargue de los detalles de la sintaxis, evitando así la necesidad de conocer el código XML por parte de las personas encargadas de la edición.

#### Funcionalidades

La herramienta que actúe como editor de conocimiento deberá proporcionar:

- Formularios para la entrada y visualización de datos basados en XML, los cuales permitirán crear o modificar datos en ficheros XML de una forma visualmente sencilla y estructurada.
- La posibilidad de definir reglas para la introducción de datos en los distintos campos de los formularios, definir reglas de formato condicional para cambiar el aspecto de los controles del formulario en función de los datos introducidos.
- La posibilidad de incluir acciones tales como alertas, valores calculados y validación de datos, o crear validaciones y acciones que se muestren según el tipo de campo.

#### Implementación

En esta tesis doctoral se ha utilizado como editor de conocimiento la aplicación Microsoft InfoPath. Esta aplicación permite diseñar formularios para la visualización y la edición de datos almacenados en ficheros XML.

Los formularios pueden desarrollarse fácilmente sin necesidad de código, simplemente arrastrando los distintos controles que se quieren incluir en el formulario. Además, en los formularios de Infopath también se pueden definir reglas para la introducción de datos, reglas de formato condicional, mostrar alertas, realizar la validación de los datos introducidos o calcular valores a partir de los mismos.

En el Anexo B: Plantillas para Editar el Conocimiento Pre-Existente se pueden ver las plantillas diseñadas con Infopath para gestionar los distintos tipos de activos de conocimiento almacenados en el repositorio de conocimiento organizacional.

### 4.4. Componentes para la distribución del conocimiento organizacional

Para realizar la distribución del conocimiento de la organización entre los distintos equipos de trabajo se utiliza un publicador de conocimiento. Este subsistema está compuesto por los componentes que se muestran en la Figura 4.3.



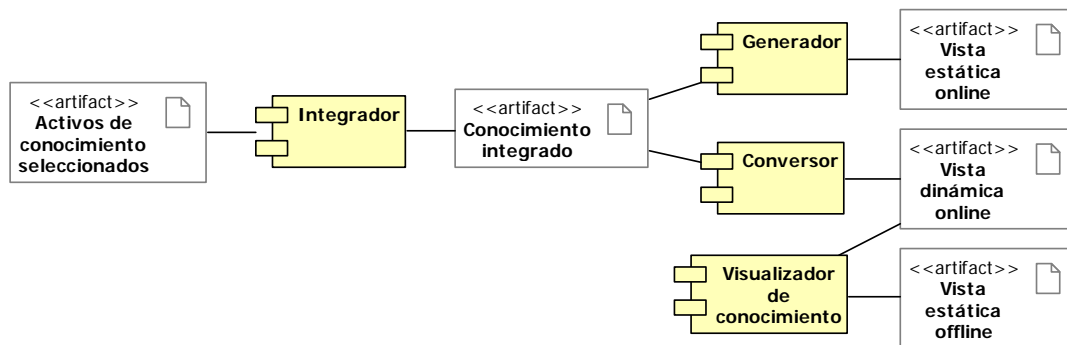


Figura 4.3. Diagrama de componentes para la distribución del conocimiento organizacional.

El *Integrador* de contenidos es un componente que permite empaquetar bajo un único soporte todos los activos de conocimiento organizacional que hayan sido previamente seleccionados para ser publicados y distribuidos para un proyecto concreto. A partir de dichos activos integrados se publicará una vista con el formato más adecuado para el equipo del proyecto que la utilizará: para el caso de una vista estática online se utilizará un *Generador* de vistas; para el caso de una vista dinámica online se utilizará un *Conversor* que permitirá transformar el conocimiento integrado a un formato importable por el *Visualizador de conocimiento*; por último, para una vista estática offline se exportará desde el visualizador el conocimiento disponible en la vista dinámica online.

A continuación se presentarán más en detalle los componentes Integrador, Generador y Conversor por ser los principales elementos involucrados en la distribución de conocimiento, mientras que el Visualizador de conocimiento será descrito en el apartado 4.5 ya que es el componente principal utilizado para la aplicación y preservación de conocimiento.

#### 4.4.1. Integrador

Para una gestión más sencilla del conocimiento de la organización, cada activo de conocimiento está registrado en el repositorio en un archivo XML distinto. Sin embargo, para la distribución de una versión concreta del conocimiento de la organización, una vez seleccionados los activos que se quieren distribuir, es necesaria una aplicación que empaquete esos activos y las relaciones existentes entre ellos para poder generar después una vista de dicho conocimiento.

#### Funcionalidades

La herramienta que actúe como integrador de conocimiento deberá realizar:

- La unión de los activos de conocimiento seleccionados para su distribución dentro de un único archivo XML. Esto se realizará partiendo de un esqueleto que establecerá una estructura jerárquica de los activos, estructura que vendrá determinada por el generador de vistas.
- El establecimiento de relaciones bidireccionales entre los activos de conocimiento a partir de las relaciones unidireccionales existentes. Es decir, cuando un activo esté enlazado con otro activo, el integrador debe insertar en el segundo otro enlace hacia el primero. Esto permitirá simplificar la edición de conocimiento ya que solo será

necesario introducir en los formularios un sentido de la relación entre dos activos, encargándose el integrador de introducir la relación en sentido contrario.

### Implementación

Puesto que no se ha encontrado ninguna aplicación que se adapte a las características requeridas para el Integrador, en esta tesis doctoral se ha optado por desarrollar esta herramienta ad-hoc. El desarrollo se ha realizado en lenguaje C# (C-Sharp) utilizado para ello la aplicación Microsoft Visual Studio.

Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros. Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002).

El diagrama de clases diseñado para desarrollar el Integrador se muestra en la Figura 4.4.

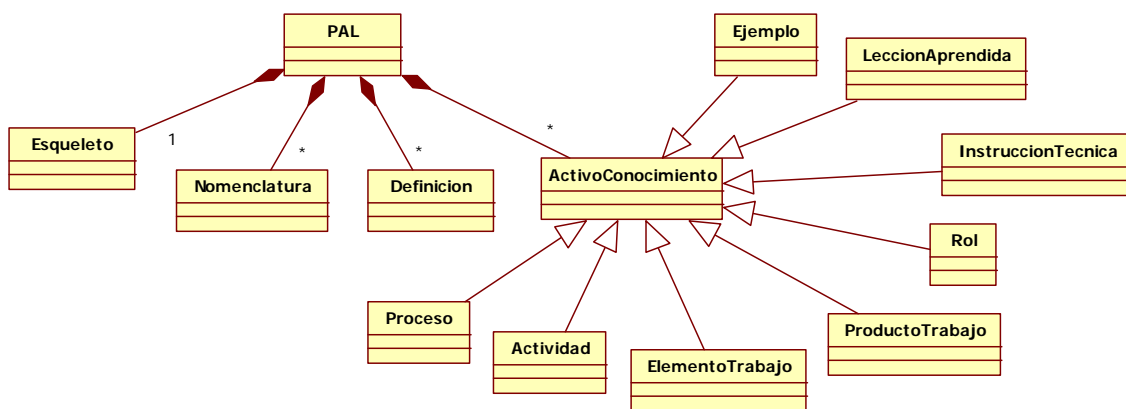


Figura 4.4. Diagrama de clases del Integrador.

En la Figura 4.5 se presentan las relaciones entre los activos de conocimiento. El Integrador deberá establecer relaciones bidireccionales entre estos elementos a partir de las relaciones unidireccionales definidas entre los mismos y registradas en el repositorio de conocimiento organizacional.

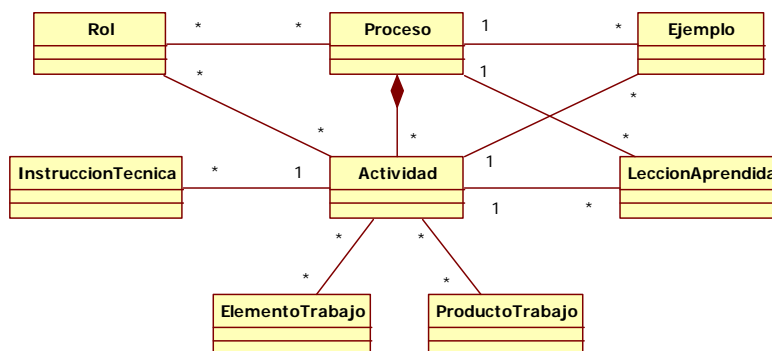


Figura 4.5. Relaciones entre activos de conocimiento.

### 4.4.2. Generador

El objetivo del subsistema de distribución del conocimiento organizacional es la creación y posterior publicación de una vista que permita acceder a una versión del conocimiento de la organización, teniendo dicha vista el formato más adecuado para el equipo del proyecto que la vaya a utilizar.

Uno de estos formatos es la vista estática online, la cual se caracteriza por no poder ser modificada directamente por las personas que tienen acceso a la misma y por su disponibilidad a través de internet. En este caso, se ha optado por utilizar páginas HTML como vista estática online. Por lo tanto, es necesaria una aplicación que permita la generación de páginas HTML partiendo del conocimiento previamente empaquetado por el Integrador.

### Funcionalidades

La herramienta que actúe como generador de vistas deberá proporcionar:

- Generación de un conjunto de páginas en formato HTML a partir de información contenida en un fichero estructurado.
- Definición de la estructura a la que debe ajustarse el fichero de entrada que contiene la información relativa a los activos de conocimiento de la organización.
- Posibilidad de personalizar los contenidos para adaptarlos según los activos de conocimiento de la organización.
- Posibilidad de personalizar el aspecto de las páginas generadas para adaptarlas a la imagen corporativa de la organización.

### Implementación

Para realizar la generación de vistas estáticas online se utilizó en esta tesis la herramienta Visual Studio Team System (VSTS) Process Guidance Generator, la cual se adapta a las necesidades requeridas para el Generador. VSTS Process Guidance Generator es una herramienta de libre distribución desarrollada originalmente por personal de Microsoft para generar las páginas de la Microsoft Software Foundation (MSF) Guidance, la cual ya fue mencionada en el análisis de soluciones existentes realizado en el estado de la cuestión.

Esta herramienta permite generar una EPG estática en forma de páginas HTML a partir de la información almacenada en un fichero XML. El contenido de dicho fichero XML es modificable para adaptarlo según los procesos de la organización. La estructura del fichero de entrada está definida mediante XML Schema. La herramienta utiliza también unas plantillas HTML para establecer la estructura de cada una de las páginas que conforman la EPG, como son por ejemplo los menús o la distribución de los propios contenidos en distintas páginas. Además, una vez generados los ficheros HTML, una hoja de estilos permite modificar el aspecto visual de dichas páginas.

### 4.4.3. Conversor

Otro de los formatos que puede tener el Visualizador de conocimiento utilizado por un equipo de proyecto es la vista dinámica online. Esta se caracteriza por su disponibilidad a través de internet y por permitir que todas las personas que tienen acceso a la misma puedan realizar contribuciones para la mejora del conocimiento organizacional. En este caso, se ha optado por utilizar una wiki como vista dinámica online. Más concretamente, tal y como se detalla más adelante, se utilizará MediaWiki como herramienta para desplegar este tipo de vista dinámica online. Por lo tanto, es necesaria una aplicación que transforme el conocimiento previamente empaquetado por el Integrador a un formato que pueda ser importado fácilmente desde la wiki para su publicación.

### Funcionalidades

La herramienta que actúe como conversor de formatos deberá llevar a cabo:

- La creación de un conjunto de páginas en formato importable desde MediaWiki a partir de la versión del conocimiento organizacional almacenada en un único fichero XML con estructura definida por el Generador.
- La creación de un conjunto de páginas en formato importable desde MediaWiki donde se listen los distintos activos de conocimiento de la organización a los cuales ofrece acceso la wiki.
- La creación de una página en formato importable desde MediaWiki que contenga un menú lateral para la wiki con el objetivo de proporcionar acceso a los principales contenidos ofrecidos.

### Implementación

Puesto que no se ha encontrado ninguna aplicación que se adapte a las características requeridas para el Conversor, en esta tesis doctoral se ha optado por desarrollar esta herramienta ad-hoc. El desarrollo se ha realizado en lenguaje C# (C-Sharp) utilizando para ello la aplicación Microsoft Visual Studio.

El diagrama de clases diseñado para desarrollar el Conversor se muestra en la Figura 4.6.

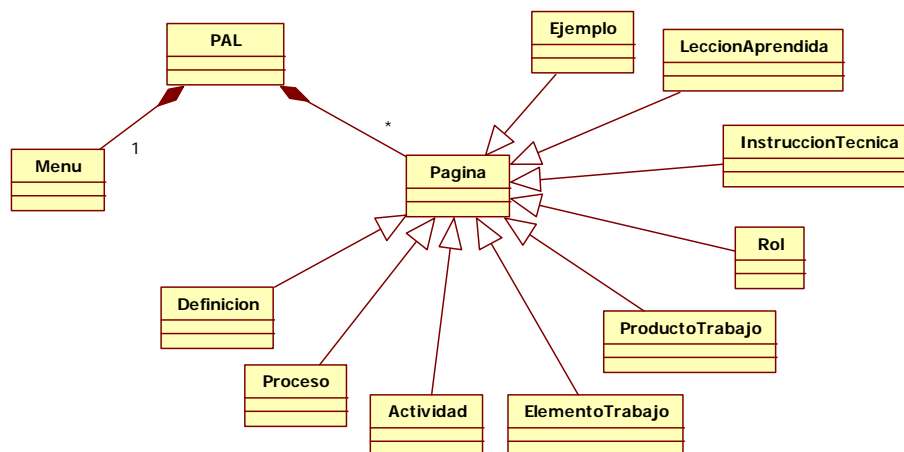


Figura 4.6. Diagrama de clases del Conversor.

## 4.5. Componentes para la aplicación y preservación de conocimiento

Dentro del proceso Aplicación del Conocimiento, el equipo de proyecto necesita disponer de una herramienta que proporcione acceso al conocimiento de la organización para poder consultar las actividades que deben ser realizadas para el desarrollo del proyecto. Como consecuencia de la ejecución de esas actividades puede surgir nuevo conocimiento tácito, por lo que el equipo de proyecto también necesita una herramienta que facilite la recogida de dicho conocimiento durante el proceso Preservación del Conocimiento.

Las wikis son herramientas que permiten realizar ambas tareas –visualización de conocimiento y recogida de nuevo conocimiento– y que, como ya se ha mencionado anteriormente, constituyen una vista dinámica online. Es por ello que en este apartado se agrupan los componentes necesarios tanto para la aplicación como para la preservación de conocimiento.

Este apartado se centra en las funcionalidades necesarias para desplegar una vista dinámica online y en la implementación realizada en el ámbito de esta tesis doctoral; esto se debe a que las posibilidades que ofrece dicha vista son las más interesantes desde el punto de vista del enriquecimiento colaborativo del conocimiento organizacional. Para el caso de las vistas estáticas (tanto online como offline), las herramientas que permiten la visualización de conocimiento no permiten también la recogida de nuevo conocimiento; en dichos casos sería necesario incluir otros mecanismos para dar soporte al proceso Preservación del Conocimiento, tales como la realización de contribuciones mediante correo electrónico o la publicación de mensajes en un foro.

### Funcionalidades

La herramienta que permita el despliegue de la vista dinámica online en forma de wiki deberá ofrecer principalmente las siguientes características:

- Incluir páginas de discusión, de forma que cada página de la wiki tenga una página de discusión propia para recoger nuevo conocimiento tácito.
- Soportar plantillas personalizadas con parámetros para facilitar la inserción de nuevo conocimiento y la modificación del ya existente.
- Poseer un sistema extensiones que permitan ampliar fácilmente las funcionalidades proporcionadas por defecto.
- Utilizar un lenguaje de marcación que permita crear páginas en la wiki, así como la posibilidad de soportar también marcaciones HTML.
- Utilizar espacios de nombres para separar páginas de distintos tipos que puedan ser tratados de distinta forma.
- Incluir un sistema de categorías jerárquico que permita crear listados.
- Proporcionar listas de seguimiento para que cada usuario pueda estar informado acerca de los cambios en el conocimiento que le resulten de interés.
- Ofrecer la posibilidad de personalizar la interfaz.

## Implementación

La herramienta elegida en esta tesis para implementar la vista dinámica online en forma de wiki es MediaWiki. Este es un motor para wikis programado en PHP y que se encuentra bajo licencia de software GNU General Public License. La herramienta fue originalmente desarrollada por la fundación Wikimedia para su uso en diversos proyectos como Wikipedia, Wictionary, Wikibooks, etc. y ha tenido una gran expansión desde el año 2005, existiendo actualmente un gran número de wikis basadas en este software que nada tienen que ver con dicha fundación. Puede ser instalado sobre servidores Web Apache o IIS, y puede usar como motor de base de datos MySQL o PostgreSQL.

MediaWiki no puede ser utilizada directamente como repositorio de conocimiento sobre activos de procesos de software ya que la instalación inicial proporciona únicamente las funcionalidades básicas de creación y edición de páginas. Es por tanto necesario configurarla adecuadamente para obtener un buen desempeño.

Las wikis, por defecto, son un sistema de libre acceso para todos los usuarios, algo que no es admisible en este caso ya que una organización, por lo general, solo querrá compartir su conocimiento entre sus empleados y, en ocasiones, con alguna empresa colaboradora. Por lo tanto, partiendo de la instalación inicial, MediaWiki se debe configurar para que el acceso sea restringido de forma que solo los usuarios registrados pueden consultar y editar el conocimiento organizacional. Además, solo un determinado grupo de usuarios de la wiki (los administradores) podrán crear nuevas cuentas de usuario.

Por otro lado, la carga de archivos es una funcionalidad habitual que proporcionan las wikis, aunque por defecto suele estar deshabilitada por motivos de seguridad. Esta característica debe habilitarse para que cualquier usuario registrado previamente en la wiki pueda subir distintos tipos de archivo que representarán activos de procesos de software y ejemplos.

Sin embargo, las características básicas que se acaban de mencionar no son suficientes para proporcionar la funcionalidad necesaria para que la wiki actúe como vista dinámica online. Es necesario por tanto realizar una configuración avanzada que proporcione funcionalidades mucho más elaboradas, poco habituales en la mayoría de wikis, para lo cual se deben utilizar herramientas desarrolladas por terceros denominadas “extensiones”.

En este caso, se hace necesaria la instalación de extensiones que permitan el uso de plantillas y formularios para estructurar los contenidos de cada activo de conocimiento, la división de contenidos extensos mediante pestañas para una mejor visualización, la inclusión de contenidos multimedia, la generación de un archivo PDF (vista estática offline) que recoja todo o parte del conocimiento organizacional, y la obtención de distintas estadísticas acerca de la utilización del sistema por parte de los usuarios (necesario tanto para el proceso Medición del Conocimiento como para recoger los datos necesarios para la validación de ITAKA).

Más concretamente, las extensiones utilizadas en esta tesis para ofrecer las funcionalidades mencionadas antes son las que se detallan a continuación:

- **Extensión Semantic MediaWiki.** Ayuda a buscar, organizar, etiquetar, examinar, evaluar y compartir los contenidos de la wiki, añadiendo anotaciones semánticas que

permiten a la wiki funcionar como una base de datos colaborativa. Aunque puede parecer que se está aumentando la complejidad, en realidad se simplifica la estructura de la wiki puesto que esta extensión ayuda a los usuarios a localizar más información en menos tiempo, y a mejorar la calidad general y la consistencia de la wiki.

- **Extensión Semantic Forms.** Permite a los usuarios añadir, editar y hacer peticiones de datos por medio de formularios, eliminando así la necesidad de utilizar el lenguaje de marcado de MediaWiki. Esta extensión está muy ligada a la extensión Semantic MediaWiki ya que está pensada para ser utilizada para estructurar datos que contengan anotaciones semánticas. Sin embargo, Semantic Forms no soporta directamente el marcado semántico en las páginas de la wiki, sino que este marcado semántico es añadido indirectamente por el usuario al introducir datos en las plantillas de los formularios. La utilización conjunta de las extensiones Semantic MediaWiki y Semantic Forms proporciona una estructura de datos sólida a la wiki.
- **Extensión HeaderTabs.** Reemplaza los encabezados de primer nivel de las páginas de la wiki por pestañas. Así se pueden mostrar páginas extensas de la wiki de una forma visualmente más estructurada.
- **Extensión Flash.** Permite la inserción de vídeos en formato Flash dentro de una página de la wiki. Esto permitirá mostrar, por ejemplo, transparencias de ayuda al aprendizaje que hayan sido convertidas previamente a formato Flash.
- **Extensión PdfBook.** Ofrece la posibilidad de componer un libro utilizando distintas páginas de la wiki y exportarlo después a un fichero con formato PDF, añadiendo además una tabla de contenidos. Así, este fichero PDF será una versión imprimible del conocimiento organizacional mostrado en la wiki en un determinado momento en el tiempo. Además, al ser un fichero que puede ser descargado, permitirá consultar el conocimiento de la organización sin necesidad de conexión a Internet.
- **Extensión CommunityVoice.** Permite a los usuarios calificar la calidad de los distintos contenidos de la wiki. Utilizando esta extensión se pueden añadir escalas de valoración en cada una de las páginas de la wiki, de manera que los usuarios registrados puedan valorar la calidad de cada página otorgándole entre 1 y 5 estrellas, por ejemplo.
- **Extensión Google Analytics Integration.** Añade códigos de rastreo de Google Analytics en todas las páginas de la wiki que son visitadas por los usuarios. La extensión excluye las páginas visualizadas por administradores y bots para evitar contaminar las estadísticas de uso. Para utilizar esta extensión es necesario disponer de una cuenta Google Analytics en donde se visualizarán las estadísticas de uso recogidas.
- **Extensión FireStats.** Se integra en la wiki para ofrecer estadísticas adicionales a las proporcionadas por MediaWiki. Por ejemplo, esta extensión permite obtener estadísticas en tiempo real acerca de los accesos a la wiki, palabras clave utilizadas en el buscador, páginas más populares, países desde los que acceden los usuarios, etc.

## 4.6. Componentes para la gestión de cambios en el conocimiento organizacional

Para trasladar el nuevo conocimiento tácito recogido hasta el repositorio de conocimiento organizacional se utiliza un gestor de cambios del conocimiento. Este subsistema está compuesto por los componentes que se muestran en la Figura 4.7.

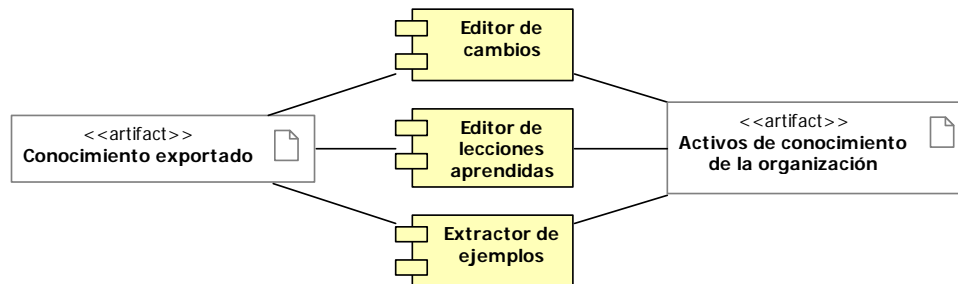


Figura 4.7. Diagrama de componentes para la distribución del conocimiento organizacional.

Para realizar la realimentación del conocimiento generado durante la ejecución de los proyectos con el objetivo de enriquecer el conocimiento organizacional, en primer lugar se debe exportar ese nuevo conocimiento que se encuentra almacenado en la base de datos asociada al proyecto. A partir de este conocimiento exportado, distintas herramientas permiten tratar el nuevo conocimiento tácito para el enriquecimiento de los activos de conocimiento almacenados en el repositorio de conocimiento de la organización. Por un lado, el *Editor de cambios* es el encargado de gestionar las modificaciones realizadas en el conocimiento explícito asociado a los activos de conocimiento de la organización. Por otro lado, el *Editor de lecciones aprendidas* facilita la gestión del nuevo conocimiento tácito generado en forma de discusiones durante la ejecución del proyecto para su almacenamiento en el repositorio de conocimiento organizacional en forma de lecciones aprendidas. Finalmente, el *Extractor de ejemplos* proporciona soporte para la gestión del nuevo conocimiento tácito generado en forma de ejemplos durante la ejecución del proyecto.

