

Unidades de Reutilización de Software por 15.000 €

Presentación



Tipo Documento :	Promocional
Autor :	The Reuse Company
Departamento :	Marketing
Fecha Creación :	01/05/2006
Última Modificación :	01/05/2006
Referencia Doc. :	

ÍNDICE

1	INTRODUCCIÓN.....	3
2	LA REUTILIZACIÓN DE SOFTWARE	4
2.1	DEFINICIONES	4
2.2	¿QUÉ SE ENTENDÍA POR REUTILIZACIÓN?	4
2.3	¿QUÉ SE DEBERÍA ENTENDER HOY DÍA POR REUTILIZACIÓN?.....	5
3	UNIDAD DE REUTILIZACIÓN DEL CONOCIMIENTO	8
3.1	SU MISIÓN.....	8
3.2	PROS Y CONTRAS DE LA REUTILIZACIÓN DE ACTIVOS	8
3.2.1	Obstáculos a la reutilización.....	8
3.2.2	¿Cuál es el conocimiento a tratar?	9
3.2.3	¿Cuál es el ciclo de vida de este conocimiento?	9
3.2.4	¿Qué es una ontología y cómo se crea?.....	12
3.3	SU ESTRUCTURA	13
3.3.1	Arquitecto de dominios.....	13
3.3.2	Experto de dominio.....	13
3.3.3	Catalogador	13
3.3.4	Jefe de Unidad – Reuse Manager.....	13
3.4	SUS ACTIVIDADES PRINCIPALES	13
3.4.1	Gestión.....	14
3.4.2	Planificación.....	14
3.4.3	Ejecución.....	14
3.4.4	Evaluación	15
3.5	RELACIÓN CON OTROS DEPARTAMENTOS	15
4	HERRAMIENTAS Y METODOLOGÍA	16
4.1	METODOLOGÍA DE REUTILIZACIÓN: IRM	16
4.1.1	Introducción.....	16
4.1.2	El proceso.....	17
4.2	HERRAMIENTAS DE REUTILIZACIÓN.....	19
4.2.1	swREUSER	19
4.2.2	Reutilización de modelos Rational Rose	20
4.2.3	REUSE Server.....	20
4.2.4	domainREUSER	20
5	QUÉ ES LA ‘UNIDAD DE REUTILIZACIÓN DE CONOCIMIENTO POR 15.000€’ KU15K..	21
6	REFERENCIAS	22

1 INTRODUCCIÓN

“El Conocimiento es Poder”¹. Esta es una de las frases más comúnmente repetida en muchos de los ámbitos de la vida moderna. Por esta razón, The Reuse Company considera que la ausencia de una Unidad de Conocimiento en todas y cada una de las organizaciones es un aspecto que hace que dichas organizaciones no alcancen las cotas que inicialmente se esperaba de ellas.

¿Qué es la Unidad de Conocimiento? ¿Cómo personalizarlo para empresas centradas en la producción de software (en una unidad de reutilización de SW)? ¿Cuáles son sus objetivos? ¿Cómo poner dicha unidad en marcha? Éstas y otras preguntas serán el contenido del presente documento.

Antes de comenzar a describir esta unidad, vamos a ver aspectos relacionados con la reutilización de software (una de sus principales misiones). Lo veremos tanto en su aproximación más clásica, como también se abordarán las más modernas técnicas de reutilización de software.

¹ “El Conocimiento es Poder”: Sir Francis Bacon, S. XVI

2 LA REUTILIZACIÓN DE SOFTWARE

2.1 Definiciones

Afortunadamente el desarrollo de software está dejando de ser un arte, para convertirse definitivamente en una ingeniería. En la misma línea, las técnicas de reutilización de software utilizadas en las pasadas décadas, han quedado desfasadas, y difícilmente pueden ser aplicadas a los modernos entornos de trabajo.

¿Qué se considera reutilización? Comencemos por unas definiciones:

“Se define la reutilización de software como el uso de cualquier tipo de artefacto (también llamado activo), o parte del mismo, creado con anterioridad, en un nuevo proyecto.” [IEEE]

Y para precisar aún más la reutilización implica que:

- » Cada artefacto reutilizado en el nuevo proyecto ha sido ligeramente modificado para ajustarse a una definición de problema, o a una necesidad de solución en el nuevo proyecto, o bien
- » Cada artefacto reutilizable fue creado para funcionar en un contexto, entorno o aplicación completamente diferente a que se le va a dar en el nuevo proyecto.
- » Cada artefacto reutilizado ha necesitado ser configurado para adaptarse a las especificaciones o requerimientos del nuevo proyecto.

Las buenas prácticas asociadas a la reutilización de software prometen:

- » Un aumento de la Productividad, mediante la mejora de los tiempos en los que se desarrollan los nuevos proyectos informáticos (Time to Market -TTM).
- » Un aumento de la calidad de los productos.
- » Una reducción de los costes de desarrollo.
- » Mejoras en las actividades de Mantenimiento y soporte de aplicación.
- » Mejoras en las actividades de control y planificación por la reducción de desviaciones en los desarrollos.

2.2 ¿Qué se entendía por reutilización?

Hasta hace poco tiempo, a todos se nos había enseñado que escribir código directamente a partir de un documento que describía vagamente las necesidades de usuario no era una buena idea. Sabíamos que teníamos que comenzar por una correcta captura de requisitos, identificar todos los interesados en el proyecto (stakeholders), modelar el negocio donde va a residir la futura aplicación, realizar diagramas de análisis, luego de diseño... Sin embargo, ¿se llevaban a cabo todas estas tareas?

La respuesta a esta pregunta resultaba ser un rotundo NO. Y prueba de ello puede ser, por ejemplo, The Chaos Report [CHAO].

Con esto en mente, ¿cómo era la reutilización de software que se aplicaba en aquellos tiempos? Claramente pasaba por reutilizar código fuente? Sin embargo, reutilizar código fuente es claramente problemático debido a:

- » Alguien que quiera reutilizar código en antiguas plataformas de desarrollo (COBOL, FORTRAN, Lisp, C, ...) estaría perdido en los tiempos actuales
- » Algo más actual, los que apostaron por reutilización basada en componentes COM, CORBA [JACO] se encuentran hoy día con que la tecnología actual nos orienta hacia los Web Services: ¿cuál será dentro de 5 años?

Es obvio que debemos apuntar algo más arriba a la hora de reutilizar software. Es decir, cuanto más arriba subamos en el ciclo de vida, más valor tiene el activo a reutilizar, puesto que se ha requerido un mayor poder de abstracción para crearlo; y más independientemente somos de la plataforma tecnológica, lo que evita riesgos de obsolescencia antes de la reutilización.

2.3 ¿Qué se debería entender hoy día por reutilización?

Naveguemos pues aguas arriba, partamos del código fuente, y veamos qué otros activos de software podemos reutilizar. Los diseños de nuestros sistemas podrían ser la primera parada en este viaje. A la hora de reutilizar diseños vemos que ya hay mucho camino recorrido, los famosos patrones de diseño [GAMM] son unas ideas estupendas que podemos reutilizar.

Segunda parada, diagramas de análisis. A este respecto es importante hablar de la reutilización de patrones de análisis [FOWL].

Ambos tipos de patrones han de expresarse en algún lenguaje que permita describirlos. Gamma en su famoso libro ya indicó qué elementos debería incluir este tipo de lenguajes. Sin embargo, el elemento más significativo de estos lenguajes está expresando en lenguaje UML, para ser más concretos, en forma de diagramas de clase.