Todos estos componentes involucrados proporcionan las funcionalidades necesarias para poder llevar a cabo las dos alternativas propuestas en esta tesis para la gestión de cambios en el conocimiento de la organización: con revisión de cambios (supervisada) o sin revisión de cambios (no supervisada). A continuación se presentan más en detalle cada uno de estos componentes.

### 4.6.1. Editor de cambios

El Editor de cambios es el componente encargado de detectar y gestionar las modificaciones realizadas en el conocimiento explícito de cada uno de los activos de conocimiento de la organización.

#### Funcionalidades

La herramienta que proporcione soporte para la gestión de cambios en el conocimiento explícito de la organización deberá permitir:



- Detectar las modificaciones realizadas en un activo de conocimiento dentro del ámbito de un proyecto en relación con la anterior versión publicada del mismo activo de conocimiento para dicho proyecto.
- Comparar el activo de conocimiento modificado con el activo de conocimiento original correspondiente almacenado en el repositorio de conocimiento organizacional.
- Modificar el activo de conocimiento almacenado en el repositorio de conocimiento organizacional según los cambios realizados.
- Permitir la intervención del usuario en el proceso de gestión de cambios en el conocimiento explícito de la organización de manera que se pueda decidir aceptar los cambios, rechazarlos o realizar alguna modificación adicional.

## Implementación

Puesto que no se ha encontrado ninguna aplicación que se adapte a las características requeridas para el Editor de cambios, en esta tesis doctoral se ha optado por desarrollar esta herramienta ad-hoc. El desarrollo se ha realizado en lenguaje C# (C-Sharp) mediante la aplicación Microsoft Visual Studio en base al diagrama de clases diseñado para esta herramienta que se muestra en la Figura 4.8.

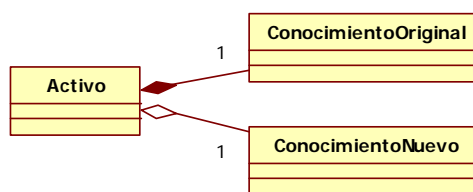


Figura 4.8. Diagrama de clases del Editor de cambios.

### 4.6.2. Editor de lecciones aprendidas

El Editor de lecciones aprendidas es el componente encargado de detectar y gestionar las discusiones registradas en el ámbito del proyecto en relación con cada uno de los activos de conocimiento de la organización, así como de la inclusión de las mismas en el repositorio de conocimiento organizacional en forma de lecciones aprendidas.

## Funcionalidades

La herramienta que proporcione soporte para la gestión del nuevo conocimiento tácito generado en forma de discusiones deberá permitir:

- Detectar las discusiones creadas en relación con un activo de conocimiento dentro del ámbito de un proyecto.
- Recuperar las lecciones aprendidas relacionadas con el mismo activo de conocimiento que se encuentren almacenadas en el repositorio de conocimiento organizacional.
- Convertir las discusiones en lecciones aprendidas y fusionarlas con las lecciones aprendidas existentes en el repositorio de conocimiento organizacional para ese mismo activo de conocimiento.

- Permitir la intervención del usuario en el proceso de gestión del nuevo conocimiento tácito generado en forma de discusiones de manera que se pueda decidir aceptar las discusiones, rechazarlas o realizar alguna modificación adicional.

### Implementación

Puesto que no se ha encontrado ninguna aplicación que se adapte a las características requeridas para el Editor de lecciones aprendidas, en esta tesis doctoral se ha optado por desarrollar esta herramienta ad-hoc. El desarrollo se ha realizado en lenguaje C# (C-Sharp) mediante la aplicación Microsoft Visual Studio en base al diagrama de clases diseñado para esta herramienta que se muestra en la Figura 4.9.

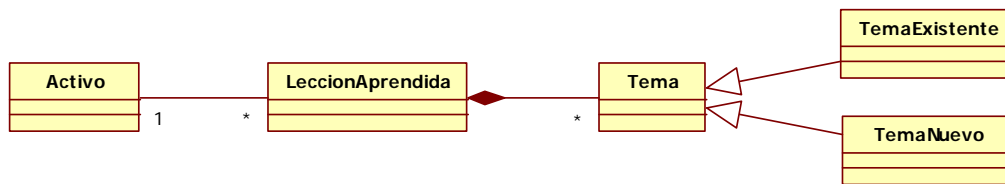


Figura 4.9. Diagrama de clases del Editor de lecciones aprendidas.

El resultado es un editor visual (Figura 4.10) que muestra en la parte izquierda las discusiones registradas durante la ejecución del proyecto y en la parte derecha las lecciones aprendidas almacenadas en el repositorio de conocimiento organizacional. Para un manejo más sencillo de esta herramienta, el conocimiento se muestra clasificado según el activo de conocimiento con el que está relacionado y, además, según el tema al que pertenezcan las discusiones o las lecciones aprendidas, pudiendo el usuario seleccionar el activo y el título del tema concretos que quiere gestionar. Para la gestión supervisada basta con indicar qué porción de una discusión se quiere transformar en lección aprendida y en lugar dónde se quiere introducir este nuevo conocimiento. Asimismo, el editor permite realizar modificaciones adicionales o revertir las últimas modificaciones realizadas por el usuario.

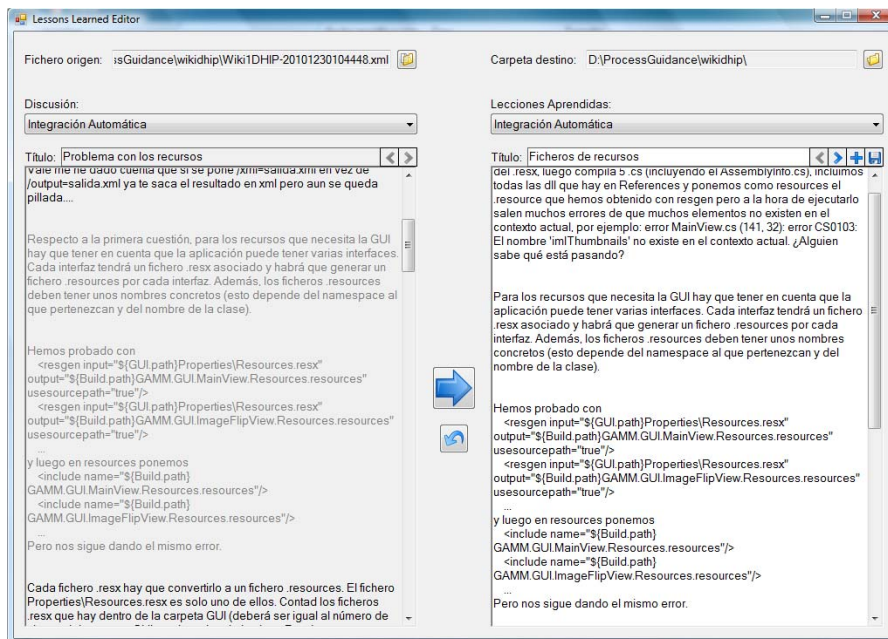


Figura 4.10. Fusión de discusiones en forma de lecciones aprendidas utilizando el editor.

### 4.6.3. Extractor de ejemplos

El Extractor de ejemplos es el componente encargado de detectar y gestionar los ejemplos registrados en el ámbito del proyecto en relación con cada uno de los activos de conocimiento de la organización.

#### Funcionalidades

La herramienta que proporcione soporte para la gestión del nuevo conocimiento tácito generado en forma de ejemplos deberá permitir:

- Detectar los nuevos ejemplos registrados en relación con un activo de conocimiento dentro del ámbito de un proyecto.
- Incluir los nuevos ejemplos en el conjunto de ejemplos ya existentes en el repositorio de conocimiento organizacional para ese mismo activo de conocimiento.
- Permitir la intervención del usuario en el proceso de gestión del nuevo conocimiento tácito generado en forma de ejemplos de manera que se pueda decidir aceptar los nuevos ejemplos, rechazarlos o realizar alguna modificación adicional.

#### Implementación

Puesto que no se ha encontrado ninguna aplicación que se adapte a las características requeridas para el Extractor de ejemplos, en esta tesis doctoral se ha optado por desarrollar esta herramienta ad-hoc. El desarrollo se ha realizado en lenguaje C# (C-Sharp) mediante la aplicación Microsoft Visual Studio en base al diagrama de clases diseñado para esta herramienta que se muestra en la Figura 4.11.

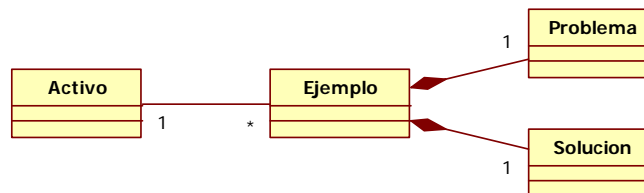


Figura 4.11. Diagrama de clases del Extractor de ejemplos.

## 4.7. Componentes para la medición de conocimiento

Para evaluar si el conocimiento de la organización es gestionado de manera adecuada y eficiente es necesaria una herramienta que proporcione datos acerca de la utilización de los activos de conocimiento de la organización por parte de los distintos miembros de los equipos de proyecto.

#### Funcionalidades

La herramienta que permita la medición de conocimiento deberá:

- Obtener datos estadísticos acerca del comportamiento de los miembros de los equipos de proyecto en relación con la visualización de los activos de conocimiento de la

organización, tales como número de accesos, número de activos consultados en cada visita, o tiempo dedicado a la utilización del sistema.

- Permitir la generación de distintos informes, tanto numéricos como gráficos, a partir de los datos estadísticos recogidos.

## **Implementación**

En esta tesis doctoral se han utilizado para la medición del conocimiento dos herramientas de analítica Web, concretamente el servicio Google Analytics y la extensión FireStats para MediaWiki. Ambas herramientas realizan la recogida de numerosos datos estadísticos sobre sitios Web y permiten además obtener informes acerca de los accesos y de la navegación de los usuarios.

# Capítulo 5: Validación de la Solución Propuesta

5.1. Descripción del método.....	134
5.2. Contexto.....	135
5.3. Planificación.....	137
5.4. Recogida de datos y métodos de análisis.....	140
5.4.1. PI1: Efectividad de ITAKA para capturar el conocimiento tácito.....	140
5.4.2. PI2: Factores que facilitan el uso de ITAKA como fuente de conocimiento relevante.....	141
5.4.3. PI3: Nivel de calidad del nuevo conocimiento elicitado al utilizar ITAKA.....	141
5.4.4. PI4: Factores que influyen en la calidad de los activos de conocimiento.....	142
5.4.5. PI5: Efectividad de las distintas alternativas (supervisada y no-supervisada).....	142
5.4.6. PI6: Contribución de ITAKA a la mejora de la facilidad del aprendizaje de procesos.....	142
5.4.7. PI7: Evaluación subjetiva de la utilidad de los mecanismos de gestión de la configuración propuestos.....	143
5.5. Resultados.....	143
5.5.1. PI1: Efectividad de ITAKA para capturar el conocimiento tácito.....	143
5.5.2. PI2: Factores que facilitan el uso de ITAKA como fuente de conocimiento relevante.....	146
5.5.3. PI3: Nivel de calidad del nuevo conocimiento elicitado al utilizar ITAKA.....	150
5.5.4. PI4: Factores que influyen en la calidad de los activos de conocimiento.....	153
5.5.5. PI5: Efectividad de las distintas alternativas (supervisada y no-supervisada).....	157
5.5.6. PI6: Contribución de ITAKA a la mejora de la facilidad del aprendizaje de procesos.....	158
5.5.7. PI7: Evaluación subjetiva de la utilidad de los mecanismos de gestión de la configuración propuestos.....	161
5.6. Conclusiones de la validación.....	163



La hipótesis enunciada para esta tesis doctoral, y que se trata de validar mediante el desarrollo de esta fase, es la siguiente:

Si se dispone de un marco de trabajo para la gestión del conocimiento destinado al aprendizaje y adopción de procesos de software, basado en una PAL y que incorpora mecanismos de gestión de la configuración y de gestión del cambio, **entonces**:

- **H1.1:** Se incrementa la efectividad en la captura del conocimiento tácito generado por los ingenieros de software durante la ejecución de los proyectos de la organización, convirtiendo la PAL en una fuente relevante de conocimiento.
- **H1.2:** Se facilita el enriquecimiento y la diseminación de activos de conocimiento de alta calidad elicitados durante la ejecución de la iniciativa de aprendizaje.
- **H1.3:** Se ayuda a mejorar la capacidad para el aprendizaje y la adopción de nuevas prácticas para el desarrollo de software dentro de la organización.

Para validar estas sub-hipótesis se llevó a cabo un experimento controlado que permitió comprobar si ITAKA contribuye a proporcionar un mecanismo efectivo para ayudar a los ingenieros de software a generar, acceder, reutilizar y preservar el conocimiento necesario para aprender adecuadamente un nuevo proceso software que esté siendo introducido en una organización de desarrollo de software. Así, las preguntas de investigación que se plantean para esta fase de validación son los que se exponen a continuación.

En relación con el análisis de la efectividad de los mecanismos de adquisición de conocimiento tácito para crear un repositorio de conocimiento organizacional (**H1.1**):

- **PI1:** ¿Constituye ITAKA una aproximación efectiva para capturar el conocimiento tácito generado por los ingenieros de software durante la ejecución de los proyectos y almacenarlo junto con el restante conocimiento organizacional?
- **PI2:** ¿Cuáles son los factores que facilitan el uso de ITAKA como fuente de conocimiento relevante para los ingenieros de software para el aprendizaje y adopción de un nuevo proceso software?

En relación con el análisis de la calidad de los activos de conocimiento (ejemplos, discusiones y lecciones aprendidas) elicitados durante la ejecución de la iniciativa de aprendizaje (**H1.2**):

- **PI3:** ¿Qué nivel de calidad tiene el nuevo conocimiento elicitado por los ingenieros de software al utilizar ITAKA?
- **PI4:** ¿Cuáles son los factores que influyen en la calidad de los activos de conocimiento generados al utilizar ITAKA?
- **PI5:** ¿Cuál es la efectividad de las distintas alternativas –supervisada y no-supervisada– para la incorporación de los cambios propuestos por los ingenieros de software durante el proceso de gestión de cambios?

En relación con la mejora de la facilidad en el aprendizaje y la adopción de nuevas prácticas para el desarrollo de software dentro de la organización (**H1.3**):

- **PI6:** ¿Contribuye ITAKA a la mejora de la facilidad del aprendizaje de nuevos procesos software que deban ser adoptados por los ingenieros de software?
- **PI7:** ¿Cuál es la evaluación subjetiva de la utilidad de los mecanismos de gestión de la configuración propuestos para preservar y enriquecer el conocimiento organizacional?

La Tabla 5.1 recoge esta información de manera resumida.

**Tabla 5.1. Relación de hipótesis y preguntas de investigación.**

Hipótesis	Preguntas de investigación
<b>H1.1:</b> Se incrementa la efectividad en la captura del conocimiento tácito.	<b>PI1:</b> Efectividad de ITAKA para capturar el conocimiento tácito.
	<b>PI2:</b> Factores que facilitan el uso de ITAKA como fuente de conocimiento relevante.
<b>H1.2:</b> Se facilita el enriquecimiento y la diseminación de activos de conocimiento de alta calidad.	<b>PI3:</b> Nivel de calidad del nuevo conocimiento elicitado al utilizar ITAKA.
	<b>PI4:</b> Factores que influyen en la calidad de los activos de conocimiento.
	<b>PI5:</b> Efectividad de las distintas alternativas (supervisada y no-supervisada).
<b>H1.3:</b> Se ayuda a mejorar la capacidad para el aprendizaje y la adopción de nuevas prácticas para el desarrollo de software.	<b>PI6:</b> Contribución de ITAKA a la mejora de la facilidad del aprendizaje de procesos.
	<b>PI7:</b> Evaluación subjetiva de la utilidad de los mecanismos de gestión de la configuración propuestos.

Las distintas secciones de este capítulo describen el método de experimentación utilizado para la validación, el contexto y la planificación del experimento realizado, la estrategia seguida para la recogida y el análisis de datos durante el experimento, y finalmente la presentación de los resultados obtenidos en relación con cada una de las preguntas de investigación mencionadas antes.

## 5.1. Descripción del método

Las técnicas de experimentación utilizadas para validar ITAKA fueron la realización de un experimento controlado [Wohlin, 2001], la observación de los participantes [Seaman, 1999], y la realización de encuestas [Wohlin, 2000] (p.11).

En un experimento controlado, el investigador tiene el control sobre el estudio y sobre cómo los participantes ejecutan las tareas que se les asignaron, a diferencia de los casos de estudio en donde el investigador es un simple observador [Wohlin, 2001]. Los experimentos se llevan a cabo habitualmente para comparar diferentes métodos, técnicas, procedimientos, etc. controlando quién utiliza un método y quién utiliza otro [Wohlin, 2000]. Dicha comparación se basa en el estudio de si las variables dependientes definidas para el experimento (por ejemplo, el esfuerzo) son influenciadas por las variables independientes (por ejemplo, aplicar un método u otro) y en qué medida se ven afectadas. En la presente tesis doctoral se optó por la realización de un experimento controlado al ser el método de investigación que más se



ajustaba a las necesidades para validar las hipótesis formuladas. El experimento controlado permite la comparación de la efectividad de ITAKA en relación con una PAL sin soporte para la gestión de la configuración ni para la gestión del cambio, así como determinar cómo influye la utilización de ITAKA sobre determinados aspectos como la calidad de los activos de conocimiento almacenados. Además, también ofrece la posibilidad de separar los participantes de manera que unos utilicen los mecanismos de gestión de cambios supervisada y que otros utilicen los mecanismos no-supervisados, lo cual permite comparar las características y beneficios de ambos mecanismos.

La observación de los participantes pretende capturar comportamientos e interacciones que podrían no ser percibidos de otra manera. La observación no implica que el observador esté vinculado a la tarea que realizan los participantes, sino simplemente que el observador está presente recogiendo datos con el consentimiento de los participantes [Seaman, 1999]. La observación de los participantes durante el experimento llevado a cabo en relación con esta tesis doctoral estuvo orientada a capturar información no cuantitativa en relación con el uso del sistema por parte de los participantes.

Las encuestas se llevan a cabo cuando el uso de una técnica o herramienta ya ha tenido lugar, o antes de ser introducida. Pueden ser vistas como una foto de la situación para capturar el estado actual. Aunque las encuestas pueden proporcionar un gran número de variables para evaluar, es necesario enfocarlas a la obtención de la mayor cantidad posible de información utilizando el menor número posible de variables para facilitar así su análisis. En general, las encuestas se aplican sobre una muestra de la población y la información recogida puede ser manejada de manera cuantitativa o cualitativa [Wohlin, 2000] (p.10). En el caso de esta tesis doctoral, las encuestas realizadas en el ámbito del experimento tuvieron la finalidad de recoger información acerca de cómo perciben los usuarios el sistema, principalmente en relación a su utilidad y calidad.

Para la experimentación llevada a cabo en esta tesis doctoral se han tenido en cuenta las pautas propuestas por Kitchenham et al. [Kitchenham, 2002] y por Wohlin et al. [Wohlin, 2000]. Las primeras presentan pautas generales para la experimentación en Ingeniería del Software y consejos acerca del diseño, ejecución, análisis y presentación de estudios empíricos, mientras que las segundas presentan las distintas fases con mayor detalle. Además, para la documentación del experimento que se realiza en este capítulo se han tomado como base las recomendaciones indicadas en [Jedlitschka, 2008] y en [Kitchenham, 2008] para que la información sea sencilla de localizar, se pueda comprender fácilmente cómo se ha realizado el experimento y se pueda evaluar la validez de los resultados (presentada anteriormente en el apartado 1.7), facilitando con todo ello que el experimento sea repetible.

## 5.2. Contexto

La implementación de ITAKA ha sido validada en un curso de formación sobre desarrollo ágil de software en el cual participaron 46 ingenieros de software junior, 6 mujeres y 40 hombres, con edades comprendidas entre los 22 y los 25 años, y que fueron divididos en 12 equipos de trabajo. El entorno de un curso de formación es adecuado para probar el marco de trabajo propuesto ya que requiere la colaboración entre los ingenieros de software para adoptar un

nuevo proceso de desarrollo de software, situación muy similar a la que encontrarán en una organización de ingeniería del software. Esto es algo habitual en organizaciones TIC de innovación que deben adaptar sus procesos software a unas restricciones, aproximaciones y tecnologías de desarrollo siempre cambiantes.

Estos ingenieros de software junior (en adelante, participantes) eran estudiantes universitarios en su último curso con una amplia formación previa en métodos y herramientas de ingeniería del software, habiendo tenido ya experiencias previas relacionadas con el desarrollo de software. Estos participantes se consideran representativos para el experimento puesto que su educación, experiencia previa y entrenamiento son similares a los profesionales de la industria que empiezan a formalizar procesos de desarrollo de software. Es más, estos participantes tienen un perfil similar al personal que trabaja en organizaciones TIC de innovación.

El experimento fue realizado en dos fases, tal y como se muestra en la Figura 5.1, presentando los participantes de ambas fases una formación académica y experiencia en desarrollo de software similares.

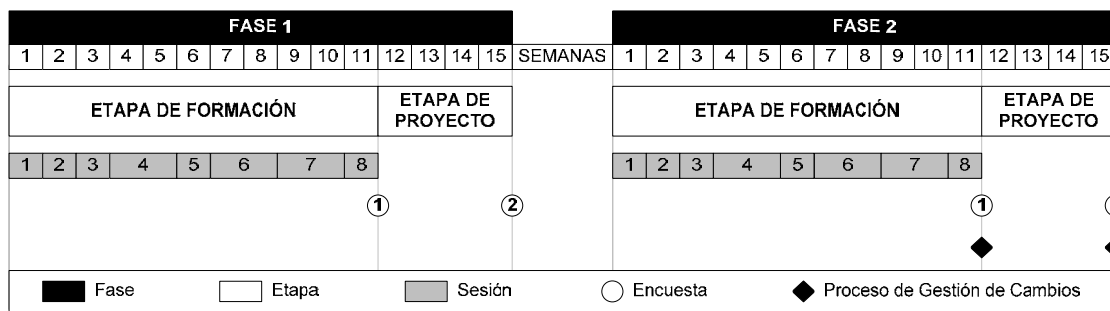


Figura 5.1. Escala de tiempos del experimento.

### Fase 1: PAL-Wiki sin gestión del cambio ni gestión de la configuración (Septiembre 2009 – Enero 2010)

En esta fase, los participantes utilizaron la PAL-Wiki presentada en [Amescua, 2010]. Las funcionalidades que ofrecía esta wiki eran: modificación de activos de conocimiento pre-existentes en la wiki por medio de formularios, registro de ejemplos en forma de ficheros con los artefactos producidos en los proyectos, y participación en discusiones proporcionadas por la propia wiki. No existía ningún mecanismo de realimentación para enriquecer el conocimiento organizacional.

### Fase 2: ITAKA con gestión del cambio y gestión de la configuración (Septiembre 2010 – Enero 2011)

En esta fase, los participantes utilizaron la solución propuesta en esta tesis doctoral. En lugar de la PAL-Wiki utilizada en la fase anterior, en este caso se desplegaron tres wikis para proporcionar a los participantes el acceso a un mismo conocimiento organizacional. Los participantes fueron distribuidos en equipos de trabajo, quedando agregados en unidades aquellos equipos que debían trabajar más estrechamente. Así se formaron tres unidades de trabajo, asignando a cada unidad una de las tres wikis desplegadas en esta fase.

El despliegue de tres wikis distintas era necesario para poder evaluar la efectividad de los distintos mecanismos de realimentación que permiten incorporar activos nuevos o mejorados en el repositorio de conocimiento organizacional. Aunque todas ellas permitían a los participantes acceder al mismo conocimiento organizacional, cada una ofrecía una manera diferente de gestionar los cambios propuestos por los ingenieros de software durante el proceso de gestión de cambios. Así, las wikis desplegadas fueron:

- a) Wiki con gestión de cambios supervisada (en adelante, “wiki supervisada”): sus usuarios podían realizar cambios en el conocimiento pre-existente así como introducir nuevo conocimiento tácito. Tanto los cambios como el nuevo conocimiento eran supervisados por los instructores, los cuales decidían si dichas aportaciones debían ser añadidas al repositorio de conocimiento organizacional o no.
- b) Wiki con gestión de cambios no-supervisada (en adelante, “wiki no-supervisada”): sus usuarios podían realizar cambios en el conocimiento pre-existente así como introducir nuevo conocimiento tácito. Ni los cambios ni el nuevo conocimiento eran supervisados por los instructores, por lo que todas las aportaciones de los usuarios eran siempre añadidas al repositorio de conocimiento organizacional.
- c) Wiki de solo-lectura: sus usuarios no podían cambiar directamente el conocimiento pre-existente ni introducir nuevo conocimiento tácito. Cuando un usuario quería realizar alguna aportación al repositorio de conocimiento organizacional tenía que ponerse en contacto con los instructores (por correo electrónico, por ejemplo), los cuales se encargaban de realizar dicho cambio en el conocimiento organizacional. Este mecanismo es el utilizado habitualmente para gestionar el conocimiento almacenado en una PAL.

Los mecanismos de realimentación mencionados arriba permitían en esta segunda fase la recogida de todo el nuevo conocimiento adquirido por medio de las tres wikis y su almacenamiento en un mismo repositorio para enriquecer el conocimiento organizacional.

### 5.3. Planificación

La ejecución de cada una de las dos fases del experimento se dividió en tres etapas, tal y como se muestra en la Figura 5.2.

#### Etapa de configuración (1)

La primera etapa de cada fase del experimento consistía en la configuración del conocimiento inicial disponible para los participantes y su consecuente publicación en forma de wiki. En ambas fases, el contenido inicial de la wiki era el mismo. Las diferencias entre las wikis usadas en cada fase residían principalmente en las funcionalidades ofrecidas para mejorar el conocimiento pre-existente o para añadir conocimiento tácito en forma de discusiones o ejemplos. En la fase 1, estas funcionalidades se basaban en la propuesta PAL-Wiki [Amescua, 2010], mientras que en la fase 2 se ofrecían las funcionalidades descritas en el Capítulo 3: Solución Propuesta.

Esta primera etapa se corresponde con el Proceso 1: Identificación del Conocimiento, el Proceso 2: Formalización del Conocimiento y el Proceso 3: Distribución del Conocimiento descritos anteriormente. Los encargados de realizar el trabajo correspondiente a esta etapa fueron los propios instructores del curso de formación.

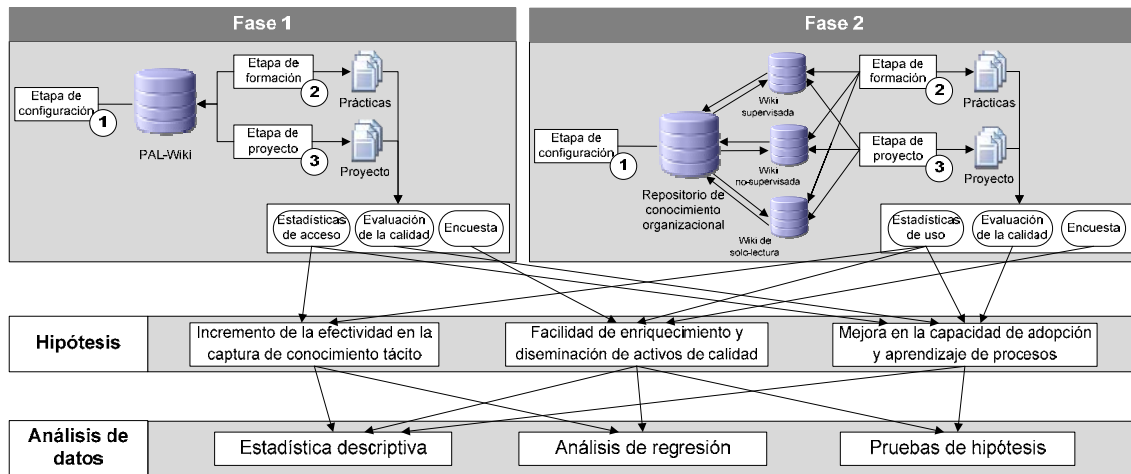


Figura 5.2. Etapas de ejecución del experimento.

### Etapa de formación (2)

La segunda etapa consistía en el entrenamiento de los participantes acerca de las distintas técnicas de desarrollo ágil de software. En ambas fases, la duración de la etapa de formación fue de 11 semanas y se dividió en 8 sesiones: a) Historias de usuario y Pruebas de aceptación, b) Planificación adaptativa, c) Estándares de codificación, d) Diseño simple, e) Reorganización de código, f) Pruebas unitarias, g) Propiedad colectiva de código e Integración continua, y finalmente h) Seguimiento del proyecto. Durante esta etapa de formación, los participantes se fueron familiarizando con la PAL, explorando sus contenidos para llevar a cabo las prácticas correspondientes al curso de formación.

En la fase 1, la PAL-Wiki y sus funcionalidades (modificación del conocimiento pre-existente, participación en discusiones y publicación de ejemplos) fueron presentadas durante la primera semana de esta etapa de formación. En la fase 2, ITAKA fue presentada de manera análoga durante la tercera semana de esta etapa de formación. En ambas fases, periódicamente se recordaba a los participantes cómo utilizar la PAL y se les hacía ver la importancia de compartir el conocimiento tácito generado para tratar de mejorar el aprendizaje y la adopción del desarrollo ágil de software. Sin embargo, en ningún momento se les proporcionó ninguna recompensa adicional por la utilización de la PAL.

En ambas fases la PAL fue utilizada para recoger conocimiento tácito que fomentaba la adopción del proceso de desarrollo ágil de software considerado en este curso de formación.

Una vez completadas las 8 sesiones de entrenamiento, los participantes realizaron una encuesta en la que se recogían datos acerca de la utilización de la PAL usada para dar soporte al proceso de aprendizaje (PAL-Wiki en la fase 1 e ITAKA en la fase 2). Esta encuesta constaba de 5 preguntas cerradas acerca del uso de la wiki, 8 preguntas de opinión utilizando una escala Likert de 4 puntos, y 27 elementos de diferencial semántico utilizando una escala Likert de 5

puntos. Las preguntas cerradas preguntaban acerca del número de accesos y de contribuciones realizadas por el encuestado. Las preguntas de opinión tenían como objetivo recoger datos acerca de la facilidad en el uso de los mecanismos de adquisición de nuevo conocimiento y acerca de la utilidad de dicho conocimiento. Los elementos de diferencial semántico estaban destinados a medir el nivel de utilidad y el nivel de calidad de los distintos artefactos. Por último, las preguntas abiertas pretendían recoger datos acerca de los beneficios obtenidos y de los problemas encontrados por los encuestados con el uso del sistema, así como recoger sugerencias acerca de posibles mejoras.

### Etapa de proyecto (3)

La última etapa del experimento consistía en el desarrollo de una pequeña aplicación software (Figura 5.3). En ambas fases, esta etapa de proyecto comenzó en la semana 12 y duró 4 semanas.

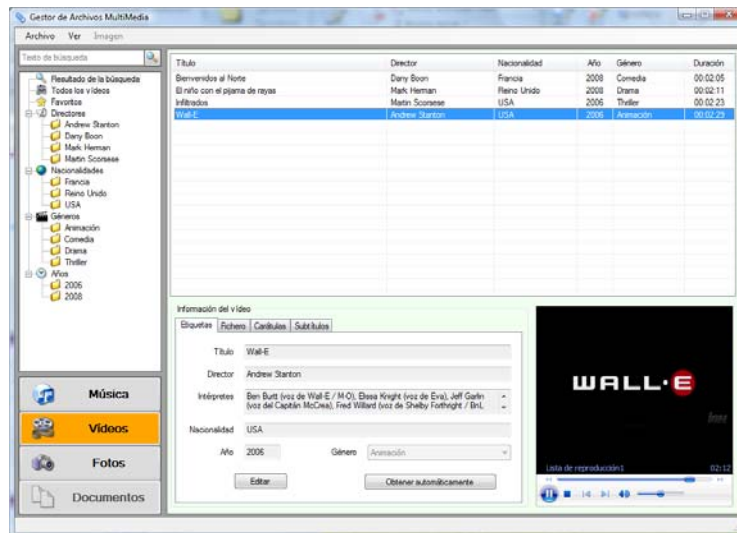


Figura 5.3. Aplicación software desarrollada durante el experimento.

El proyecto a desarrollar en ambas fases consistía en un gestor de ficheros multimedia. Durante la fase 1, cada equipo de trabajo tuvo que desarrollar las funcionalidades necesarias para gestionar vídeos, incluyendo gestión de ficheros de vídeo, catalogación de vídeos, búsqueda de vídeos, gestión avanzada de información de vídeos (incluyendo metadatos y carátulas), reproducción de vídeos y gestión de subtítulos (incluyendo edición de subtítulos y la posibilidad de mostrarlos durante la reproducción del vídeo). Durante la fase 2, cada equipo de trabajo tuvo que desarrollar las funcionalidades necesarias para gestionar fotos, incluyendo gestión de ficheros de fotos, catalogación de fotos, búsqueda de fotos, gestión de metadatos de fotos, algunas funcionalidades sencillas de procesamiento de fotos (rotar, reflejar, redimensionar, aplicar filtros sencillos), y detección de fotos duplicadas utilizando un algoritmo basado en los colores que conforman cada una de las fotos.

Durante esta etapa, el Proceso 4: Aplicación del Conocimiento y el Proceso 5: Preservación del Conocimiento fueron realizados por los participantes utilizando los mecanismos propuestos para la mejora del conocimiento pre-existente, la participación en discusiones y el registro de ejemplos. Las diferencias entre la fase 1 y la fase 2 residen en las mejoras presentadas en el

Capítulo 3: Solución Propuesta debidas a los principios de gestión del cambio y de gestión de la configuración introducidos en esta tesis doctoral.

Además, al principio de esta etapa de proyecto durante la fase 2, se desplegó una nueva versión de cada una de las tres wikis. Esta tarea se corresponde con el Proceso 6: Gestión de Cambios. Estas nuevas versiones contenían el conocimiento almacenado en el repositorio de la organización al comienzo de la etapa de proyecto en la fase 2, esto es, el conocimiento pre-existente del comienzo de la etapa de formación más el nuevo conocimiento adquirido durante el entrenamiento realizado en dicha etapa. De esta forma, las tres wikis volvieron de nuevo a ofrecer el mismo conocimiento organizacional.

Al final de esta etapa, los participantes realizaron una segunda encuesta acerca de las capacidades de la PAL (PAL-Wiki en la fase 1 e ITAKA en la fase 2) para proporcionar información durante la etapa de proyecto. Esta encuesta constaba de 5 preguntas cerradas acerca del uso de la wiki, 12 preguntas de opinión utilizando una escala Likert de 4 puntos, 30 elementos de diferencial semántico utilizando una escala Likert de 5 puntos, y 4 preguntas abiertas. Las preguntas cerradas preguntaban acerca del número de accesos y de contribuciones realizadas por el encuestado. Las preguntas de opinión tenían como objetivo recoger datos acerca de la facilidad en el uso de los mecanismos de adquisición de nuevo conocimiento y acerca de la utilidad de dicho conocimiento. Los elementos de diferencial semántico estaban destinados a medir el nivel de utilidad y el nivel de calidad de los distintos artefactos. Por último, las preguntas abiertas pretendían recoger datos acerca de los beneficios obtenidos y de los problemas encontrados por los encuestados con el uso del sistema, así como recoger sugerencias acerca de posibles mejoras.

#### **5.4. Recogida de datos y métodos de análisis**

En este apartado se indican los datos recogidos en relación con cada una de las preguntas de investigación formuladas al comienzo del presente capítulo, así como los métodos utilizados para el análisis de dichos datos. Los datos recogidos incluyen datos de acceso y uso de las distintas wikis utilizadas a lo largo de las dos fases en que estuvo dividido el experimento, datos acerca de las calificaciones obtenidas por los participantes en el curso de formación y aprendizaje de procesos ágiles de desarrollo de software en el cual se realizó el experimento, y datos obtenidos a partir de las encuestas y entrevistas realizadas a los participantes, así como las observaciones anotadas por los instructores del curso de formación. El análisis de los datos obtenidos fue llevado a cabo utilizando una combinación de estadística descriptiva, análisis de regresiones y pruebas de hipótesis (Figura 5.2). Este análisis permite realizar una discusión de los resultados para cada una de las cuestiones formuladas para guiar el experimento descrito al comienzo de este capítulo.

##### **5.4.1. PI1: Efectividad de ITAKA para capturar el conocimiento tácito**

La efectividad de los mecanismos de realimentación representa su capacidad para capturar el conocimiento tácito de los distintos proyectos, enriqueciendo de esta manera el conocimiento de la organización. Partiendo de esto, para abordar la discusión de la pregunta de investigación 1, los datos recogidos se centran en estadísticas acerca de los accesos a las wikis y el número

de contribuciones de cada tipo (modificación del conocimiento pre-existente, ejemplos, discusiones y lecciones aprendidas), así como información cualitativa obtenida a partir de las anotaciones de los instructores, observaciones y entrevistas informales con los participantes.

Para discutir acerca de la efectividad del mecanismo de realimentación en relación con esta pregunta de investigación 1, las estadísticas de acceso a la PAL-Wiki de la fase 1 y las estadísticas de uso de las tres wikis de la fase 2 fueron analizadas, así como los datos obtenidos a partir de las encuestas realizadas durante el experimento. El análisis de estos datos se llevó a cabo utilizando estadística descriptiva.

#### **5.4.2. PI2: Factores que facilitan el uso de ITAKA como fuente de conocimiento relevante**

El objetivo de esta pregunta de investigación 2 es estudiar cuáles son los factores que facilitan el uso de ITAKA como fuente de conocimiento relevante para los ingenieros de software para el aprendizaje y adopción de un nuevo proceso software. Para ello, tan solo es necesario tomar datos en la fase 2, que es cuando se introdujo ITAKA. En concreto, los datos considerados fueron: estadísticas acerca de los accesos a las wikis, el número de contribuciones de cada tipo (modificación del conocimiento pre-existente, ejemplos, discusiones y lecciones aprendidas), información cualitativa obtenida a partir de las anotaciones de los instructores, observaciones y entrevistas informales con los participantes, y, por último, una evaluación subjetiva por parte de los participantes obtenida por medio de encuestas realizadas al final de las etapas de formación y de proyecto durante las dos fases del experimento.

Para discutir acerca de la efectividad del mecanismo de realimentación en relación con esta pregunta de investigación 2, las estadísticas de acceso a la PAL-Wiki de la fase 1 y las estadísticas de uso de las tres wikis de la fase 2 fueron analizadas, así como los datos obtenidos a partir de las encuestas realizadas durante el experimento. El análisis de estos datos se llevó a cabo utilizando varios análisis de regresiones.

#### **5.4.3. PI3: Nivel de calidad del nuevo conocimiento elicitado al utilizar ITAKA**

En relación con la pregunta de investigación 3 se tuvieron en cuenta dos puntos de vista: calidad percibida por los instructores y calidad percibida por los participantes. Para analizar los niveles de calidad percibida por los instructores se utilizaron datos acerca del número de contribuciones aceptadas en relación con el número total de contribuciones realizadas por los ingenieros de software. En cuanto a los niveles de calidad percibida por los participantes, su análisis se realizó en base a los datos recogidos en las encuestas realizadas al final de las etapas de formación y de proyecto durante las dos fases del experimento, las cuales ya han sido comentadas anteriormente.

Para discutir acerca de la calidad de los activos de conocimiento elicitados durante la ejecución de la iniciativa de aprendizaje en relación con esta pregunta de investigación 3, se analizaron los distintos tipos de contribuciones realizadas durante el experimento, el número de contribuciones aceptadas en relación con el número de contribuciones realizadas de cada tipo,

y los datos obtenidos a partir de las encuestas realizadas durante la fase 2 del experimento. Para el análisis de estos datos se utilizó estadística descriptiva y pruebas de hipótesis.

#### **5.4.4. PI4: Factores que influyen en la calidad de los activos de conocimiento**

Los datos recogidos en relación con la pregunta de investigación 4 se centran en estadísticas acerca del número de contribuciones de cada tipo (modificación del conocimiento pre-existente, ejemplos, discusiones y lecciones aprendidas) realizadas durante la fase 2, y en la evaluación subjetiva realizada por parte de los participantes y que fue obtenida por medio de las encuestas durante la fase 2 del experimento.

Para discutir acerca de la calidad de los activos de conocimiento elicitados durante la ejecución de la iniciativa de aprendizaje en relación con esta pregunta de investigación 4, se analizaron los distintos tipos de contribuciones realizadas durante el experimento, el número de contribuciones aceptadas en relación con el número de contribuciones realizadas de cada tipo, y los datos obtenidos a partir de las encuestas realizadas durante la fase 2 del experimento, además de las observaciones anotadas por los instructores. Para el análisis de estos datos se utilizaron varios análisis de regresiones.

#### **5.4.5. PI5: Efectividad de las distintas alternativas (supervisada y no-supervisada)**

En cuanto a la pregunta de investigación 5, los datos considerados son los obtenidos mediante las encuestas realizadas en la fase 2 del experimento, siendo estos datos analizados para cada tipo de realimentación (supervisada y no-supervisada) y para cada tipo de contribución (modificación del conocimiento pre-existente, ejemplos, discusiones y lecciones aprendidas) realizada durante la fase 2 del experimento, ya que es cuando se introdujeron dichos mecanismos de realimentación.

Para discutir acerca de la calidad de los activos de conocimiento elicitados durante la ejecución de la iniciativa de aprendizaje en relación con esta pregunta de investigación 5, se analizaron los distintos tipos de contribuciones realizadas durante el experimento, el número de contribuciones aceptadas en relación con el número de contribuciones realizadas de cada tipo, y los datos obtenidos a partir de las encuestas realizadas durante la fase 2 del experimento. Para el análisis de estos datos se utilizaron pruebas de hipótesis.

#### **5.4.6. PI6: Contribución de ITAKA a la mejora de la facilidad del aprendizaje de procesos**

El aprendizaje de nuevos procesos o de prácticas eficientes de desarrollo de software viene dado por la capacidad de adoptar estos nuevos procesos o prácticas. Analizar el aprendizaje de los procesos permite determinar si ITAKA proporciona un entorno efectivo de aprendizaje para el desarrollo de productos software de calidad. Teniendo esto en cuenta, los datos que se recogieron para su análisis en relación con la pregunta de investigación 6 se centraron en la calidad de los productos desarrollados durante las dos fases del experimento. Para determinar la calidad de dichos productos se aplicó un proceso de evaluación multi-fuente por el cual dos



expertos evaluaron de manera individual los mismos productos utilizando unos criterios comunes pre-establecidos (ver Anexo A: Criterios para la Evaluación de Productos).

Para discutir acerca de la mejora en el aprendizaje de procesos de desarrollo de software en relación con la pregunta de investigación 6 se analizaron las estadísticas de acceso a la PAL-Wiki de la fase 1 y las estadísticas de uso de las tres wikis de la fase 2, además de las calificaciones obtenidas por los participantes en base a la calidad de los productos desarrollados y de los datos recogidos mediante las encuestas en relación con la utilidad percibida respecto a cada tipo de contribución. El análisis de estos datos se realizó utilizando pruebas de hipótesis.

#### **5.4.7. PI7: Evaluación subjetiva de la utilidad de los mecanismos de gestión de la configuración propuestos**

Finalmente, en relación con la pregunta de investigación 7, los datos considerados para el análisis y posterior discusión son los recogidos mediante las encuestas a lo largo de todo el experimento en relación con la utilidad percibida respecto a cada tipo de contribución (modificación del conocimiento pre-existente, ejemplos, discusiones y lecciones aprendidas).

Para discutir acerca de la mejora en el aprendizaje de procesos de desarrollo de software en relación con la pregunta de investigación 7 se analizaron las estadísticas de acceso a la PAL-Wiki de la fase 1 y las estadísticas de uso de las tres wikis de la fase 2, además de las calificaciones obtenidas por los participantes en base a la calidad de los productos desarrollados y de los datos recogidos mediante las encuestas en relación con la utilidad percibida respecto a cada tipo de contribución. El análisis de estos datos se realizó utilizando estadística descriptiva y pruebas de hipótesis.

### **5.5. Resultados**

En este apartado se presentan los resultados obtenidos a partir del análisis de los datos recogidos en relación con cada una de las preguntas de investigación formuladas al comienzo del presente capítulo.

#### **5.5.1. PI1: Efectividad de ITAKA para capturar el conocimiento tácito**

Con el objetivo de determinar la mejora de la efectividad proporcionada por ITAKA en relación con una PAL sin soporte para la gestión de la configuración ni para la gestión del cambio, en primer lugar se analizó el número de contribuciones, es decir, los cambios realizados en el conocimiento pre-existente, discusiones y ejemplos aportados por los participantes en las dos fases del experimento (Tabla 5.2). La tabla también muestra información adicional como es el número de artefactos producidos y el número de activos mejorados debido a las nuevas contribuciones realizadas por los participantes en cada una de las dos fases del experimento.

Tabla 5.2. Número total de contribuciones.

	Fase 1 (28 participantes)			Fase 2 (18 participantes)		
	Etapa de formación (11 sem.)	Etapa de proyecto (4 sem.)	Total (15 sem.)	Etapa de formación (11 sem.)	Etapa de proyecto (4 sem.)	Total (15 sem.)
Artefactos	168	156	324	112	106	218
Ejemplos	0	0	0	130	126	256
Discusiones	0	0	0	27	51	78
Lecciones Aprendidas	0	0	0	22	46	68
Activos actualizados	8	8	8	12	10	13

El número de artefactos por participante en ambas fases fue aproximadamente el mismo puesto que en ambas fases los participantes tenían que realizar el mismo trabajo. Sin embargo, los artefactos obtenidos en la fase 1 eran almacenados en un repositorio externo y además los artefactos aportados por un participante no estaban disponibles para el resto de participantes. Sin embargo, los artefactos obtenidos en la fase 2 fueron transformados en ejemplos y se almacenaron en el mismo repositorio que el conocimiento pre-existente de forma que todos los ejemplos aportados por un participante estaban disponibles para el resto de participantes.

En el caso de las discusiones, en la fase 1 ningún participante inició ninguna discusión, mientras que en la fase 2 hubo 78 participaciones en distintas discusiones. El motivo de este incremento en el número de discusiones en la fase 2 se debió a que los instructores animaron a los participantes a la intervención en las discusiones con el objetivo de cambiar su mentalidad hacia un uso activo y colaborativo de las nuevas características proporcionadas por la PAL en la fase 2, así como conseguir que se publicaran en la wiki las dudas relacionadas con la materia del curso en forma de discusiones y que se acostumbraran también a participar en las discusiones iniciadas por otros compañeros.

Además, al final de cada etapa en la fase 2, algunas de estas discusiones fueron almacenadas en el repositorio de conocimiento organizacional en forma de lecciones aprendidas. Como resultado, el nuevo mecanismo de realimentación permitió la obtención de 68 lecciones aprendidas en la fase 2, mientras que en la fase 1 no se obtuvo ninguna lección aprendida al no haber ninguna discusión.

Otro punto interesante es que los participantes no realizaron ningún cambio (ni en la fase 1 ni en la fase 2) en el conocimiento pre-existente proporcionado por los instructores como conocimiento inicial. La información obtenida por los instructores a través de entrevistas individuales y de la observación sugiere que este hecho se debió a que los participantes no se consideraban lo suficientemente experimentados en la materia como para realizar cambios en los contenidos. Los participantes consideraron que la contribución de nuevos activos de conocimiento en forma de ejemplos y discusiones era más efectiva para mejorar en su aprendizaje y en la adopción de las nuevas prácticas para el desarrollo de software. En cierto modo esto contradice el uso esperado de una wiki, en la cual se espera que los usuarios cuestionen y modifiquen los contenidos existentes, pero refuerza la importancia y la utilidad de la wiki (mejorada con mecanismos de gestión de la configuración y de gestión del cambio)

como una herramienta efectiva para agregar activos de conocimiento relevantes para el aprendizaje de procesos de desarrollo de software, además de conocimiento tácito específico acerca de cómo adaptar dichos procesos a las condiciones y restricciones de un determinado proyecto de software.

Como resultado de todas las contribuciones realizadas por los participantes al conocimiento organizacional, algunos activos fueron mejorados con nuevo conocimiento. Con el uso del nuevo mecanismo de realimentación se obtuvo un aumento del número de activos mejorados en la fase 2 con respecto a la fase 1, principalmente debido al aumento en el número de discusiones y en las correspondientes lecciones aprendidas.

Los datos también indican que los participantes necesitaron de un esfuerzo menor para utilizar las características de la PAL en la fase 2 en relación con las características de la PAL en la fase 1. Sus visitas a las wikis fueron más breves y necesitaron acceder a menos páginas en cada visita. Sin embargo, se produjo un aumento en el número de visitas por participante, lo cual revela un uso más activo de las wikis. Estos datos serán analizados más en detalle en el siguiente apartado.

Es importante destacar por último que un análisis individualizado del número y naturaleza de las contribuciones (ejemplos, discusiones y lecciones aprendidas) y de sus autores indica que la efectividad de ITAKA puede verse afectada positiva o negativamente por la actitud de los participantes en relación con las prácticas de compartición de conocimiento introducidas en esta tesis doctoral. En la fase 2, aquellos usuarios que participaron más activamente en las actividades de compartición de conocimiento durante la etapa de formación realizaron menos aportaciones durante la etapa de proyecto; esto pudo deberse a que su mayor compromiso hizo que sus expectativas fueran mayores, las cuales sin embargo no se vieron cumplidas debido a la baja participación del resto de participantes. Por el contrario, aquellos usuarios que no participaron tan activamente en la etapa de formación mostraron un mayor compromiso en la etapa de proyecto; esto pudo deberse a que las aportaciones de otros participantes más comprometidos les hicieron ver el valor añadido que aportaba el nuevo conocimiento disponible en la PAL.

Como conclusión, se puede afirmar que una PAL mejorada con mecanismos de gestión de la configuración y de gestión del cambio es efectiva en el sentido en que permite a los ingenieros de software aportar nuevo conocimiento a la organización en forma de ejemplos y de discusiones (las cuales se transforman en lecciones aprendidas). Los mecanismos propuestos para gestionar el conocimiento organizacional son efectivos en la captura, formalización y distribución del conocimiento tácito que surge de las interacciones entre los miembros de una organización de software, permitiendo que el nuevo conocimiento proveniente de varias fuentes pueda ser difundido a otros equipos de trabajo de distintas maneras en función de las necesidades de cada equipo.

### 5.5.2. PI2: Factores que facilitan el uso de ITAKA como fuente de conocimiento relevante

La Tabla 5.3 presenta algunos datos recogidos durante la fase 2 en relación con la pregunta de investigación 2. Las semanas 1 y 2 no figuran en la tabla puesto que los nuevos mecanismos no estuvieron completamente operativos hasta la tercera semana del experimento, por lo que las estadísticas acerca de su uso no se pudieron recoger hasta la semana 3. Por otro lado, la semana 16 representa el resultado del proceso de gestión de cambios llevada a cabo por los instructores después de finalizar la etapa de proyecto.

**Tabla 5.3. Estadísticas acerca de las nuevas contribuciones y del uso de las wikis en la fase 2.**

Semana	Ejemplos	Discusiones	Lecciones Aprendidas	Visitas	Tiempo medio (segundos)	Nº págs. visitadas
3	0	0	0	4	38	16
4	52	0	0	17	59	78
5	13	0	0	10	37	31
6	13	2	0	19	497	278
7	0	0	0	13	692	87
8	13	1	0	15	470	112
9	0	7	0	38	417	451
10	26	14	0	76	645	572
11	13	3	0	42	744	1412
12	0	0	22	90	567	1905
13	0	24	0	66	270	447
14	0	22	0	118	532	1252
15	126	5	0	42	744	420
16	0	0	46	-	-	-

La segunda semana de la tabla muestra el número de ejemplos aportados por los participantes durante la fase 2. Como se puede ver, no aparecen aportaciones en forma de ejemplos en todas las semanas ya que estos ejemplos se crearon a partir de los artefactos producidos durante las 8 sesiones de la etapa de formación y a partir de los artefactos producidos durante la etapa de proyecto, lo cual implica que este tipo de aportaciones solo se realizó al finalizar cada una de las sesiones de formación y al finalizar la etapa de proyecto. Además, puesto que el sistema no estuvo completamente operativo hasta la semana 3, los ejemplos creados durante estas primeras semanas no pudieron ser subidos hasta la semana 4.

Teniendo en cuenta el escenario descrito antes, la aproximación cuantitativa no permite encontrar regresiones válidas que ayuden a determinar los factores que influyen en la creación de nuevos ejemplos. Sin embargo, las observaciones y las entrevistas informales con los participantes indicaron que el aumento del número de visitas entre las semanas 12 y 15 se debió parcialmente al número de ejemplos acumulados en el repositorio organizacional. Durante la etapa de formación (semanas 1 a 11), los participantes aportaron gradualmente ejemplos al repositorio al finalizar las distintas sesiones. Puesto que los ejemplos de una sesión no eran necesarios para las siguientes sesiones de formación, dichos ejemplos no fueron necesarios de nuevo hasta la etapa de proyecto (semanas 12 a 15). En dichas semanas, los participantes accedieron al nuevo conocimiento recogido en forma de ejemplos durante la etapa de formación para adaptarlo a sus respectivos proyectos, incrementando así el número

de visitas a las wikis. El número de visitas decreció finalmente en la semana 15 puesto que muchos de los participantes ya habían finalizado sus proyectos en dicha semana.

En cuanto a las discusiones, se encontró una regresión múltiple ( $R^2 = 0.7727$ ,  $p$ -valor = 0.0030) entre las variables *número de visitas a las wikis* (V), *tiempo medio empleado en las wikis* (T) y *número de páginas visitadas en las wikis* (P), que actúan como predictores del *número de nuevas discusiones en las wikis* (D). La ecuación de regresión obtenida es la siguiente:

$$D = -1.292172 + 0.321763 \cdot V - 0.000543 \cdot T - 0.011198 \cdot P \quad (E1)$$

Esta ecuación indica que el número de nuevas discusiones se ve incrementado con el uso intensivo de las wikis, pero solo si los activos de conocimiento son sencillos de localizar y si los mecanismos de participación en las discusiones son fáciles de manejar. Estos valores numéricos junto con la información extraída de las entrevistas y de la observación indican que las discusiones fueron un mecanismo útil para los participantes a la hora de encontrar experiencias previas acerca de cómo adaptar las prácticas de desarrollo de software ágiles ya existentes a los nuevos proyectos (cuantas más discusiones hay, menos tiempo se pierde en las wikis porque se necesita menos esfuerzo para obtener la información que se necesita).

Un análisis más detallado de la ecuación E1 muestra:

- Una relación positiva entre el número de nuevas discusiones y el número de visitas en las wikis. Esto indica que cuantos más usuarios acceden a las wikis, más interacciones se producen entre ellos.
- Una relación negativa entre el número de nuevas discusiones y el tiempo medio empleado en las wikis. Esto indica que cuanto más tiempo emplean los usuarios en las wikis, menos están centrados en localizar conocimiento útil. Cuando los usuarios están centrados y comprometidos, la tendencia a enriquecer el conocimiento aumenta. Además, cuando un usuario crea o participa en una discusión, el resto de usuarios de esa wiki recibe una notificación informando acerca de la discusión, de manera que si otro usuario quiere participar en esa discusión no necesita perder el tiempo buscándola en la wiki. Así, si los usuarios pierden mucho tiempo en las wikis es debido a que están buscando información acerca de los procesos ágiles de desarrollo de software pero no están realizando aportaciones a las discusiones.

Partiendo de esto y teniendo en cuenta que el número de discusiones aumentó significativamente en la fase 2 con respecto a la fase 1, cabe esperar un descenso en la duración de las visitas en la fase 2. Para verificar esto se compararon los datos recogidos en ambas fases y los resultados se presentan en la Figura 5.4, donde los datos se han normalizado dividiendo entre el número de participantes en cada fase. La figura muestra un incremento general en el número de visitas por usuario en la fase 2. Esto indica que las wikis fueron usadas como referencia del conocimiento más en la fase 2 que en la fase 1. La figura también muestra un aumento significativo en el número de visitas por usuario que dedicaron menos de 30 segundos en las wikis. Esto es, la duración de las visitas en la fase 2 fue menor que en la fase 1, algo esperado como ya se mencionó antes.

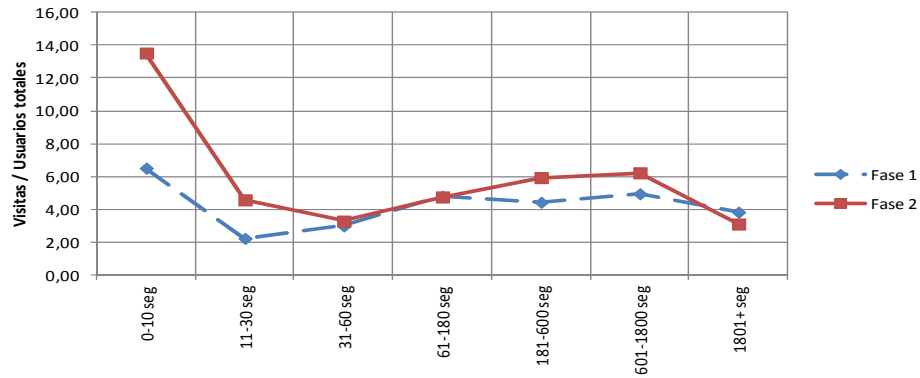


Figura 5.4. Número de visitas por usuario vs. Duración de las visitas.

- Una relación negativa entre el número de nuevas discusiones y el número de páginas visitadas en las wikis. Esto indica que cuantas más páginas visitan los usuarios, menos están centrados en localizar conocimiento útil. Puesto que los usuarios reciben notificaciones con enlaces directos a las nuevas aportaciones a una discusión, si quieren participar en dicha discusión solo tienen que pulsar en el enlace para acceder directamente a la discusión si la necesidad de acceder a través de otras páginas de la wiki. Así, al igual que en el punto anterior, si los usuarios visitan un gran número de páginas es más verosímil que se deba a que están consultando información acerca de los procesos ágiles de desarrollo de software pero no a que estén realizando aportaciones a las discusiones.

Partiendo de esto y teniendo en cuenta que el número de discusiones aumentó significativamente en la fase 2 con respecto a la fase 1, cabe esperar un descenso en el número de páginas visitadas en la fase 2. La Figura 5.5 muestra otra comparativa de los datos recogidos en ambas fases, estando los datos normalizados dividiendo entre el número de participantes en cada fase. La figura vuelve a mostrar un incremento general en el número de visitas por usuario en la fase 2, así como un aumento significativo en el número de visitas por usuario que accedieron a menos de 4 páginas en la visita. Es decir, la cantidad de páginas visitadas en cada visita fue menor en la fase 2 que en la fase 1, lo cual apoya las conclusiones obtenidas a partir de la Figura 5.4.

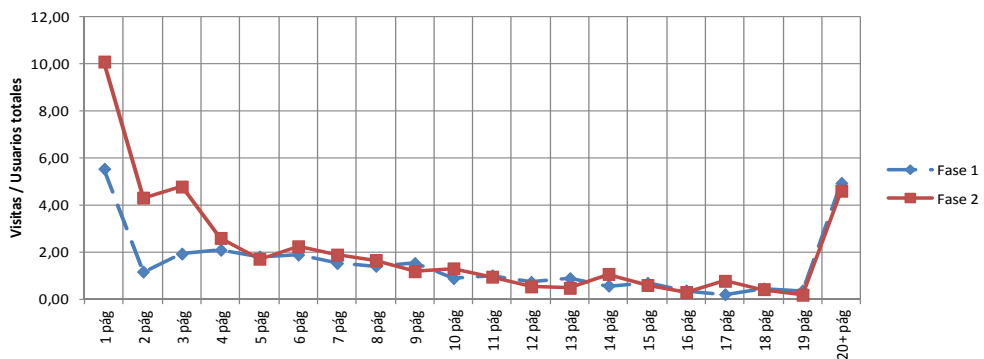


Figura 5.5. Número de visitas por usuario vs. Páginas vistas durante las visitas.

Con el objetivo de analizar más en detalle la relación entre el *número de nuevas discusiones en las wikis* (D) y las variables *número de visitas a las wikis* (V), *tiempo medio empleado en las wikis* (T) y *número de páginas visitadas en las wikis* (P), los productos cruzados de dichas variables fueron incluidos en el análisis de regresión. A continuación se presenta la nueva regresión múltiple ( $R^2 = 0.9830$ , p-valor = 0.0004) obtenida:

$$D = -1.5170 + 0.3420 \cdot V + 0.00006629 \cdot V \cdot T + 0.001191 \cdot V \cdot P + 0.000103 \cdot T \cdot P - 0.006739 \cdot T - 0.05479 \cdot P - 0.00000246 \cdot V \cdot T \cdot P \quad (E2)$$

La ecuación E2 refuerza la ecuación E1 y añade las siguientes interpretaciones a las ya mencionadas antes:

- Existe una relación positiva entre el número de nuevas discusiones y el producto de las visitas y del tiempo medio en las wikis. Esto indica que los usuarios están participando activamente en las wikis y muestra que la aportación positiva del número de visitas al número de discusiones supera la aportación negativa del tiempo dedicado en las wikis.
- Existe una relación positiva entre el número de nuevas discusiones y el producto de las visitas y de las páginas visitadas en las wikis. Esto indica que cuando los usuarios acceden muy a menudo a las wikis para visitar muchas páginas manteniendo constante el tiempo que dedican en la wiki, los usuarios están participando activamente en las wikis y el número de discusiones aumenta en aquellas páginas que tienen una mayor actividad.
- Existe una relación positiva entre el número de nuevas discusiones y el producto del tiempo medio y de las páginas visitadas en las wikis. A pesar del efecto negativo de ambas variables sobre el número de discusiones, cuando se combinan producen un efecto positivo. Cuanto más tiempo dedican los usuarios y más páginas visitan en las wikis, más centrados están en utilizar las wikis como referencia del conocimiento, lo cual les lleva a participar más en las discusiones.
- Existe una relación negativa entre el número de nuevas discusiones y el producto de las visitas, del tiempo medio y de las páginas visitadas en las wikis. Esto indica que cuando los usuarios acceden muy a menudo a las wiki para visitar un gran número de páginas y dedicando mucho tiempo, es muy verosímil que se deba a que están curioseando por las wikis y buscando información general sobre los procesos ágiles de desarrollo de software y, por lo tanto, que no estén participando en las discusiones.

Finalmente, en relación con las lecciones aprendidas, el análisis estadístico utilizando regresiones no es posible ya que las discusiones no se transformaban y combinaban con el resto de lecciones aprendidas hasta el final de cada etapa de la fase 2 (que es cuando se realizaba el proceso de gestión de cambios), no estando por tanto disponibles hasta la siguiente semana (semanas 12 y 16). Sin embargo, las evidencias obtenidas a partir de las observaciones y de las encuestas sugieren que el aumento del número de visitas en la semana 12 se debió parcialmente a las nuevas lecciones aprendidas que fueron incluidas en el repositorio al final de la etapa de formación. Aunque en dicha semana no hubo nuevas discusiones ni nuevos ejemplos, el número de visitas fue superior al de las semanas anteriores

en parte porque los participantes accedieron a los activos de conocimiento correspondientes para ver las experiencias recogidas por otros participantes durante la etapa de formación.

Resumiendo lo comentado en esta sección, los factores que contribuyen al uso de ITAKA como una fuente de conocimiento relevante son diversos dependiendo de la etapa analizada. Por un lado, durante la etapa de formación el objetivo es aprender nuevos procesos y prácticas ágiles de desarrollo de software. El conocimiento pre-existente y los ejemplos ayudan a los participantes a alcanzar este objetivo, pero los ejemplos tienen el inconveniente de que al principio no son sencillos de elaborar ya que requieren una de-contextualización y una re-contextualización. Además, las discusiones proporcionan una forma sencilla de intercambiar conocimiento entre los participantes para ayudar en el aprendizaje de los nuevos procesos. Por otro lado, durante la etapa de proyecto el objetivo es adaptar los procesos ágiles de desarrollo de software a las restricciones y al contexto del proyecto. Los ejemplos y las lecciones aprendidas ayudan a los participantes a alcanzar este objetivo, pero existe la posibilidad de que haya una falta de diversidad en estos tipos de activos de conocimiento. Todo el conocimiento tácito recogido durante la etapa de formación puede provenir de un mismo contexto (en este caso, del mismo curso de formación), de manera que en la etapa de proyecto que sigue a la formación podría haber muchos ejemplos y lecciones aprendidas pero todos ellos enfocados dentro del mismo contexto. Su diversidad debería aumentar con el uso del sistema puesto que se iría recogiendo conocimiento tácito de otros contextos diferentes con el transcurrir del tiempo.

### 5.5.3. PI3: Nivel de calidad del nuevo conocimiento elicitado al utilizar ITAKA

La discusión de los resultados obtenidos en relación con la calidad de los activos de conocimiento elicitados durante la realización del experimento se ha dividido en dos partes para considerar por un lado la calidad analizada desde el punto de vista de los instructores del curso de formación y por otro lado la calidad analizada desde el punto de vista de los participantes en el experimento.

#### *Calidad del nuevo conocimiento desde el punto de vista de los instructores*

Desde el punto de vista de los instructores, la calidad del nuevo conocimiento se puede medir en términos de cuántas contribuciones son añadidas al repositorio de conocimiento organizacional cuando se utiliza un mecanismo supervisado de gestión de cambios. Aunque las contribuciones de la wiki no-supervisada no fueron revisadas por un experto sino directamente añadidas al repositorio, la calidad de estas contribuciones fue medida para contabilizar el número de contribuciones que hubieran sido consideradas irrelevantes (y por tanto descartadas en caso de haber utilizado el mecanismo supervisado). La Tabla 5.4 presenta el número total de ejemplos y discusiones en cada etapa de la fase 2 y el número de dichas contribuciones que hubiera sido aceptado durante el proceso de gestión de cambios supervisada.



Tabla 5.4. Percepción de la calidad del nuevo conocimiento según los instructores.

			Ejemplos		Discusiones	
			Total	Aceptados	Total	Aceptadas
Fase 2	Etapa de formación	Wiki supervisada	80	80	19	16
		Wiki no-supervisada	50	41	8	6
		Wiki de solo-lectura	30	24	4	4
		Todas las wikis	160	145	31	26
	Etapa de proyecto	Wiki supervisada	76	76	13	13
		Wiki no-supervisada	50	47	41	33
		Wiki de solo-lectura	21	9	0	0
		Todas las wikis	147	132	54	46

Las proporciones entre contribuciones aceptadas respecto de contribuciones totales que se extraen de la tabla anterior indican que la calidad general de las contribuciones fue alta y muy similar en ambas etapas de la fase 2. Sin embargo, la calidad de los ejemplos parece ser mayor que la calidad de las discusiones. Esto podría deberse a que las discusiones se utilizaron principalmente para resolver dudas acerca de los objetivos de la etapa de formación o acerca de los requisitos del proyecto, lo cual en realidad no es relevante para los procesos de la organización y por tanto tampoco lo es para la realimentación. Este problema también es aplicable a las diferencias entre las wikis utilizadas, de tal manera que las discusiones provenientes de la wiki supervisada estuvieron más enfocadas a los procesos de la organización haciendo que la calidad de las contribuciones de dicha wiki fuera superior al resto de wikis.

### ***Calidad del nuevo conocimiento desde el punto de vista de los participantes***

Desde el punto de vista de los participantes, la calidad del nuevo conocimiento se midió a partir de los datos obtenidos a través de las encuestas. Así, los niveles de calidad percibidos por los participantes en relación con el nuevo conocimiento elicitado se muestran en la Tabla 5.5 en una escala de 1 a 5.

Tabla 5.5. Percepción de la calidad del nuevo conocimiento según los participantes.

		Ejemplos		Discusiones		Lecciones Aprendidas	
		Media	DE	Media	DE	Media	DE
Fase 1	Etapa de formación	–	–	–	–	n/a	n/a
	Etapa de proyecto	3.88	1.08	2.08	0.88	n/a	n/a
Fase 2	Etapa de formación	3.75	1.42	2.75	1.14	n/a	n/a
	Etapa de proyecto	4.00	1.04	3.08	1.24	2.92	0.79

La tabla anterior muestra que los ejemplos estuvieron mejor valorados que las discusiones en la fase 2. Esto se debió principalmente a que los ejemplos mostraron a los participantes cómo debían ejecutar los procesos mientras que las discusiones solo se usaron para resolver dudas concretas. Comparando las diferencias entre las etapas de proyecto en ambas fases, las valoraciones parecen ser mejores en la fase 2, especialmente en las discusiones. Por último, destacar que tan solo hubo lecciones aprendidas disponibles en la etapa de proyecto de la fase 2 (obtenidas a partir de las discusiones de la etapa de formación de la fase 2), lo cual impide

comparar su calidad con otra etapa o fase, aunque se observa que la calidad percibida en esa etapa fue aceptable (casi 3 en una escala de 1 a 5).

Para estudiar más en profundidad la evolución de la calidad percibida por los participantes en relación con los ejemplos, discusiones y lecciones aprendidas se realizó una prueba t de Student con un intervalo de confianza del 95%. Esta prueba permite comprobar si dos muestras de datos provienen de la misma distribución (con la misma media y la misma desviación estándar), por lo que puede determinar si existe alguna diferencia entre los niveles de calidad percibidos por los participantes en las dos fases del experimento. Las hipótesis definidas fueron:

*H<sub>0</sub>: niveles percibidos de calidad similares en ambas fases*

*H<sub>1</sub>: niveles percibidos de calidad mayores en la fase 2 que en la fase 1*

En relación con la **calidad de los ejemplos**, los resultados obtenidos en esta prueba de hipótesis no proporcionaron suficientes evidencias para rechazar la hipótesis nula ( $p$ -valor = 0.3703), por lo que se puede afirmar que los participantes no percibieron ninguna mejora en la calidad de los ejemplos al introducir ITAKA en la fase 2.

Sin embargo, es importante recordar las diferencias existentes entre los ejemplos de la fase 1 y los ejemplos de la fase 2. Los ejemplos de la fase 1 eran parte del conocimiento pre-existente proporcionado por los instructores al comienzo de la fase, y los participantes no pudieron crear nuevos ejemplos en el repositorio sino que simplemente subían los artefactos generados a un repositorio externo. En cambio, en la fase 2 los participantes evaluaron no solo los ejemplos correspondientes al conocimiento pre-existente sino también los ejemplos creados por otros participantes.

También cabe destacar que la calidad percibida por los participantes en relación con los ejemplos se vio afectada por la complejidad del proceso de creación de ejemplos. Esta complejidad se debe en primer lugar a que la información contextual debe ser filtrada para obtener material limpio (de-contextualización), y en segundo lugar a que también es necesario incluir información para que el ejemplo tenga suficiente significado como para poder ser reutilizado para el aprendizaje (re-contextualización). A pesar de ello, la calidad percibida respecto a los ejemplos en la fase 2 fue razonablemente alta y creció a medida que los participantes se familiarizaron con este proceso.

En relación con la **calidad de las discusiones**, los resultados obtenidos en la prueba de hipótesis proporcionaron fuertes evidencias para rechazar la hipótesis nula ( $p$ -valor = 0.0116), por lo que se puede afirmar que los participantes percibieron mejora en la calidad de las discusiones al introducir ITAKA en la fase 2. Esta mejora fue debida a que en la fase 2 estuvo más clara la finalidad del uso de las discusiones, a que se indicaron algunas prácticas efectivas para la correcta creación de discusiones, y a que los instructores animaron a los participantes con el objetivo de cambiar su mentalidad hacia un uso activo y colaborativo de ITAKA. En consecuencia se produjo un aumento en las aportaciones de los participantes en forma de discusiones, lo cual llevó a un aumento en la calidad percibida en relación con las mismas.

Sin embargo, ITAKA fue un elemento novedoso para los participantes, por lo que se necesitó de cierto tiempo para que se familiarizaran con el sistema y para lograr una involucración activa en su uso. Este podría ser el principal motivo por el cual la mejoría de la calidad percibida es más evidente en la etapa de proyecto que en la etapa de formación de la fase 2.

El compromiso alcanzado por los participantes a lo largo de la fase 2 en relación con el uso de las discusiones también afectó positivamente a su participación en las wikis haciendo que se contribuyera más y con mejor calidad. Las estadísticas apoyan esta afirmación porque los participantes que contribuyeron más activamente fueron los que aportaron contribuciones de mayor calidad y los que obtuvieron productos de mayor calidad en la etapa de proyecto, mientras que los participantes que no contribuyeron activamente realizaron aportaciones de menor calidad y obtuvieron productos de menor calidad en la etapa de proyecto. Este resultado es consistente con [Radziwill, 2004], donde los autores utilizaron una wiki para mejorar la efectividad de un equipo de desarrollo de software, llegando a la conclusión que su sistema solamente era efectivo cuando todos los miembros del equipo estaban altamente comprometidos, eran responsables y querían cumplir con los requisitos de documentación como un acompañamiento necesario para el desarrollo de código.

Por último, en relación con la **calidad de las lecciones aprendidas**, el análisis cuantitativo no es posible ya que tan solo hubo lecciones aprendidas disponibles en la etapa de proyecto de la fase 2 (obtenidas a partir de las discusiones de la etapa de formación de la fase 2). Sin embargo, el análisis cualitativo llevó a las siguientes apreciaciones:

- Puesto que las lecciones aprendidas provienen de las discusiones de la etapa de formación de la fase 2, estas no están disponibles hasta la etapa de proyecto, lo cual impide que los participantes hagan un uso efectivo de ellas.
- Algunos participantes que realizaron aportaciones a las discusiones de manera activa durante la etapa de formación de la fase 2 evaluaron las lecciones aprendidas menos positivamente, lo cual pudo deberse a que su mayor compromiso les creó grandes expectativas, las cuales sin embargo no se vieron cumplidas debido a la baja participación del resto de participantes.
- Algunos participantes que no realizaron aportaciones a las discusiones de manera activa durante la etapa de formación de la fase 2 descubrieron en la etapa de proyecto el valor añadido que podían aportar las discusiones en forma de lecciones aprendidas, lo cual hizo que valoraran las lecciones aprendidas de manera muy positiva.

#### 5.5.4. PI4: Factores que influyen en la calidad de los activos de conocimiento

Para la discusión acerca de los factores que influyen en la calidad de los nuevos activos de conocimiento creados al utilizar ITAKA, en esta sección se realiza un análisis cuantitativo de los datos recogidos en función de la naturaleza de las contribuciones realizadas (ejemplos, discusiones y lecciones aprendidas).

En el caso de los ejemplos (Tabla 5.6), se encontró una regresión múltiple ( $R^2 = 0.9507$ ,  $p$ -valor = 0.0101) en la que las variables *número total de ejemplos (N)*, *número de ejemplos aceptados (A)*, su *utilidad para el aprendizaje (UA)*, su *utilidad para el proyecto (UP)* y la *sencillez del mecanismo para crear ejemplos (SM)* actúan como predictores de la *calidad de los ejemplos (Qej)*.

Tabla 5.6. Factores relacionados con la calidad de los ejemplos en la etapa de proyecto en la fase 2.

Participante	Qej	N	A	UA	UP	SM
P1	5.00	9	9	4.00	3.75	2.00
P2	4.00	10	10	3.20	2.80	4.00
P3	5.00	10	10	4.00	2.50	n/a
P4	4.00	9	9	3.20	3.55	2.00
P5	4.00	10	10	4.00	2.55	2.00
P6	5.00	9	9	4.00	4.00	2.00
P7	4.00	10	10	3.20	3.55	4.00
P8	4.00	9	9	2.40	3.50	2.00
P9	n/a	10	9	n/a	n/a	3.00
P10	4.00	10	9	2.40	2.60	1.00
P11	4.00	10	10	4.00	n/a	2.00
P12	4.00	10	10	3.20	3.25	1.00
P13	1.00	10	9	0.80	0.95	1.00

$$Qej = -1.85700 + 0.95444 \cdot N + 0.67793 \cdot UA + 0.89276 \cdot UP + 0.01562 \cdot SM - 0.87471 \cdot A \quad (E3)$$

El análisis de la ecuación E3 muestra que:

- El número total de ejemplos es un potenciador de su calidad en el sentido en que, cuantos más ejemplos se crean, más instancias representando situaciones diferentes hay disponibles para los usuarios, lo cual hace aumentar la percepción general acerca de la calidad de los ejemplos.
- La utilidad de los ejemplos para el aprendizaje es también un potenciador de su calidad. Los participantes encontraron que los ejemplos fueron útiles para ayudarles a aprender cómo aplicar su conocimiento para enfrentarse a diferentes problemas en futuros proyectos. Incluso si un usuario no es capaz de encontrar un ejemplo adecuado para resolver un problema concreto, los ejemplos que consulte le pueden ser útiles puesto que le pueden dar indicaciones sobre soluciones similares aplicadas en otros contextos.
- Otro potenciador de la calidad de los ejemplos es su utilidad para el proyecto. Es habitual que se produzcan situaciones problemáticas similares en proyectos distintos, y los ejemplos pueden proporcionar el conocimiento necesario para abordar estos problemas. Así, es esencial describir correctamente el problema que trata de resolver cada ejemplo garantizando su independencia de las características del proyecto. Este es el motivo por el cual el proceso de de-contextualización y re-contextualización es tan importante ya que de esta manera se facilitarán las búsquedas eficientes de ejemplos por parte de los usuarios.

- La sencillez del mecanismo utilizado para aportar nuevos ejemplos es considerado como uno de los elementos básicos para aumentar la calidad de los ejemplos. Los participantes consideraron que los mecanismos actuales para aportar nuevos ejemplos eran laboriosos de utilizar debido a esfuerzo extra requerido para realizar el proceso de de-contextualización y re-contextualización, lo cual conllevó una reducción de la calidad percibida respecto de los ejemplos. La mejora de estos mecanismos podría ayudar a facilitar el proceso y, por tanto, a mejorar la calidad de los ejemplos.
- El número de ejemplos aceptados es un inhibidor de la calidad de los mismos. Esto resalta la relevancia del mecanismo supervisado de gestión de cambios puesto que ayuda a realizar un filtrado de los ejemplos que deberían ser aceptados en la realimentación. Cuando la supervisión es rigurosa, solo los mejores ejemplos pasan a formar parte del repositorio de conocimiento organizacional, aumentando así la calidad general de los ejemplos. Así se podría evitar incluir ejemplos similares a los ya existentes ya que no serían considerados de utilidad por estar repetidos y podrían reducir la calidad general percibida respecto de los ejemplos.

En el caso de las discusiones (Tabla 5.7), se encontró una regresión múltiple ( $R^2 = 0.8149$ ,  $p$ -valor = 0.0448) en la que las variables *número total de discusiones* (N), *su utilidad para el aprendizaje* (UA), *su utilidad para el proyecto* (UP) y *la sencillez del mecanismo para participar en las discusiones* (SM) actúan como predictores de la *calidad de las discusiones* (Qd).

Tabla 5.7. Factores relacionados con la calidad de las discusiones en la etapa de proyecto en la fase 2.

Participante	Qd	N	UA	UP	SM
P1	5.00	0	4.00	3.80	4.00
P2	3.00	0	2.40	2.80	3.00
P3	3.00	0	2.40	2.40	3.00
P4	4.00	0	3.20	n/a	3.00
P5	3.00	0	3.20	2.80	3.00
P6	5.00	8	2.40	2.35	3.00
P7	3.00	1	2.40	2.60	3.00
P8	1.00	0	1.60	2.60	2.00
P9	n/a	1	n/a	n/a	n/a
P10	2.00	0	1.60	1.90	n/a
P11	4.00	10	1.60	2.35	4.00
P12	2.00	10	1.60	2.85	3.00
P13	2.00	0	1.60	1.45	4.00

$$Qd = -0.47970 + 0.19890 \cdot N + 1.88260 \cdot UA + 0.27361 \cdot SM - 0.86165 \cdot UP \quad (E4)$$

El análisis de la ecuación E4 muestra que:

- Las discusiones son un buen mecanismo para fomentar la colaboración entre los miembros de un equipo de trabajo a través de la generación de una base de conocimiento común. Cuantas más aportaciones se hagan a las discusiones, más se trabajará en equipo para superar los restos del proyecto, resultando en una mejor consideración de la calidad de las discusiones.

- A diferencia de los ejemplos, el mecanismo actual para participar en las discusiones es considerado como muy sencillo de usar, lo cual puede estimular la calidad de las discusiones. Sin embargo, debe tenerse en cuenta que se está pasando de un entorno de conversaciones informales cara a cara y de envío de correos electrónicos a un entorno basado en la base de conocimiento común creada a partir de las discusiones en las wikis. Este cambio requiere de cierto tiempo por parte de los usuarios para que se familiaricen con esta nueva mentalidad, a pesar de que su uso no sea complejo. Algunos participantes que inicialmente rechazaron el uso de discusiones, con el tiempo fueron aumentando su participando, lo cual hizo que su percepción acerca de la calidad de las discusiones también aumentara.
- Las discusiones son consideradas de mayor utilidad cuando se centran en resolver asuntos relacionados con un proceso, práctica o técnica porque ayudan a su comprensión y aprendizaje, lo cual provoca como consecuencia que la calidad de las discusiones se perciba como mayor.
- Las discusiones que por el contrario se centran más en aspectos específicos de un proyecto concreto hacen que se reduzca la calidad general percibida acerca de las discusiones puesto que no todos los usuarios las consideran útiles para otros proyectos. En ocasiones también ocurre que al crear una nueva discusión, el usuario que la crea la asocia a un elemento de la wiki que no es el más apropiado, confundiendo así a otros usuarios y empeorando la calidad percibida. Por lo tanto, se deberían dar algunas guías o indicaciones para ayudar a los usuarios a seleccionar el elemento correcto al cual se debería asociar una nueva discusión.

En el caso de las lecciones aprendidas (Tabla 5.8), se encontró una regresión múltiple ( $R^2 = 0.9547$ ,  $p$ -valor = 0.0003588) en la que las variables *número total de discusiones de la etapa de formación que se convirtieron en lecciones aprendidas en la etapa de proyecto* (N), *su utilidad para el aprendizaje* (UA), *su utilidad para el proyecto* (UP) y *el número de nuevas aportaciones a las discusiones en la etapa de proyecto* (ND) actúan como predictores de la *calidad de las lecciones aprendidas* (Qla).

**Tabla 5.8. Factores relacionados con la calidad de las lecciones aprendidas en la etapa de proyecto en la fase 2.**

Participante	Qla	N	UA	UP	ND
P1	4	0	3.2	4.00	0
P2	3	2	2.4	2.80	0
P3	3	0	1.6	2.65	0
P4	3	1	3.2	2.10	0
P5	3	0	2.4	1.90	0
P6	2	5	3.2	2.40	8
P7	3	4	2.4	2.15	1
P8	3	0	1.6	3.05	0
P9	n/a	0	n/a	n/a	1
P10	3	0	2.4	1.90	0
P11	4	2	2.4	n/a	10
P12	1	2	0.8	1.70	10
P13	3	0	1.6	1.65	0

$$Qla = 1.92123 + 0.21960 \cdot UA + 0.28930 \cdot UP - 0.02057 \cdot N - 0.15265 \cdot ND \quad (E5)$$

El análisis de la ecuación E5 muestra que:

- Tanto la utilidad para el aprendizaje como la utilidad para el proyecto son potenciadores de la calidad percibida respecto de las lecciones aprendidas. Las lecciones aprendidas ayudan a consolidar el nuevo conocimiento que proviene de las discusiones, resultando este proceso de consolidación clave para aumentar la calidad de las lecciones aprendidas.
- El número de discusiones en la etapa de formación que han pasado a ser lecciones aprendidas en la etapa de proyecto resulta ser un inhibidor de la calidad de las lecciones aprendidas. Esto sugiere que el rol de la gestión de cambios supervisada es muy importante puesto que no todas las discusiones deberían ser aceptadas en el proceso de realimentación. Cuando la supervisión es rigurosa, solamente las mejores discusiones pasan a formar parte del repositorio de conocimiento organizacional, de manera que la calidad de las consiguientes lecciones aprendidas aumenta. Además, sería conveniente publicar los criterios por los cuales se aceptan o rechazan las aportaciones en esta supervisión, lo cual aumentaría la transparencia del proceso, algo que solicitaron algunos de los participantes durante el experimento.
- El número de nuevas aportaciones a las discusiones en la etapa de proyecto también es un inhibidor de la calidad de las lecciones aprendidas. Si en la etapa de formación hubo muchas discusiones de baja calidad que pasaron a ser lecciones aprendidas, la creación de muchas discusiones en la etapa de proyecto hace pensar a los participantes que la calidad de las lecciones aprendidas será menor. Al igual que en el punto anterior, la gestión de cambios supervisada juega un papel importante para filtrar las nuevas aportaciones y decidir si deberían o no convertirse en lecciones aprendidas.

### 5.5.5. PI5: Efectividad de las distintas alternativas (supervisada y no-supervisada)

Para comparar los diferentes mecanismos (supervisado y no-supervisado) para añadir nuevo conocimiento al repositorio de conocimiento organizacional se utilizaron los datos obtenidos en la segunda encuesta de la fase 2 relacionados con los niveles de calidad percibidos por los participantes de las wikis supervisada y no-supervisada. Con estos datos se realizó una prueba t de Student con un intervalo de confianza del 95%. Esta prueba puede determinar si existe alguna diferencia en los niveles de calidad percibidos por los participantes que utilizaron la wiki supervisada respecto de los niveles de calidad percibidos por los participantes que utilizaron la wiki no-supervisada. Las hipótesis definidas para esta primera prueba fueron:

*H<sub>0</sub>: niveles percibidos de calidad similares usando distintos mecanismos*

*H<sub>1</sub>: niveles percibidos de calidad diferentes usando distintos mecanismos*

**Tabla 5.9. Niveles de calidad percibidos utilizando distintos mecanismos de realimentación (supervisado y no-supervisado).**

Contribuciones	Supervisado		No-supervisado		Primera prueba		Segunda prueba	
	Media	DE	Media	DE	t	p-valor	t	p-valor
Ejemplos	4.38	0.52	3.25	1.5	1.4573	0.2316	1.4573	0.1158
Discusiones	3.38	1.30	2.50	1.00	1.2872	0.2347	1.2872	0.1174
Lecciones Aprendidas	3.00	0.53	2.75	1.26	0.3806	0.7252	0.3806	0.3626

Como se puede ver en la columna correspondiente a la primera prueba de hipótesis de la Tabla 5.9, ninguno de los resultados presenta un p-valor por debajo del nivel de significación de 0.05, por lo que la hipótesis nula no se puede rechazar para ningún tipo de contribución (ejemplos, discusiones y lecciones aprendidas) y no se puede asegurar en un principio que los niveles de calidad percibidos sean diferentes dependiendo del mecanismo de realimentación que utilice la wiki a la cual se tiene acceso. Sin embargo, puesto que los p-valores obtenidos son bajos, se decidió analizar más en profundidad los datos mediante la realización de una segunda prueba de hipótesis cambiando la hipótesis alternativa  $H_1$ :

$H_0$ : niveles percibidos de calidad similares usando distintos mecanismos

$H_1$ : media de calidad con mecanismo supervisado es mayor que con no-supervisado

Los resultados que se muestran en la columna correspondiente a la segunda prueba de hipótesis de la Tabla 5.9 siguen sin ser concluyentes, pero revelan que usando el mecanismo supervisado es más verosímil que aumente el nivel de calidad percibido por los participantes respecto a los activos de conocimiento elicitados. Durante la realización del experimento, la supervisión se ha mostrado como un mecanismo útil para filtrar las contribuciones de los usuarios en función de la calidad de las mismas. Existen algunos factores que pueden reducir la calidad de estas contribuciones: ejemplos repetidos, discusiones asociadas con elementos inapropiados en la wiki, discusiones de poca calidad que se convierten en lecciones aprendidas, etc. Estas situaciones pueden evitarse si un experto supervisa las nuevas contribuciones antes de que pasen a formar parte del repositorio organizacional. El experto podría editar (si fuera necesario) una contribución para adaptarla al nivel requerido de calidad o incluso descartarla si su adaptación no es posible.

### 5.5.6. PI6: Contribución de ITAKA a la mejora de la facilidad del aprendizaje de procesos

La Tabla 5.10 compara los niveles de calidad de los productos obtenidos durante la etapa de proyecto en ambas fases del experimento expresados en una escala de 0 a 1. Como ya se ha mencionado anteriormente, la calidad de los productos se determinó aplicando un proceso de evaluación multi-fuente por el cual dos expertos evaluaron de manera individual los mismos productos utilizando unos criterios comunes pre-establecidos (ver Anexo A: Criterios para la Evaluación de Productos). Los productos de aquellos participantes que no completaron ambas etapas (formación y proyecto) no han sido incluidos en este análisis.



Tabla 5.10. Niveles de calidad de los productos en la etapa de proyecto de las fases 1 y 2.

Productos de trabajo	Fase 2		Fase 1		Primera prueba		Segunda prueba	
	Media	DE	Media	DE	t	p-valor	t	p-valor
Historias de usuario	0.92	0.16	0.99	0.03	-0.8501	0.4555	-0.8501	0.7722
Pruebas de aceptación	0.85	0.10	0.87	0.16	-0.2000	0.8465	-0.2000	0.5768
Planificación adaptativa	0.56	0.32	0.58	0.25	-0.0913	0.9306	-0.0913	0.5347
Estándares de codificación	1.00	0.00	0.50	0.55	2.2361	0.0756	2.2361	0.0378
Diseño simple	0.60	0.08	0.75	0.17	-1.8443	0.1043	-1.8443	0.9479
Reorganización de código	0.78	0.26	0.72	0.25	0.3805	0.7158	0.3805	0.3579
Pruebas unitarias	0.75	0.23	0.38	0.15	2.8556	0.0361	2.8556	0.0181
Propiedad colectiva de código	0.85	0.19	0.89	0.17	-0.3380	0.7469	-0.3380	0.6266
Integración continua	0.97	0.06	0.83	0.15	1.9766	0.0886	1.9766	0.0443
Seguimiento del proyecto	0.75	0.29	0.71	0.25	0.2370	0.8208	0.2370	0.4104

Para comparar los niveles de calidad en las dos fases del experimento se utilizó una prueba t de Student con un intervalo de confianza del 95%. Esta prueba puede determinar si existe alguna diferencia entre los niveles de calidad de los productos obtenidos en las dos fases del experimento. Las hipótesis definidas para esta primera prueba fueron:

$H_0$ : niveles de calidad similares en los productos de ambas fases

$H_1$ : niveles de calidad diferentes en los productos de ambas fases

Como se puede ver en la columna correspondiente a la primera prueba de hipótesis de la Tabla 5.10, tan solo en el caso de las pruebas unitarias el p-valor obtenido está por debajo del nivel de significación de 0.05, por lo que la hipótesis nula se puede rechazar para el caso de las pruebas unitarias pero no en general. Además, la tabla muestra otros resultados relevantes para la primera prueba de hipótesis que también requieren un análisis más en detalle, como es el caso de estándares de codificación, diseño simple e integración continua.

En cuanto al resto de partes mostradas en la Tabla 5.10, no se observan diferencias significativas en la calidad de los productos obtenidos. En general, sus niveles de calidad ya eran altos en la fase 1 (más de 0.75 en una escala de 0 a 1), por lo que es difícil mejorarlos. Sin embargo, hay una excepción que es la calidad de los productos obtenidos durante la planificación adaptativa de proyectos, que tiene la media más baja. Esta parte requiere cierta experiencia por parte de los participantes para planificar de manera adecuada un proyecto. Sin embargo, es en este curso de formación en el que se realizó el experimento cuando los participantes planifican un proyecto por primera vez, por lo que resulta difícil para algunos de ellos obtener una planificación buena (con alto nivel de calidad).

La primera prueba de hipótesis mostró que algunos niveles de calidad eran diferentes usando ITAKA que usando PAL-Wiki, pero todavía no se puede afirmar si esas diferencias fueron positivas o negativas. Por ello, con el objetivo de analizar más en profundidad los cambios en la calidad de los productos de trabajo, se realizó una prueba t de Student adicional manteniendo la misma hipótesis nula  $H_0$  pero cambiando la hipótesis alternativa  $H_1$ :

$H_0$ : niveles de calidad similares en los productos de ambas fases

$H_1$ : media de calidad en la fase 2 es mayor que en la fase 1

Como se puede ver en la columna correspondiente a la segunda prueba de hipótesis de la Tabla 5.10, los nuevos resultados indican que la calidad de los productos mejoró en pruebas unitarias, estándares de codificación e integración continua. A pesar de estas mejoras, también parece haber un descenso en la calidad de los productos relacionados con diseño simple.

Las pruebas unitarias y la integración continua son partes complejas dentro de un proyecto ágil de desarrollo de software y los participantes encuentran difícil probar e integrar software de manera correcta. Estas partes requieren de una gran cantidad de conocimiento tácito, por lo que los ejemplos, las discusiones y las lecciones aprendidas ayudaron a los participantes a compartir sus experiencias y a clarificar sus dudas a lo largo de la fase 2 del experimento. Es más, las nuevas aportaciones en forma de ejemplos fueron también útiles para definir mejor los estándares de codificación para la etapa de proyecto de la fase 2. Puesto que los estándares de codificación obtenidos en la etapa de formación estuvieron disponibles para todos como ejemplos al comienzo de la etapa de proyecto, los participantes pudieron escoger las mejores reglas de cada estándar de codificación obteniendo así un producto final mejor en la etapa de proyecto.

La mayor dificultad está en que obtener estas mejoras requiere de cierto tiempo, de tal manera que los beneficios se muestran más claramente en las últimas partes que las primeras, tal y como se puede observar en la Figura 5.6. Una vez que el sistema está completamente desplegado, los participantes tienen que habituarse a su uso y necesitan algo de práctica para entender cómo utilizarlo eficientemente. Es más, los instructores deben animar a los participantes con el objetivo de cambiar su mentalidad hacia un uso activo y colaborativo de ITAKA.

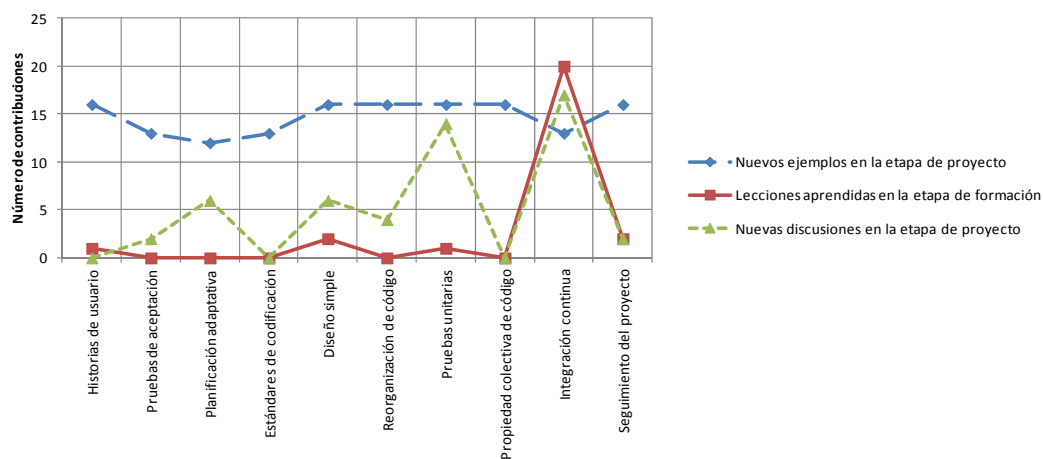


Figura 5.6. Nuevo conocimiento en la etapa de proyecto de la fase 2.

Por último, en relación con el descenso de la calidad de los productos relacionados con diseño simple, esta es una parte en la que los participantes habitualmente tienen problemas. En ocasiones realizan modificaciones en el código fuente que desarrollan pero olvidan actualizar el diseño de acuerdo con dichas modificaciones del código, de forma que la implementación del software no se corresponde con los diagramas diseñados. En esta ocasión, los participantes cometieron más errores documentando sus proyectos en la fase 2 que en la fase 1, pero sus diseños eran correctos.

En resumen, los resultados presentados en esta sección muestran que los participantes fueron capaces de adoptar las nuevas prácticas de desarrollo de software y de ser productivos utilizando un sistema de gestión del conocimiento que era nuevo para ellos. En la etapa de proyecto se apreciaron mejoras en la calidad de determinados productos obtenidos en la fase 2 con respecto a los obtenidos en la fase 1, concretamente en los productos correspondientes a estándares de codificación, pruebas unitarias e integración continua. La mayor dificultad reside en que la obtención de estas mejoras requiere de un cierto tiempo de adaptación, de manera que los beneficios se muestran más claramente a medida que se utiliza ITAKA.

Los resultados y conclusiones obtenidos en relación con esta pregunta de investigación han sido también incluidos en un artículo [Heredia, 2012] que ha sido aceptado para su publicación en la revista *Journal of Information Science and Engineering*. En dicho artículo se presenta la parte de la solución correspondiente a distribución del conocimiento, preservación del conocimiento y gestión de cambios, utilizando los datos mostrados en este apartado para validar si esta aproximación contribuye a mejorar la facilidad del aprendizaje y adopción de nuevos procesos y prácticas para el desarrollo de software.

### 5.5.7. PI7: Evaluación subjetiva de la utilidad de los mecanismos de gestión de la configuración propuestos

En general, los participantes consideraron que los ejemplos resultaron útiles para resolver dudas, optimizar el tiempo y enriquecer su conocimiento tácito. Por ello, la utilidad de los ejemplos fue evaluada de manera muy positiva en las encuestas, especialmente en la wiki supervisada y en la wiki de solo-lectura con valores superiores a 4.3 en una escala de 1 a 5 (Tabla 5.11).

Tabla 5.11. Utilidad del nuevo conocimiento percibida por los participantes.

			Ejemplos	Discusiones	Lecciones Aprendidas	Promedio
Fase 1	Etapa de formación	PAL-Wiki	4.47	2.86	n/a	3.66
	Etapa de proyecto	PAL-Wiki	–	–	n/a	–
Fase 2	Etapa de formación	Wiki supervisada	4.38	3.00	n/a	3.69
		Wiki no-supervisada	3.00	1.75	n/a	2.38
		Wiki de solo-lectura	4.67	n/a	n/a	4.67
		Todas las wikis	4.00	2.58	n/a	3.29
	Etapa de proyecto	Wiki supervisada	4.50	3.25	3.00	3.58
		Wiki no-supervisada	3.25	2.25	1.75	2.42
		Wiki de solo-lectura	4.33	n/a	3.33	3.83
		Todas las wikis	4.13	2.92	2.73	3.26

Sin embargo, los participantes no estuvieron de acuerdo acerca de la utilidad de las discusiones y de las lecciones aprendidas. Por un lado, los usuarios de la wiki supervisada consideraron que las discusiones y las lecciones aprendidas eran útiles, pero no tanto como los ejemplos. Por otro lado, los usuarios de la wiki no-supervisada consideraron que las discusiones no eran útiles. En este caso, la falta de participación en la wiki no-supervisada afectó a la percepción de la utilidad de este tipo de contribuciones, y esto a su vez afectó a la calidad percibida.

Para comparar de nuevo los diferentes mecanismos (supervisado y no-supervisado) para añadir nuevo conocimiento al repositorio de conocimiento organizacional, se realizó un análisis de los datos obtenidos de la segunda encuesta de la fase 2 en relación con los niveles de utilidad percibidos por los usuarios tanto de la wiki supervisada como de la no-supervisada. Para ello se realizó una prueba t de Student con un intervalo de confianza del 95%. Esta prueba puede determinar si existe alguna diferencia en los niveles de utilidad percibidos por los participantes que utilizaron la wiki supervisada respecto de los niveles de utilidad percibidos por los participantes que utilizaron la wiki no-supervisada. Las hipótesis definidas para esta primera prueba fueron:

$H_0$ : niveles percibidos de utilidad similares usando distintos mecanismos

$H_1$ : niveles percibidos de utilidad diferentes usando distintos mecanismos

**Tabla 5.12. Niveles de utilidad percibidos utilizando distintos mecanismos de realimentación (supervisado y no-supervisado).**

Contribuciones	Supervisado		No-supervisado		Primera prueba		Segunda prueba	
	Media	DE	Media	DE	t	p-valor	t	p-valor
Ejemplos	4.50	3.25	3.25	1.71	1.4293	0.2404	1.4293	0.1202
Discusiones	3.25	0.71	2.25	0.96	1.8516	0.1269	1.8516	0.0634

Como se puede ver en la columna correspondiente a la primera prueba de hipótesis de la Tabla 5.12, ninguno de los resultados presenta un p-valor por debajo del nivel de significación de 0.05, por lo que la hipótesis nula no se puede rechazar para ningún tipo de contribución (ejemplos y discusiones) y no se puede asegurar en un principio que los niveles de utilidad percibidos sean diferentes dependiendo del mecanismo de realimentación que utilice la wiki a la cual se tiene acceso. Sin embargo, puesto que los p-valores obtenidos son bajos, se decidió analizar más en profundidad los datos mediante la realización de una segunda prueba de hipótesis cambiando la hipótesis alternativa  $H_1$ :

$H_0$ : niveles percibidos de utilidad similares usando distintos mecanismos

$H_1$ : media de utilidad con mecanismo supervisado es mayor que con no-supervisado

Los resultados que se muestran en la columna correspondiente a la segunda prueba de hipótesis de la Tabla 5.12 siguen sin ser del todo concluyentes, pero revelan que usando el mecanismo supervisado es más verosímil que aumente el nivel de utilidad percibido por los participantes.

Puesto que el análisis cuantitativo no fue concluyente, también se decidió realizar un análisis cualitativo de la utilidad de los nuevos activos de conocimiento, a partir del cual se obtuvieron las siguientes apreciaciones:

- La aplicación de una estrategia de gestión de cambios supervisada facilita la creación de un conjunto bien formado de activos de conocimiento que cubren los casos generales para la aplicación efectiva de un proceso o técnica ágil en las situaciones más habituales. Además, este conjunto también incluye otros activos de conocimiento relacionados con casos especiales que hayan tenido que ser tenidos en cuenta en proyectos concretos. Por el contrario, la estrategia de gestión de cambios no-

supervisada puede implicar la realimentación de conocimiento tácito relacionado con temas repetidos, reduciendo así el valor añadido de todo el conjunto de activos de conocimiento.

- La aplicación de una estrategia supervisada también permite revisar la redacción de las discusiones a la hora de transformarlas en lecciones aprendidas. De esta manera, las lecciones aprendidas obtenidas mediante realimentación supervisada están mejor organizadas y redactadas que aquellas obtenidas mediante realimentación no-supervisada. La aplicación de unas normas de estilo acerca de cómo participar adecuadamente en las discusiones ayuda a los ingenieros de software a participar de manera más efectiva en las discusiones y facilita la obtención de lecciones aprendidas por medio de la estrategia no-supervisada.

## 5.6. Conclusiones de la validación

Los resultados del experimento controlado muestran que ITAKA es efectivo en la captura, formalización y distribución del conocimiento tácito que surge de las diferentes interacciones entre los miembros de una organización de ingeniería del software. Estos nuevos activos provenientes del conocimiento tácito son almacenados junto con el conocimiento explícito existente dentro del mismo repositorio de conocimiento organizacional, quedando así disponibles para ser reutilizados por cualquier miembro de la organización.

Durante el proceso de validación se descubrió que los participantes tienden más a añadir nuevos activos de conocimiento (ejemplos y discusiones) que a modificar los activos ya existentes proporcionados por los instructores del curso de formación. El análisis cualitativo de los datos recogidos sugiere que este hecho fue debido a que los participantes no se consideraban lo suficientemente experimentados como para realizar cambios en el conocimiento pre-existente, de ahí que prefirieran complementarlo mediante la aportación de nuevos activos de conocimiento. Esto contradice en cierta medida el uso esperado de cualquier wiki, en la cual la realización de cambios en el conocimiento existente es algo que cabe esperar. Todo esto sugiere que en un entorno de aprendizaje de procesos de software, la modificación de los activos de conocimiento relacionados con esos procesos es una excepción más que una regla.

El análisis de los datos tomados durante el experimento permitió encontrar algunos factores que contribuyen al uso de ITAKA como una fuente de conocimiento relevante. Por un lado, durante la etapa de formación parecen tener mayor relevancia el conocimiento pre-existente y los ejemplos ya que facilitan el aprendizaje de los nuevos procesos y prácticas de desarrollo de software. Por otro lado, durante la etapa de proyecto parecen tener mayor relevancia los ejemplos y las lecciones aprendidas puesto que ayudan a adaptar los procesos de desarrollo de software a las restricciones y al contexto específico del proyecto.

En relación con el enriquecimiento del conocimiento pre-existente mediante discusiones, se puede afirmar que los factores que más estimulan el uso de discusiones son el número de visitas a las wikis que permiten visualizar el conocimiento y el grado en que los usuarios están centrados en la búsqueda de información. Así, cuando las visitas no se enfocan a buscar información concreta aumenta el tiempo de las mismas y el número de páginas vistas durante

cada visita, actuando ambos factores como inhibidores del número de discusiones. El uso de los mecanismos de transformación de discusiones en lecciones aprendidas ayuda a adquirir este tipo de conocimiento tácito de una forma homogénea.

Por otro lado, los mecanismos de realimentación permiten aportar ejemplos para enriquecer el conocimiento de la organización. Estos ejemplos se construyen a partir de los artefactos obtenidos durante la ejecución de los procesos en los proyectos y tienen varias ventajas sobre los artefactos de los cuales provienen: en primer lugar, están disponibles para todos los miembros de la organización, no solo para los miembros del equipo de trabajo, permitiendo que sean utilizados como guía para la ejecución de ese u otros proyectos; en segundo lugar, proporcionan información del contexto en el cual fueron creados; finalmente, también proporcionan información acerca del problema que resuelven y de cómo lo resuelven, e incluso pueden contener otros artefactos relacionados con el problema o con la solución.

En cuanto a los factores que influyen en la calidad del conocimiento, los resultados indican que cuanto mayor es el número de contribuciones de los usuarios al sistema, mayor es la calidad percibida por los usuarios, siendo los ejemplos los más influyentes por encima de discusiones y lecciones aprendidas. Esto sugiere que el uso de ejemplos en ITAKA es más aceptado que el uso de discusiones o lecciones aprendidas.

Aparte de la cantidad, otro factor que muestra influencia en la calidad es la dispersión del conocimiento. La calidad del conocimiento percibida por los usuarios es mayor cuando el nuevo conocimiento está relacionado con temas variados y, además, no existe conocimiento duplicado.

La usabilidad de las herramientas y de los procesos dentro de ITAKA también tiene gran influencia en la calidad percibida. La sencillez de los mecanismos para participar en las discusiones fue un factor positivo en relación con la calidad de este tipo de activos. Sin embargo, los participantes encontraron dificultades en los mecanismos para aportar ejemplos, lo cual afectó negativamente a la calidad percibida respecto de los ejemplos.

Un último factor que puede afectar a la percepción de la calidad es el nivel de conocimiento adquirido por los participantes. Aquellos usuarios que contribuyeron más activamente a las discusiones en las etapas tempranas del experimento evaluaron de manera menos positiva las discusiones aportadas por otros participantes. Esto pudo ser debido a que, cuanto más participan, más conocimiento tácito esperan adquirir del resto de participantes en compensación, de forma que si las aportaciones de otros participantes son pobres, la calidad general percibida es menor. El caso contrario ocurre con los que menos participan, los cuales valoran muy positivamente la calidad del conocimiento puesto que adquieren conocimiento tácito de otros participantes sin tener ellos que aportar mucho al resto.

En cuanto a la comparación entre los dos mecanismos de realimentación propuestos (supervisado y no-supervisado), esta no fue concluyente pero reveló que utilizar el mecanismo supervisado hacía más probable un aumento de la calidad percibida por los participantes. La revisión de las contribuciones ha probado ser un mecanismo útil para filtrar este nuevo conocimiento permitiendo, por ejemplo, reducir el número de ejemplos duplicados, evitar que las contribuciones estén asociadas con activos a los que no corresponden, impedir que

discusiones de baja calidad pasen a formar parte de las lecciones aprendidas, etc. Estas situaciones son evitables si un experto o un jefe de equipo revisan las contribuciones antes de que pasen a formar parte del conocimiento organizacional. Este experto puede editar (si es necesario) los contenidos de la contribución para adaptarla al nivel requerido de calidad o incluso descartarla si su adaptación no es viable.

Por otro lado, los datos también indican que los participantes necesitaron de un esfuerzo menor para utilizar las características de ITAKA en relación con las características de otras PAL. Sus visitas a las wikis fueron más breves y necesitaron acceder a menos páginas en cada visita. Sin embargo, se produjo un aumento en el número de visitas por participante, lo cual revela un uso más activo de las wikis. Sin embargo, a pesar del poco esfuerzo requerido para utilizar las funcionalidades del sistema, se necesita de cierto tiempo para acostumbrarse a su uso y para lograr que los usuarios se involucren. Así, la calidad de los productos desarrollados mejora con el paso del tiempo.

Para facilitar la implementación de ITAKA ha sido fundamental la publicación de los objetivos, estrategias, roles, responsabilidades y beneficios entre todos los implicados en su uso. Para transmitir de forma clara esta información es recomendable utilizar distintos canales, desde la publicación en web hasta las conversaciones cara a cara. Es muy importante maximizar la visibilidad del proceso de compartición de conocimiento ya que este tipo de aproximaciones, aunque no son complejas, suelen requerir un cambio en la mentalidad de las personas implicadas hacia una cultura de aprendizaje colaborativo.

También se puede concluir que ITAKA ayuda a los ingenieros de software a mejorar su aprendizaje en relación con los procesos de software y a la correcta adopción de dichos procesos para desarrollar mejores productos. Los ejemplos proporcionan una base para producir artefactos. Las discusiones y las lecciones aprendidas ayudan en la ejecución de tareas complejas que requieren gran cantidad de conocimiento tácito ya que permiten a los ingenieros de software compartir experiencias para resolver sus dudas.

Por último, ITAKA ha demostrado ser útil en la adquisición de nuevo conocimiento proveniente de distintas fuentes y en la diseminación a través de distintas plataformas de este conocimiento elicitado. Así, el mismo conocimiento organizacional puede ser distribuido en diferentes formatos dependiendo de las necesidades de sus usuarios.





# Capítulo 6: Conclusiones y Futuras Líneas de Investigación

6.1. Conclusiones .....	169
6.2. Líneas futuras.....	173



En este capítulo se describen las conclusiones obtenidas a partir de la realización de la presente tesis doctoral, así como las posibles ideas que pueden abrir nuevas líneas de investigación a partir de la realización de esta tesis doctoral.

## 6.1. Conclusiones

Esta tesis doctoral propone una solución teórico-práctica para el enriquecimiento del conocimiento organizacional, llamada ITAKA (Interactive TAcit Knowledge Administration), la cual proporciona un marco de trabajo compuesto por un modelo de conocimiento, los procesos para su gestión y una plataforma tecnológica para la administración interactiva y colaborativa del conocimiento tácito necesario para facilitar el aprendizaje y la adopción de nuevos procesos de desarrollo de software.

La creación de ITAKA estuvo motivada principalmente por la necesidad de proporcionar mecanismos para la elicitación de activos de conocimiento tácito **de alta calidad** en forma de ejemplos, discusiones, lecciones aprendidas o cambios en el conocimiento pre-existente, siendo estos activos generados durante el aprendizaje y la adopción de nuevas prácticas de ingeniería del software. **ITAKA administra el conocimiento tácito** generado en los proyectos facilitando la interacción entre los miembros de los equipos de trabajo de una organización. Además, **ITAKA permite el intercambio de experiencias** entre ingenieros de software que trabajen en **distintos equipos de proyecto y/o distintas localizaciones** y que utilicen **distintos medios para acceder a los mismos activos de conocimiento de la organización**.

ITAKA aplica los principios de KM en el ámbito de la Ingeniería del Software, incorporando además mecanismos de gestión del cambio y de gestión de la configuración, con el objetivo de que el conocimiento tácito que se genera durante la realización de los proyectos se fusione con el resto del conocimiento ya existente en la organización, **quedando registrado todo el conocimiento en un único repositorio**. Además, la solución hace uso de las tecnologías de la información para dar soporte al despliegue práctico del marco de trabajo que se propone en esta tesis doctoral. De esta manera, ITAKA se compone de dos elementos principales: un proceso para ayudar a las organizaciones a llevar a cabo una buena gestión de su conocimiento organizativo, y una plataforma para dar soporte a dicho proceso.

Respecto a la aportación de esta tesis doctoral en relación con los **mecanismos de gestión de cambios**, la solución propone dos mecanismos de alternativos para la realimentación de conocimiento tácito de los proyectos: uno supervisado, en el cual un experto evalúa la calidad del nuevo conocimiento y decide si los cambios deben trasladarse al repositorio de conocimiento organizacional, y otro no-supervisado, en el cual el nuevo conocimiento pasa directamente a formar parte del repositorio de conocimiento organizacional.

Por otro lado, respecto a la aportación de esta tesis doctoral en relación con los **mecanismos de gestión de la configuración**, se puede afirmar que la solución propuesta contempla varios aspectos de las distintas actividades de la gestión de la configuración:

- Dentro de la actividad de identificación de la configuración, los elementos de configuración seleccionados han sido los activos de conocimiento, definiéndose también relaciones de dependencia, derivación y composición entre ellos. Además, se

establece una línea base cada vez que se distribuye una nueva versión del conocimiento organizacional a un equipo de trabajo, estableciéndose además como biblioteca la vista correspondiente donde queda almacenada dicha versión del conocimiento.

- En cuanto a la actividad de control de cambios de la configuración, se debe tener en cuenta que se producen cambios en los activos de conocimiento almacenados en las diferentes vistas con un nivel de control de cambios informal, y también se producen cambios en los activos de conocimiento almacenados en el repositorio de conocimiento organizacional con un nivel de control de cambios informal cuando se utiliza el mecanismo no-supervisado y semi-formal cuando se utiliza el mecanismo supervisado. Solo en estos dos últimos casos hay definido un comité de control de cambios, estando formado únicamente por el bibliotecario para el mecanismo no-supervisado, al que se suman los expertos para el mecanismo supervisado. En cuanto a la solicitud de cambios, esta no se realiza de manera expresa sino que está implícita en los mecanismos de realimentación cuando el bibliotecario recopila todos los cambios realizados en las distintas vistas.
- Respecto a la generación de informes de estado, siguiendo los propios principios de las técnicas ágiles de desarrollo de software del curso de formación en el que se realizó la validación de la solución propuesta, estos no se generan al no ser considerados como imprescindibles, aunque podrían resultar de utilidad. Cabe destacar que sí se lleva un registro sobre los cambios realizados tanto en las vistas como en el repositorio de conocimiento organizacional.
- De igual manera, tampoco se realiza ninguna auditoría de la configuración, quedando a criterio del bibliotecario y de los expertos la revisión de los cambios que se produzcan en el conocimiento organizacional.
- Finalmente, mencionar que la elaboración de un plan que detalle las actividades de gestión de la configuración a realizar durante los proyectos no se abordado al considerarse parte del día a día de los propios proyectos, por lo que se ha optado por incluirlo en la definición de los distintos procesos de la solución.

Aparte de las aportaciones de este trabajo de investigación mencionadas antes, hay que destacar que la solución propuesta ofrece **mecanismos sencillos para registrar conocimiento**, tanto en el caso del conocimiento pre-existente (conocimiento inicial) como en el caso del nuevo conocimiento que se quiera preservar en los proyectos. La posibilidad de definir los distintos activos de conocimiento utilizando texto y elementos multimedia hace que no sea necesaria (aunque sí recomendable) la presencia de un experto en procesos para la formalización de los procesos de la organización.

Para la validación de esta solución se realizó un experimento controlado que involucró a 46 ingenieros de software junior. En la primera fase del experimento, los usuarios utilizaron una PAL sin mecanismos de gestión de la configuración ni de gestión del cambio, mientras que en la segunda fase se utilizó ITAKA. Los resultados obtenidos sugieren que ITAKA es efectivo en la captura, formalización y distribución del conocimiento tácito que surge de las diferentes interacciones entre los miembros de una organización de ingeniería del software. ITAKA

también proporciona un entorno efectivo para el aprendizaje y la adopción de nuevos procesos y prácticas para el desarrollo de productos software de calidad. A pesar del poco esfuerzo requerido para utilizar las funcionalidades del sistema, se necesita de cierto tiempo para acostumbrarse a su uso y para lograr que los usuarios se involucren. Así, la calidad de los productos desarrollados mejora con el paso del tiempo.

Un análisis más exhaustivo de estos resultados indica que el éxito en la utilización de ITAKA depende de los siguientes factores:

**1. Utilización centrada principalmente en la formación y la resolución de problemas.**

Cada una de las dos fases del experimento estuvo dividida en dos etapas: en la primera se utilizaba una solución para el aprendizaje, y en la segunda se utilizaba la misma solución para la ejecución de un proyecto concreto. Al finalizar cada etapa se realizó una encuesta para explorar la utilidad percibida en relación con el conocimiento tácito en forma de ejemplos, discusiones y lecciones aprendidas. El análisis de los resultados obtenidos muestra que los usuarios consideran ITAKA como útil para el aprendizaje (etapa de formación) y para la resolución de problemas (etapa de proyecto) para los tres tipos de activos de conocimiento mencionados. Otros aspectos acerca de la utilidad, como por ejemplo que ITAKA ayuda a aprender más rápido o a desarrollar productos en menor tiempo recibieron peores puntuaciones, al igual que otras cuestiones como por ejemplo si ITAKA ayudaba a entender mejor el contexto de negocio de los proyectos. Por este motivo, el uso de ITAKA para actividades distintas de la formación y de la resolución de problemas debería ser analizado con mayor profundidad.

**2. Participación de la comunidad de usuarios.** Se ha descubierto que la participación de los miembros de la comunidad puede ser positiva o negativa dependiendo de varios factores. Estos son:

- a. Número de contribuciones (ejemplos, discusiones, lecciones aprendidas) publicadas en ITAKA. Los resultados indican que cuanto mayor es el número de contribuciones de los usuarios al sistema, mayor es la calidad del sistema percibida por los miembros de la comunidad. El número de ejemplos publicados es más influyente en la calidad que el número de discusiones o de lecciones aprendidas. Esto sugiere que el uso de ejemplos en ITAKA es más aceptado que el uso de discusiones o lecciones aprendidas. De manera similar, algunos participantes mencionaron que la falta de participación por parte de otras personas afectó negativamente a la calidad del conocimiento gestionado por ITAKA. Los participantes que contribuyeron de forma más activa en la preservación de conocimiento tácito fueron los que desarrollaron productos de mayor calidad en la etapa de proyecto, mientras que aquellos que no contribuyeron tan activamente fueron los que desarrollaron productos de menor calidad que los anteriores.
- b. Conocimiento duplicado. El número de ejemplos aceptados en el proceso de gestión de cambios tuvo una influencia negativa en la percepción de la calidad. El análisis incluye tanto aquellas contribuciones que fueron supervisadas por

los expertos como aquellas que no fueron supervisadas. El análisis cualitativo de los ejemplos aportados mostró que algunos aportaban conocimiento duplicado, mayormente en el caso de los ejemplos que no fueron supervisados. Esto sugiere que, aunque la participación puede ayudar a aumentar la percepción de utilidad de ITAKA, es altamente recomendable la utilización de algún mecanismo para filtrar el conocimiento. De lo contrario, la participación puede provocar un efecto contrario en la utilidad del sistema percibida por los miembros de la comunidad.

- c. **Dispersión del nuevo conocimiento adquirido.** El análisis cualitativo de las aportaciones realizadas por los participantes muestra un gran número de ejemplos centrados en temas específicos, mientras que se aportaron pocos ejemplos relacionados con el resto de temas. Es más, algunos ejemplos fueron creados y vinculados a activos que no correspondían, provocando confusión entre los participantes. Esto afectó negativamente a la calidad percibida de los ejemplos. Por lo tanto, es importante identificar si los ejemplos se están vinculando dentro del tema adecuado, así como identificar maneras de aportar ejemplos que representen un amplio rango de situaciones.
3. **Usabilidad de las herramientas/procesos.** La sencillez de los mecanismos para participar en las discusiones fue un factor positivo en relación con la calidad de este tipo de activos. Por otro lado, los participantes encontraron dificultades en los mecanismos para aportar ejemplos, lo cual afectó negativamente a la calidad percibida respecto de los ejemplos. Es necesario por tanto que los usuarios se familiaricen con las herramientas y los procesos relacionados con ITAKA para que la aportación del nuevo conocimiento se realice de una manera eficiente. Este efecto puede reducirse proporcionando unas guías que expliquen el funcionamiento de los mecanismos para aportar nuevo conocimiento en forma de ejemplos o discusiones. En este sentido, los participantes consideraron que el uso de estos mecanismos resultaba muy sencillo y rápido una vez que aprendieron a manejarlos.
4. **Adquisición de conocimiento vs. Percepción de calidad del conocimiento.** El nivel de conocimiento adquirido por los participantes puede afectar a su percepción de la calidad del conocimiento organizacional. Aquellos usuarios que contribuyeron más activamente a las discusiones en las etapas tempranas del experimento evaluaron de manera menos positiva las discusiones aportadas por otros participantes. Esto pudo ser debido a que, cuanto más participan, más conocimiento tácito esperan adquirir del resto de participantes en compensación, de forma que si las aportaciones de otros participantes son pobres, la calidad general percibida es menor. El caso contrario ocurre con los que menos participan, los cuales valoran muy positivamente la calidad del conocimiento puesto que adquieren conocimiento tácito de otros participantes sin tener ellos que aportar mucho al resto.
5. **Tendencia al crecimiento del conocimiento organizacional.** Durante el experimento, los participantes tendieron más a añadir nuevos activos de conocimiento (ejemplos y discusiones) que a modificar los activos ya existentes proporcionados por los

instructores del curso de formación. El análisis cualitativo de los datos recogidos sugiere que este hecho fue debido a que los participantes no se consideraban lo suficientemente experimentados como para realizar cambios en el conocimiento pre-existente, de ahí que prefirieran complementarlo mediante la aportación de nuevos activos de conocimiento. Esto contradice en cierta medida el uso esperado de cualquier wiki, en la cual la realización de cambios en el conocimiento existente es algo que cabe esperar. Todo esto sugiere que en las organizaciones de software que pretenden adoptar nuevos procesos, la efectividad de la wiki, mejorada mediante funcionalidades refinadas de gestión de la configuración y de gestión del cambio, depende en gran medida de la capacidad de agregar activos de conocimiento que sean relevantes para el aprendizaje de los procesos de la organización y para su adaptación a las condiciones específicas de cada proyecto.

6. **Gestión de cambios supervisada.** La validación realizada implementó dos mecanismos de realimentación alternativos para la gestión de cambios en el repositorio de conocimiento organizacional: uno en el que los cambios se incorporaban al repositorio después de ser supervisados por un experto y otro en el que los cambios se incorporaban directamente al repositorio sin ser supervisados. La comparación entre ambos mecanismos no fue concluyente, pero reveló que utilizar el mecanismo supervisado hacía más probable un aumento de la calidad percibida por los participantes. La revisión de las contribuciones ha probado ser un mecanismo útil para filtrar este nuevo conocimiento permitiendo, por ejemplo, reducir el número de ejemplos duplicados, evitar que las contribuciones estén asociadas con activos a los que no corresponden, impedir que discusiones de baja calidad pasen a formar parte de las lecciones aprendidas, etc. Estas situaciones son evitables si un experto o un jefe de equipo supervisa las contribuciones antes de que pasen a formar parte del conocimiento organizacional. Este experto puede editar (si es necesario) los contenidos de la contribución para adaptarla al nivel requerido de calidad o incluso descartarla si su adaptación no es viable.
7. **Cambio cultural.** Para facilitar la utilización eficiente de ITAKA ha sido fundamental la publicación de los objetivos, estrategias, roles, responsabilidades y beneficios entre todos los implicados en su uso. Para transmitir de forma clara esta información es recomendable utilizar distintos canales, desde la publicación en web hasta las conversaciones cara a cara. Es muy importante maximizar la visibilidad del proceso de compartición de conocimiento ya que este tipo de aproximaciones, aunque no son complejas, suelen requerir un cambio en la mentalidad de las personas implicadas hacia una cultura de aprendizaje colaborativo, lo cual no es sencillo de conseguir. De esta manera se pasa de valorar lo que una persona conoce a valorar lo que una persona comparte con su equipo de trabajo.

## 6.2. Líneas futuras

Una vez concluida esta tesis doctoral, su realización ha hecho surgir algunas ideas que pueden abrir nuevas líneas de investigación en relación con la presente tesis doctoral. Estas ideas se enumeran a continuación:

- En primer lugar, esta tesis ha establecido mecanismos para la creación de un conocimiento pre-existente de la organización y para el enriquecimiento del mismo por medio de mecanismos de realimentación a partir del conocimiento tácito generado en la ejecución de proyectos. Sin embargo, sería interesante que distintas organizaciones pudieran intercambiar distintos activos de conocimiento que pudieran resultar de utilidad. Para ello sería necesario establecer mecanismos de búsqueda y registro de activos que permitieran a una organización que trabaje utilizando ITAKA importar activos de conocimiento de fuentes externas pertenecientes a otras organizaciones con un entorno diferente de ITAKA. Sería de gran utilidad para esto la definición de alguna meta-ontología que facilitara esta compartición de activos de una manera sencilla y semi-automatizada.
- Explotación de las capacidades semánticas de la wiki. La extensión Semantic MediaWiki utilizada en esta tesis doctoral añade anotaciones semánticas a los distintos contenidos de la wiki. Sin embargo, la semántica tan solo se ha aprovechado para la creación de formularios (extensión Semantic Forms) que facilitan la creación y edición de contenidos en las wikis desplegadas. Sería conveniente explotar en mayor medida las capacidades semánticas que ofrece la wiki para ayudar en la búsqueda y compartición de conocimiento.
- También se ha mencionado que ITAKA permite desplegar wikis cuyos contenidos dependerán de los procesos requeridos en el proyecto para el cual estén destinadas, ya que en cada proyecto se ejecutan unos procesos distintos dentro del conjunto de todos los procesos de la organización. Sin embargo, sería también muy útil poder personalizar las wikis para adaptarse a roles concretos o incluso a personas concretas. De esta manera, un jefe de proyecto tendría solamente disponibles los activos de conocimiento relacionados con aquellos procesos y actividades en los que estuviera involucrado, los cuales serían diferentes de los activos disponibles para un desarrollador. Esta funcionalidad permitiría personalizar los contenidos mostrados por la wiki dependiendo por ejemplo de la capacitación del usuario, de los proyectos en los que esté involucrado, o del entorno en el que se mueva.
- Finalmente, se ha comprobado que la utilización eficiente de ITAKA requiere un periodo de adaptación para que los usuarios asimilen los nuevos procesos para la gestión del conocimiento de la organización. Estos procesos modifican la forma de trabajar de las personas, que ahora deben adoptar una cultura orientada a la colaboración para el aprendizaje y que además necesitan la adquisición de ciertas capacidades para la utilización eficiente de los nuevos procesos de gestión del conocimiento (por ejemplo, aprender cómo editar un activo de conocimiento). Esto conduce a la transformación de la empresa en una organización inteligente. Se hace necesario por tanto una investigación más en profundidad de las implicaciones que esto conlleva.



# Capítulo 7: Referencias



- [Aaen, 2001] Aaen, I.; Arent, J.; Mathiassen, L.; Ngwenyama, O.: "A Conceptual MAP of Software Process Improvement". En *Scandinavian Journal of Information Systems*, 13, pp. 123-146, 2001.
- [Aaen, 2003] Aaen, I.: "Software process improvement: Blueprints versus Recipes". En *IEEE Software*, 20 (5), pp. 86-93, 2003.
- [Aguiar, 2005] Aguiar, A.; David, G.: "WikiWiki weaving heterogeneous software artifacts". En *Proceedings of the 2005 international symposium on Wikis (WikiSym)*, pp. 67-74, San Diego, CA, USA, 16-18 Octubre 2005. ACM Press.
- [Ajmal, 2008] Ajmal, M.M.; Koskinen, K.U.: "Knowledge transfer in project-based organizations: An organizational culture perspective". En *Project Management Journal*, 39 (1), pp. 7-15, 2008.
- [Alavi, 2001] Alavi, M.; Leidner, D.: "Knowledge Management and Knowledge Management Systems: Conceptual Foundations and Research Issues". En *MIS Quarterly*, 25 (1), pp. 107-136, 2001.
- [Allison, 2007] Allison, I.; Merali, Y.: "Software Process Improvement as Emergent Change: A Structural Analysis". En *Information and Software Technology*, 49 (6), pp. 668-681, 2007.
- [Amescua, 2010] Amescua, A.; Bermón, L.; García, J.; Sanchez-Segura, M.I.: "Knowledge repository to improve agile development processes learning". En *IET Software*, 4 (6), pp. 434-444, 2010.
- [Ardimento, 2009] Ardimento, P.; Baldassarre, M.T.; Cimitile, M.; Visaggio, G.: "Empirical Validation of Knowledge Packages as Facilitators for Knowledge Transfer". En *Journal of Information & Knowledge Management*, 8 (3), pp. 229-240, 2009.
- [Arent, 2000] Arent, J.; Nørbjerg, J.: "Software Process Improvement as Organizational Knowledge Creation: A Multiple Case Analysis". En *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, pp. 40-45, Wailea, HI, USA, 4-7 Enero 2000. IEEE Computer Society Press.
- [Armitage, 1994] Armitage, J.W.; Kellner, M.I.: "A Conceptual Schema for Process Definitions and Models". En *Proceedings of the 3rd International Conference on the Software Process*, pp. 153-165, Reston, VA, USA, 10-11 Octubre 1994. IEEE Computer Society Pres.
- [Assimakopoulos, 2006] Assimakopoulos, D.; Yan, J.: "Sources of Knowledge Acquisition for Chinese Software Engineers". En *R&D Management*, 36 (1), pp. 97-106, 2006.
- [Aurum, 2008] Aurum, A.; Daneshgar, F.; Ward, J.: "Investigating Knowledge Management practices in software development organizations – An Australian experience". En *Information and Software Technology*, 50 (6), pp. 511-533, 2008.

- [Basili, 1994] Basili, V.R.; Caldiera, G.; Rombach, H.D.: "Experience Factory". J. Marciniak (Ed.), *Encyclopedia of Software Engineering*, John Wiley & Sons, 1994.
- [Basili, 2002] Basili, V.R.; Seaman, C.: "The Experience Factory Organization". En *IEEE Software*, 19 (3), pp. 30-31, 2002.
- [Baumeister, 2011] Baumeister, J.; Reutelshoefer, J.; Puppe, F.: "Engineering intelligent systems on the knowledge formalization continuum". En *International Journal of Applied Mathematics and Computer Science*, 21 (1), pp. 27-39, 2011.
- [Becker-Kornstaedt, 1999a] Becker-Kornstaedt, U. et al.: "Support for the Process Engineer: The Spearmint Approach to Software Process Definition and Process Guidance". En *Proceedings of the 11th Conference on Advanced Information Systems Engineering (CAiSE)*, pp. 119-133, Heidelberg, Germany, 14-18 Junio 1999. *Advanced Information Systems Engineering*, M. Jarke y A. Oberweis (Eds.), LNCS 1626, pp. 119-133, Springer Berlin / Heidelberg, 1999.
- [Becker-Kornstaedt, 1999b] Becker-Kornstaedt, U.; Verlage, M.: "The V-Modell Guide: Experience with a Web-Based Approach for Process Support". En *Proceedings of the Software Technology and Engineering Practice (STEP)*, pp. 161-168, Pittsburgh, PA, USA, 30 Agosto - 2 Septiembre 1999. IEEE Computer Society Press.
- [Bermón, 2010] Bermón, L.: "Librería de Activos para la Gestión del Conocimiento sobre Procesos Software: PAL-Wiki". Tesis doctoral. Universidad Carlos III de Madrid. 2010.
- [Birkinshaw, 2002] Birkinshaw, J.; Sheehan, T.: "Managing the knowledge life cycle". En *MIT Sloan Management Review*, 44 (1), pp. 75-83, 2002.
- [Bjørnson, 2008] Bjørnson, F.O.; Dingsøy, T.: "Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used". En *Information and Software Technology*, 50 (11), pp. 1055-1068, 2008.
- [Borges, 2002] Borges, L.S.M.; Falbo, R.A.: "Managing Software Process Knowledge". En *Proceedings of the International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA)*, pp. 227-232, Foz do Iguazu, Brasil, 6-8 Junio 2002.
- [Calvo-Manzano, 2008] Calvo-Manzano, J.A.; Cuevas, G.; Feliu, T.S.; Serrano, A.: "Process Asset Library to Support Software Process Improvement in Small Settings". En *Proceedings of the 15th European Systems and Software Process Improvement and Innovation (EuroSPI)*, Dublin, Irlanda, 3-5 Septiembre 2008. *Software Process Improvement*, 16, pp. 25-35, Springer Berlin / Heidelberg, 2008.
- [Capell, 2004] Capell, P.: "Benefits of Improvement Efforts". Special Report CMU/SEI-2004-SR-010. Software Engineering Institute (SEI), Carnegie Mellon University. 2004.

- [Chau, 2005] Chau, T.; Maurer, F.: "A case study of wiki-based experience repository at a medium-sized software company". En *Proceedings of the 3rd international conference on Knowledge capture (K-CAP)*, pp. 185-186, Banff, Canadá, 2-5 Octubre 2005. ACM Press.
- [Chiu, 2007] Chiu, C.M.; Chiu, C.S.; Chang, H.C.: "Examining the integrated influence of fairness and quality on learners' satisfaction and Web-based learning continuance intention". En *Information Systems Journal*, 17 (3), pp. 271-287, 2007.
- [Choi, 2008] Choi, B.; Poon, S. K.; Davis, J. G.: "Effects of knowledge management strategy on organisational performance: A complementarity theory-based approach". En *Omega – International Journal of Management Science*, 36 (2), pp. 235-251, 2008.
- [CMMI, 2010] Capability Maturity Model Integration for Development (CMMI-DEV) version 1.3. Technical Report CMU/SEI-2010-TR-033. Software Engineering Institute (SEI), Carnegie Mellon University. 2010.
- [Colomo-Palacios, 2011] Colomo-Palacios, R.; Fernandes, E.; Soto-Acosta, P.; Sabbagh, M.: "Software product evolution for Intellectual Capital Management: The case of Meta4 PeopleNet". En *International Journal of Information Management*, 31 (4), pp. 395-399, 2011.
- [Cuevas, 2002] Cuevas, G.: "Gestión del Proceso Software". Centro de Estudios Ramón Areces. 2002.
- [Cuevas, 2008] Cuevas, G.; San Feliu, T.; Calvo-Manzano, J.A.: "Change Management as a Key Factor for Improve the Process Deployment: A Case Study". En *Proceedings of the 15th European Systems and Software Process Improvement and Innovation (EuroSPI)*, pp. 13-23, Dublin, Irlanda, 3-5 Septiembre 2008.
- [Davenport, 2000] Davenport, T.H.; Prusak, L.: "Working knowledge: How organizations manage what they know". Harvard Business School Press, Boston, MA, USA. 2000.
- [Derniame, 2004] Derniame, J.C.; Oquendo, F.: "Key Issues and New Challenges in Software Process Technology". En *UPGRADE*, 5 (5), pp. 11-16, 2004.
- [Desouza, 2003] Desouza, K.C.: "Barriers to effective use of knowledge management systems in software engineering". En *Communications of the ACM*, 46 (1), pp. 99-101, 2003.
- [Dingsøy, 2004] Dingsøy, T.; Moe, N.B.: "The Process Workshop. A Tool to Define Electronic Process Guides in Small Software Companies". En *Proceedings of the Australian Software Engineering Conference (ASWEC)*, pp. 350-357, Melbourne, Australia, 13-16 Abril 2004.
- [Dingsøy, 2008] Dingsøy, T.; Moe, N.B.: "The Impact of Employee Participation on the Use of an Electronic Process Guide: A Longitudinal Case Study". En *IEEE Transactions on Software Engineering*, 34 (2), pp. 212-225, 2008.

- [Dingsøyr, 2009] Dingsøyr, T.; Bjørnson, F.O.; Shull, F.: "What Do We Know about Knowledge Management? Practical Implications for Software Engineering". En *IEEE Software*, 26 (3), pp. 100-103, 2009.
- [Dybå, 2005a] Dybå, T.; Kitchenham, B.A.; Jørgensen, M.: "Evidence-based software engineering for practitioners". En *IEEE software*, 22 (1), pp. 58-65, 2005.
- [Dybå, 2005b] Dybå, T.: "An Empirical Investigation of the Key Factors for Success in Software Process Improvement". En *IEEE Transactions on Software Engineering*, 31 (5), pp. 410-424, 2005.
- [Ebersbach, 2008] Ebersbach, A.; Glaser, M.; Heigl, R.: "Wiki: Web Collaboration". 2nd Edition. Springer. 2008.
- [EPF, 2012] Eclipse Foundation. Eclipse Process Framework Project.  
<http://www.eclipse.org/epf/>  
(último acceso: 30 mayo 2012).
- [Falbo, 2004] Falbo, R.A.; Borges, L.S.M.; Valente, F.F.R.: "Using Knowledge Management to Improve Software Process Performance in a CMM Level 3 Organization". En *Proceedings of the 4th International Conference on Quality Software (QSIC)*, pp. 162-169, Braunschweig, Alemania, 8-9 Septiembre 2004. IEEE Computer Society.
- [Fehér, 2006] Fehér, P.; Gábor, A.: "The role of knowledge management supporters in software development companies". En *Software Process Improvement and Practice*, 11 (3), pp. 251-260, 2006.
- [Fogle, 2001] Fogle, S.; Loulis, C.; Neuendorf, B.: "The Benchmarking Process: One Team's Experience". En *IEEE Software*, 18 (5), pp. 40-47, 2001.
- [Fokaefs, 2009] Fokaefs, M.; Bauer, K.; Stroulia, E.: "WikiDev 2.0: Web-based Software Team Collaboration". En *2009 ICSE Workshop on Wikis for Software Engineering (Wikis4SE)*, Vancouver, Canadá, 19 Mayo 2009.
- [Freeze, 2011] Freeze, T.R.; Kulkarni, U.: "Understanding the Composition of Knowledge Management Capability". Information Resources Management Association (Ed.), *Organization Learning and Knowledge: Concepts, Methodologies Tools and Applications*, Vol. 1. IGI Global, Hershey, PA, USA. 2011.
- [Fuggetta, 2005] Fuggetta, A.; Sfardini, L.: "Software Engineering Methods and Technologies". R.H. Thayer y M. Dorfman (Eds.), *Software Engineering*, Vol. 2: The Supporting Process, 3rd edition. IEEE Computer Society. 2005.
- [Garcia, 2004] Garcia, S.: "What is a Process Asset Library? Why Should You Care?". Aimware Professional Services Inc., Boston, MA, USA. 2004.

- [García, 2006] García, S.; Graettinger, C.P.; Kost, K.: "Proceedings of the 1st International Research Workshop for Process Improvement in Small Settings". Special Report CMU/SEI-2006-SR-001. Software Engineering Institute (SEI), Carnegie Mellon University, 2006.
- [García, 2011] García, J.; Amescua, A.; Sanchez-Segura, M.I.; Bermón, L.: "Design guidelines for software processes knowledge repository development". En *Information and Software Technology*, 53 (8), pp. 834-850, 2011.
- [Goffin, 2011] Goffin, K.; Koners, U.: "Tacit Knowledge, Lessons Learnt, and New Product Development". En *Journal of Product Innovation Management*, 28 (2), pp. 300-318, 2011.
- [Gupta, 2009] Gupta, A.; Jingyue, L.; Conradi, R; Rønneberg, H.; Landre, E.: "A case study comparing defect profiles of a reused framework and of applications reusing it". En *Empirical Software Engineering*, 14 (2), pp. 227-255, 2009.
- [Hall, 2002] Hall, T.; Rainer, A.; Baddoo, N.: "Implementing software process improvement: An empirical study". En *Software Process: Improvement and Practice*, 7 (1), pp. 3-15, 2002.
- [Handzic, 2004] Handzic, M.: "Knowledge Management: through the Technology Glass". World Scientific Publishing. 2004.
- [Haumer, 2006] Haumer, P.: "Increasing Development Knowledge with EPFC". En *Eclipse Review*, 1 (2), pp. 26-33, 2006.
- [He, 2007] He, J.; Yan, H.; Liu, C.; Jin, M.: "A Framework of Ontology-Supported Knowledge Representation in Software Process". En *Proceedings of the 2007 International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Chengdu, China, 15-16 Octubre 2007. Atlantis Press.
- [Heredia, 2012] Heredia, A.; Garcia-Guzman, J.; Amescua, A.; Sanchez-Segura, M.I.: "Interactive Knowledge Asset Management: Acquiring and Disseminating Tacit Knowledge". En *Journal of Information Science and Engineering*, 2012 (en publicación).
- [Humphrey, 1989] Humphrey, W.S.: "Managing the Software Process". Addison-Wesley Professional, Reading, MA, USA. 1989.
- [IPA, 2012] Osellus Company. IRIS Process Author.  
<http://www.osellus.com/IRIS-PA>  
(último acceso: 30 mayo 2012).
- [IPRC, 2006] The International Process Research Consortium: "A Process Research Framework". E. Forrester (Ed.). Software Engineering Institute (SEI), Carnegie Mellon University. 2006.
- [ISO, 2006] International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 15504. Information Technology – Process Assessment. 2006.

- [ISO, 2008] International Organization for Standardization and International Electrotechnical Commission. ISO/IEC 12207. Systems and Software Engineering – Software Life Cycle Processes. 2008.
- [Ivarsson, 2012] Ivarsson, M.; Gorschek, T.: “Tool support for disseminating and improving development practices”. En *Software Quality Journal*, 20 (1), pp. 173-199, 2012.
- [Jahn, 2011] Jahn, K.; Nielsen, P.A.: “A Vertical Approach to Knowledge Management: Codification and Personalization in Software Processes”. En *International Journal of Human Capital and Information Technology Professionals*, 2 (2), pp. 28-38, 2011.
- [Jedlitschka, 2008] Jedlitschka, A.; Ciolkowski, M.; Pfahl, D.: “Reporting Experiments in Software Engineering”. F. Shull, J. Singer y D.I.K. Sjøberg (Eds.), *Guide to Advanced Empirical Software Engineering*, Springer-Verlag London. 2008.
- [Juristo, 2001] Juristo, N.; Moreno, A.M.: “Basics of Software Engineering Experimentation”. Kluwer Academic Publishers, Springer. 2001.
- [Kamsu-Foguem, 2008] Kamsu-Foguem, B.; Coudert, T.; Béler, C.; Geneste, L.: “Knowledge formalization in experience feedback processes: An ontology-based approach”. En *Computers in Industry*, 59 (7), pp. 694-710, 2008.
- [Kellner, 1998] Kellner, M.I.; Becker-Kornstaedt, U.; Riddle, W.E.; Tomal, J.; Verlage, M.: “Process Guides. Effective Guidance for Process Participants”. En *Proceedings of the 5th International Conference on the Software Process*, pp. 11-25, Lisle, IL, USA, 14-17 Junio 1998. ISPA Press.
- [Kitchenham, 2002] Kitchenham, B.A. et al.: “Preliminary Guidelines for Empirical Research in Software Engineering”. En *IEEE Transactions on Software Engineering*, 28 (8), pp. 721-734, 2002.
- [Kitchenham, 2008] Kitchenham, B.A. et al.: “Evaluating guidelines for reporting empirical software engineering studies”. En *Empirical Software Engineering*, 13 (1), pp. 97-121, 2008.
- [Kneuper, 2002] Kneuper, R.: “Supporting Software Processes Using Knowledge Management”. S.K. Chang (Ed.), *Handbook of Software Engineering and Knowledge Engineering*, Vol. 2. World Scientific Publishing Company. 2002.
- [Komi-Sirviö, 2002] Komi-Sirviö, S.; Mäntyniemi, A.; Seppänen, V.: “Toward a Practical Solution for Capturing Knowledge for Software Projects”. En *IEEE Software*, 19 (3), pp. 60-62, 2002.
- [Kulpa, 2008] Kulpa M.K.; Johnson, K.A.: “Interpreting the CMMI: A Process Improvement Approach”. 2nd edition. Auerbach Publications. 2008.
- [Kurniawati, 2004a] Kurniawati, F.; Jeffery, R.: “The Long-term Effects of an EPG-ER in a Small Software Organisation”. En *Proceedings of the Australian Software Engineering Conference (ASWEC)*, pp. 128-137, Melbourne, Australia, 13-16 Abril 2004. IEEE Press.



- [Kurniawati, 2004b] Kurniawati, F.; Kitchenham, A.; Jeffery, R.: "Implementing a process model-based software development support environment: comparing an open source component with a proprietary tool". En *5th International Workshop on Software Process Simulation and Modeling (ProSim) – 26th International Conference on Software Engineering*, (2004/911), pp. 133-139, Edingburgh, Scotland, UK, 24-25 Mayo 2004.
- [Kurniawati, 2006] Kurniawati, F.; Jeffery, R.: "The Use and Effects of an Electronic Process Guide and Experience Repository". En *Information and Software Technology*, 48 (7), pp. 566-577, 2006.
- [Lin, 2007] Lin, K-J.: "Building Web 2.0". En *Computer Magazine*, 40 (5), pp. 101-102. IEEE Computer Society. Mayo 2007.
- [Liou, 2011] Liou, J.C.: "On Improving CMMI in an Immature World of Software Development". En *Journal of Information Science and Engineering*, Vol. 27 (1), pp. 213-226, 2011.
- [Louridas, 2006] Louridas, P.: "Using Wikis in Software Development". En *IEEE Software*, 23 (2), pp. 88-91, 2006.
- [Maier, 2007] Maier, R.: "Knowledge Management Systems: Information and Communication Technologies for Knowledge Management". 3rd edition. Springer. 2007.
- [Majchrzak, 2006] Majchrzak, A.; Wagner, C.; Yates, N.: "Corporate Wiki users: Results of a survey". En *Proceedings of the 2006 international symposium on Wikis (WikiSym)*, pp. 99-104, Odense, Dinamarca, 21-23 Agosto 2006. ACM Press.
- [Mathiassen, 2005] Mathiassen, L.; Ngwenyama, O.K.; Aaen, I.: "Managing Change in Software Process Improvement". En *IEEE Software*, 22 (6), pp. 84-91, 2005.
- [McFeeley, 1996] McFeeley, B.: "IDEAL: A User's Guide for Software Process Improvement". Handbook CMU/SEI-96-HB-001. Software Engineering Institute (SEI), Carnegie Mellon University. 1996.
- [Medina, 2010] Medina, F.: "Marco Metodológico para la Mejora de la Eficiencia de Uso de los Procesos Software". Tesis doctoral. Universidad Carlos III de Madrid. 2010.
- [Moe, 2005] Moe, N.B.; Dingsøyr, T.: "The Impact of Process Workshop Involvement on the Use of an Electronic Process Guide". En *Conference on Software Engineering and Advanced Applications, 31st EUROMICRO*, pp. 188-195, Oporto, Portugal, 30 Agosto - 3 Septiembre 2005.
- [Moe, 2006] Moe, N.B.; Dybå, T.: "The Use of an Electronic Process Guide in a Medium-sized Software Development Company". En *Software Process Improvement and Practice*, 11, pp. 21-34, 2006.
- [MSF, 2012] Microsoft Solutions Framework. Herramientas y plantillas de Visual Studio. <http://msdn.microsoft.com/es-es/vstudio/aa718795> (último acceso: 30 mayo 2012).

- [Murugesan, 2007] Murugesan, S.: "Understanding Web 2.0". En *IT Professional*, 9 (4), pp. 34-41. IEEE Computer Society. Julio-Agosto 2007.
- [Musser, 2006] Musser, J.; O'Reilly, T.: "Web 2.0 Principles and Best Practices". En *O'Reilly Radar*. Technical Report. Noviembre 2006.
- [Natali, 2002] Natali, A.C.C.; Falbo, R.A.: "Knowledge Management in Software Engineering Environments". En *Proceedings of the 16th Brazilian Symposium on Software Engineering (SBES)*, pp. 238-253, Gramado, Brazil, 16-18 Octubre 2002.
- [Niazi, 2006] Niazi, M.; Wilson, D.; Zowghi, D.: "Critical Success Factors for Software Process Improvement Implementation: An Empirical Study". En *Software Process: Improvement and Practice*, 11 (2), pp. 193-211, 2006.
- [Nonaka, 1994] Nonaka, I.: "A Dynamic Theory of Organizational Knowledge Creation". En *Organization Science*, 5 (1), pp. 14-37, 1994.
- [Nonaka, 1995] Nonaka, I.; Takeuchi, H.: "The knowledge creating company: how Japanese companies create the dynamics of innovation". Oxford University Press. 1995.
- [Nonaka, 2006] Nonaka, I.; von Krogh, G.; Voelpel, S.: "Organisational knowledge creation theory: Evolutionary paths and future advances". En *Organisation Studies*, 27 (8), pp. 1179-1208, 2006.
- [Nonaka, 2009] Nonaka, I.; von Krogh, G.: "Tacit Knowledge and Knowledge Conversion: Controversy and Advancement in Organizational Knowledge Creation Theory". En *Organization Science*, 20 (3), pp. 635-652, 2009.
- [OMG, 2008] Object Management Group. "Software & Systems Process Engineering Meta-Model Specification (SPEM) version 2.0". 2008.
- [Petter, 2008] Petter, S.; Vaishnavi, V.: "Facilitating experience reuse among software project managers". En *Information Sciences*, 178 (7), pp. 1783-1802, 2008.
- [Phongpaibul, 2007] Phongpaibul, M.; Koolmanojwong, S.; Lam, A.; Boehm, B.: "Comparative Experiences with Electronic Process Guide Generator Tools". En *Proceedings of the International Conference of Software Process, 2007*. Software Process Dynamics and Agility, Q. Wang, D. Pfahl y D.M. Raffo (Eds.), LNCS 4470, pp. 61-72, Springer Berlin / Heidelberg, 2007.
- [Phuwanartnurak, 2009] Phuwanartnurak, A.J.: "Interdisciplinary Collaboration through Wikis for Software Engineering". En *2009 ICSE Workshop on Wikis for Software Engineering (Wikis4SE)*, Vancouver, Canadá, 19 Mayo 2009.
- [Pino, 2008] Pino, F.J.; García, F.; Piattini, M.: "Software process improvement in small and medium software enterprises: a systematic review". En *Software Quality Journal*, 16 (2), pp. 237-261, 2008.
- [Polanyi, 1966] Polanyi, M.: "The tacit Dimension". Doubleday & Company. 1966.

- [Radziwill, 2004] Radziwill, N.M.; Shelton, A.M.: "TWiki as a Platform for Collaborative Software Development Management". En *Proceedings of SPIE*, volumen 5496, Glasgow, Scotland, 21-26 Junio 2004. Advanced Software, Control, and Communication Systems for Astronomy, Lewis H. y Raffi G. (Eds.), pp. 609-617, SPIE Press.
- [Ras, 2009] Ras, E.: "Investigating Wikis for Software Engineering. Results of Two Case Studies". En *2009 ICSE Workshop on Wikis for Software Engineering (Wikis4SE)*, Vancouver, Canadá, 19 Mayo 2009.
- [Rech, 2007] Rech, J.; Bogner, C.; Haas, V.: "Using Wikis to Tackle Reuse in Software Projects". En *IEEE Software*, 24 (6), pp. 99-104, 2007.
- [Rico, 2004] Rico, D.F.: "ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers". J-Ross Publishing. 2004.
- [Riechert, 2009] Riechert, T.; Berger, T.: "Leveraging Semantic Data Wikis for Distributed Requirements Elicitation". En *2009 ICSE Workshop on Wikis for Software Engineering (Wikis4SE)*, Vancouver, Canadá, 19 Mayo 2009.
- [RMC, 2012] IBM. Rational Method Composer.  
<https://www.ibm.com/developerworks/rational/products/rup/>  
(último acceso: 30 mayo 2012).
- [Rus, 2002] Rus, I.; Lindvall, M.: "Knowledge Management in Software Engineering". En *IEEE software*, 19 (3), pp. 26-38, 2002.
- [Schneider, 2003] Schneider, K; von Hunnius, J.: "Effective Experience Repositories for Software Engineering". En *Proceedings of the 25th International Conference on Software Engineering (ICSE)*, pp. 534-539, Portland, OR, USA, 3-10 Mayo 2003. IEEE.
- [Schuster, 2007] Schuster, N.; Zimmerman, O.; Pautasso, C.: "ADkwik: Web 2.0 Collaboration System for Architectural Decision Engineering". En *Proceedings of the 19th International Conference on Software Engineering and Knowledge Engineering (SEKE)*, pp. 255-260, Boston, MA, USA, 9-11 Julio 2007.
- [Scorta, 2009] Scorta, I.: "A Knowledge Management Practice Investigation in Romanian Software Development Organisations". En *WSEAS Transactions on Computers*, 3 (8), pp. 459-468, 2009.
- [Scott, 2001] Scott, L.; Carvalho, L.; Jeffery, R.; D'Ambra, J.: "An Evaluation of the Spearmint Approach to Software Process Modelling". En *Proceedings of the 8th European Workshop on Software Process Technology (EWSPT)*. Software Process Technology, V. Ambriola (Ed.), LNCS 2077, pp. 77-89, Springer Berlin / Heidelberg, 2001.
- [Scott, 2002a] Scott, L.; Kurniawati, F.: "WAGNER – Web-Based Process Guide and Experience Repository". Centre for Advanced Software Engineering Research (CAESER). Report No. 02/1, University of New South Wales, Sydney, Australia, 2002.

- [Scott, 2002b] Scott, L.; Carvalho, L.; Jeffery, R.; D'Ambra, J.; Becker-Kornstaedt, U.: "Understanding the Use of an Electronic Process Guide". En *Journal of Information and Software Technology*, 44 (10), pp. 601-616, 2002.
- [Seaman, 1999] Seaman, C.B.: "Qualitative Methods in Empirical Studies of Software Engineering". En *IEEE Transactions on Software Engineering*, 25 (4), pp. 557-572, 1999.
- [SEI, 2006] CMMI Executive Overview. Software Engineering Institute (SEI), Carnegie Mellon University. 2006.
- [Sircar, 2009] Sircar, S.; Choi J.: "A study of the impact of information technology on firm performance: a flexible production function approach". En *Information Systems Journal*, 19 (3), pp. 313-339, 2009.
- [Solís, 2009] Solís, C.; Alí, N.; Babar, M.A.: "A Spatial Hypertext Wiki for Architectural Knowledge Management". En *2009 ICSE Workshop on Wikis for Software Engineering (Wikis4SE)*, Vancouver, Canadá, 19 Mayo 2009.
- [Strong, 2008] Strong, B.; Davenport, T.H.; Prusak, L.: "Organizational governance of knowledge and learning". En *Knowledge and Process Management*, 15 (2), pp. 150-157, 2008.
- [SWEBOK, 2004] SWEBOK. Guide to the Software Engineering Body of Knowledge. A Project of the IEEE Computer Society. Professional Practices Committee. IEEE Computer Society. 2004.
- [Visaggio, 2009] Visaggio, G.: "Knowledge Base and Experience Factory for Empowering Competitiveness". En *Software Engineering: International Summer Schools (ISSSE), Revised tutorial lectures 2006-2008*, Salerno, Italia. Software Engineering, A. Lucia y F. Ferrucci (Eds.), LNCS 5413, pp. 223-256, Springer Berlin / Heidelberg, 2009.
- [Wangenheim, 2006] von Wangenheim, C.G.; Weber, S.; Rossa Hauck, J.C.; Trentin, G.: "Experiences on Establishing Software Processes in Small Companies". En *Information and Software Technology*, 48, pp. 890-900, 2006.
- [Wohlin, 2000] Wohlin, C.; Runeson, P.; Höst, M.; Ohlsson, M.C.; Regnell, B.; Wesslén, A.: "Experimentation in software engineering: an introduction". Kluwer Academic Publishers, Springer. 2000.
- [Wohlin, 2001] Wohlin, C.; Höst, M.: "Controlled experiments in software engineering (Special section)". En *Information and Software Technology*, 43 (15), pp. 921-924, 2001.
- [Xiao, 2007] Xiao, W.; Chi, C.; Yang, M.: "On-line collaborative software development via wiki". En *Proceedings of the 2005 international symposium on Wikis (WikiSym)*, pp. 177-183, Montreal, Canadá, 21-23 Octubre 2007.

# |Anexos



## Anexo A: Criterios para la Evaluación de Productos

Para la evaluación de la calidad de los productos desarrollados por los distintos equipos de trabajo en sus respectivos proyectos durante el experimento se utilizaron los criterios que se indican en la siguiente tabla.

Criterio	Valor
<b>Historias de usuario (15 puntos)</b>	
Se proporciona una lista incluyendo todos los objetivos de la iteración	1
Se ha establecido la prioridad de cada objetivo identificado	3
Se ha estimado el tamaño de cada objetivo identificado	3
Se ha estimado el esfuerzo de cada objetivo identificado	3
Se ha establecido la información detallada de cada objetivo identificado	5
<b>Pruebas de aceptación (10 puntos)</b>	
Se han identificado una o varias pruebas para cada objetivo de la iteración	2
Las pruebas identificadas permiten la detección de errores potenciales	2
Se ha especificado el propósito de cada prueba	2
Se han especificado los pasos para la ejecución de cada prueba	4
<b>Planificación adaptativa (13 puntos)</b>	
Se ha determinado la máxima carga de trabajo del equipo de proyecto	3
Se ha determinado el alcance de la iteración	1
Se han identificado las tareas necesarias para alcanzar los objetivos de la iteración	1
Cada tarea está relacionada con un y solo un objetivo de la iteración	1
Las tareas indican los productos que deben obtenerse para certificar su finalización	3
Cada tarea tiene un identificador único y un responsable de la realización del trabajo	1
Se ha estimado (en horas) el esfuerzo necesario para realizar cada tarea	3
<b>Diseño simple (12 puntos)</b>	
Se ha dibujado un diagrama de secuencia para cada objetivo de la iteración	3
Se han diseñado todas las clases representadas en los diagramas de secuencia	5
Todas las clases del código fuente figuran en al menos un diagrama de secuencia	1
El código fuente ha sido implementado de acuerdo con los diagramas de secuencia	3
<b>Estándares de codificación (2 puntos)</b>	
El código fuente se ha implementado cumpliendo los estándares de codificación	2
<b>Reorganización de código (8 puntos)</b>	
Hay evidencia de la aplicación de técnicas de reorganización en el código fuente	4
El código fuente implementa el sistema de manejo de excepciones propuesto	3
El código fuente implementa los patrones propuestos	1
<b>Pruebas unitarias (41 puntos)</b>	
Los casos de prueba necesarios para verificar la correcta implementación de los objetivos del grupo 1 han sido identificados y desarrollados	15
Los casos de prueba necesarios para verificar la correcta implementación de los objetivos del grupo 1 han sido identificados y desarrollados	5
Los casos de prueba necesarios para verificar la correcta implementación de los objetivos del grupo 1 han sido identificados y desarrollados	9
Los casos de prueba necesarios para verificar la correcta implementación de los objetivos del grupo 1 han sido identificados y desarrollados	12
<b>Propiedad colectiva de código (5 puntos)</b>	
La última versión del código y de las pruebas está almacenada en el repositorio	1
La última versión de los artefactos auxiliares está almacenada en el repositorio	1
La última versión de todos los ficheros del proyecto está almacenada en el repositorio	1
El historial de cambios en el repositorio muestra un uso activo del mismo	2

Criterio	Valor
<b>Integración continua (8 puntos)</b>	
Se ha creado un fichero que permite la automatización de las tareas de integración	1
Los objetivos del fichero de configuración de la integración automatizada permiten:	
- La generación de una DLL con la lógica de negocio implementada	1
- La generación de una DLL con las pruebas unitarias de la lógica de negocio	1
- La ejecución de las pruebas unitarias de la lógica de negocio	1
- La publicación de los resultados de la ejecución de las pruebas unitarias	1
- La generación de un ejecutable enlazando la DLL de la lógica de negocio y la correcta compilación de todos los elementos gráficos de la interfaz	1
Existe un historial de los resultados de cada intento de integración	1
Los ficheros obtenidos tras una integración satisfactoria quedan disponibles vía Web para todos los miembros del equipo de proyecto (incluyendo los instructores)	1
<b>Seguimiento del proyecto (12 puntos)</b>	
Los informes muestran la evolución del trabajo y de las tareas dentro del alcance de la iteración en las fechas establecidas como hitos	3
Todos los informes contienen información actualizada sobre el resultado de la ejecución de cada prueba de aceptación	3
Todos los informes contienen información actualizada sobre el estado actual de las tareas necesarias para alcanzar los objetivos de la iteración	3
Todos los informes contienen información para el seguimiento preventivo de la iteración obtenida mediante el análisis del valor ganado	3



## Anexo B: Plantillas para Editar el Conocimiento Pre-Existente

En este anexo se muestran las distintas plantillas diseñadas con Infopath para gestionar los contenidos de los distintos tipos de activos almacenados como conocimiento pre-existente en el repositorio de conocimiento organizacional.

### Plantilla para la definición de activos

**Página Principal**

**Identificador**

  
**Nombre**  
**Contenido**

Texto

**Página Secundaria**

**Identificador**

  
**Nombre**  
**Contenido**

Texto

Insertar elemento

## Plantilla para activos de tipo elemento de trabajo

### Nombre

### Título de la Visión General

### Visión General

### Estados y Transiciones

Ruta Relativa de la Imagen del Diagrama:

Texto al Pie de la Imagen:

#### Estado

Identificador:

Nombre:

Descripción:

#### Transición

Identificador del Estado Destino:

Razones:

Nombre	Descripción
<input type="text"/>	<input type="text"/>

Insertar elemento

Insertar elemento

Insertar elemento

### Actividades en los Estados

#### Estado

#### Actividades en el Estado

Referencia

Insertar elemento

Insertar elemento

### Campos

Campo	Descripción
<input type="text"/>	<input type="text"/>

Insertar elemento

## Plantilla para activos de tipo proceso

### Nombre

### Visión General

### Enlaces

**Enlace**

Tipo de enlace:

Elemento de Trabajo enlazado:

Insertar elemento

### Roles Participantes

**Responsables**

Insertar elemento

**Encargados**

Insertar elemento

**Consultados**

Insertar elemento

**Informados**

Insertar elemento

**Propietarios**

Insertar elemento

### Criterios de Entrada

**Criterio de Entrada**

Título:

Descripción:

Insertar elemento

Cuándo:

## Actividades

**Actividad**

Nombre:

Descripción:

Insertar elemento

## Criterios de Salida

**Título**

Insertar elemento

## Plantilla para activos de tipo actividad

### Nombre

### Visión General

## Enlaces

**Enlace**

Tipo de enlace:

Producto de Trabajo enlazado:

Insertar elemento

### Roles Participantes

<b>Responsables</b>
<input type="text"/>
<input type="checkbox"/> Insertar elemento
<b>Encargados</b>
<input type="text"/>
<input type="checkbox"/> Insertar elemento
<b>Consultados</b>
<input type="text"/>
<input type="checkbox"/> Insertar elemento
<b>Informados</b>
<input type="text"/>
<input type="checkbox"/> Insertar elemento
<b>Propietarios</b>
<input type="text"/>
<input type="checkbox"/> Insertar elemento

### Criterios de Entrada

<b>Criterio de Entrada</b>
Título: <input type="text"/>
Descripción: <input type="text"/>
<input type="checkbox"/> Insertar elemento
Cuándo: <input type="text"/>

### Tareas

<b>Tarea</b>
Título: <input type="text"/>
Descripción: <input type="text"/>
<input type="checkbox"/> Insertar elemento

### Criterios de Salida

<b>Título</b>
<input type="text"/>
<input type="checkbox"/> Insertar elemento

## Plantilla para activos de tipo rol

### Identificador

### Nombre

### Título

### Visión General

## Plantilla para activos de tipo producto de trabajo

### Nombre

### Enlace al Documento

Si el valor del Enlace al Documento no se corresponde con la ubicación del documento del Producto de Trabajo seleccionado, por favor revise el identificador asignado al Producto de Trabajo en la lista de productos de trabajo.

## Plantilla para activos de tipo instrucción técnica

### Nombre

Esta plantilla solo se utiliza para crear el fichero XML que contendrá la Instrucción Técnica, manteniendo la consistencia en el nombrado de los identificadores.

Una vez creado el fichero, el texto de la Instrucción Técnica se deberá introducir dentro de las etiquetas <Body></Body> y en formato HTML, para lo cual se usará el bloc de notas. A partir de ese momento, el fichero correspondiente no se podrá abrir con Infopath, por lo que es recomendable eliminar la asociación del fichero con esta plantilla y editar el fichero en modo texto.

Para que el fichero XML admita caracteres latinos, como acentos o eñes, se deberá editar el fichero con el bloc de notas y cambiar el valor del atributo "encoding" a "ISO-8859-1".