Reutilizar estos patrones implica:

1. Identificar los patrones
2. Crear un catálogo de patrones
3. Clasificar cada patrón con respecto a dicho catálogo
4. Dar a conocer el catálogo a todo el personal de mi organización
5. Poder buscar patrones relevantes

La tarea para la reutilización de estos patrones se antoja posible, siempre que pudiera buscarlos en mi repositorio: sin embargo, no todo el conocimiento que mi organización ha almacenado en el lenguaje UML se ha expresado en forma de patrones tan correctamente identificados y clasificados. De hecho, pese a que les duela a muchos modernistas, la creación de SW se sigue realizando mediante procesos de desarrollo de software (SDP) basados en los clásicos proyectos, y utilizando metodologías clásicas espiral, cascada, etc. Por este motivo, con vistas a mejorar mi nivel de reutilización debería ser capaz de acceder a conocimiento encerrado en proyectos previos de los que no he extraído ningún tipo de patrón.

La solución a este dilema es sencilla, necesitamos algo similar (aunque con muchas otras características) a uno de esos motores de búsqueda que ahora parecen estar tan de moda. Pero, ninguno de ellos entiende información expresada en lenguaje UML. Es decir, un sistema moderno de

reutilización debería permitir a nuestras organizaciones introducir información expresada en UML en un repositorio, para posteriormente recuperarla.

Si seguimos en nuestro camino virtual por el ciclo de vida, nos encontramos las especificaciones de requisitos. En este sentido, al igual que en otros, podemos hacer una división entre reutilización horizontal y vertical. Por reutilización horizontal nos referimos a poder reutilizar requisitos entre aplicaciones enmarcadas en distintos dominios (farmacia, telecomunicaciones..., en este sentido, los requisitos de seguridad son un buen ejemplo de reutilización horizontal); mientras que por reutilización vertical nos referimos a reutilizar requisitos dentro de un mismo dominio. En este momento vemos que podemos introducir un nuevo concepto denominado patrones de requisitos, del cual nos sentimos muy orgullosos [HURT]. Según nuestra apreciación, un patrón de requisitos debe incluir:

- » La especificación de los requisitos del patrón
- » Los riesgos inherentes a los requisitos anteriores
- » Las pruebas a las que dichos requisitos han de ser sometidas
- » Otro tipo de información, al estilo de los patrones de diseño, que incluya descripción, intención, motivación, aplicabilidad...

Si somos capaces de crear este tipo de patrones en nuestras organizaciones, seremos capaces de reutilizar uno de los activos de mayor importancia en una empresa de desarrollo de software.

La última parada corresponde al modelado de dominios [PRIE]. Especialmente indicado para aquellas compañías que recurrentemente desarrollan aplicaciones para un mismo dominio. El modelado de dominio permite representar el conocimiento que una organización posee sobre un área de conocimiento. Las formas de expresar este conocimiento pueden ser diversas, aunque desde The Reuse Company recomendamos la utilización de ontologías [ONTO]. Disponer de una representación del dominio permite:

- » Tener un vocabulario común entre todos los interesados en el proyecto, tanto por parte del equipo de desarrollo como por parte del cliente
- » Poder normalizar todos los nombres de elementos en los análisis, código... Esto redundará en una mayor capacidad de reutilización de análisis y diseños
- » Poder utilizar nombres correctos a la hora de expresar requisitos, lo que también mejorará su capacidad de ser reutilizados
- » Poder traspasar de forma rápida conocimiento, desde la capa de conocimiento ontológico, hasta la capa de análisis/diseño
- » La perfecta clasificación de cualquier tipo de documento generado en el seno del proyecto
- » Permitir que nuevos desarrolladores se unan al equipo y dominen el área de aplicación del desarrollo de forma rápida

» ...

El camino parece que ya ha finalizado, sin embargo, todavía no hemos hablado del invitado especial, la Trazabilidad entre elementos. Para que el hecho de reutilizar un elemento tenga la mayor potencia posible, necesitamos que quién haya desarrollado el proyecto que queremos reutilizar haya trazado correctamente todos sus elementos. Esto implica, por ejemplo, trazar requisitos con casos de uso, requisitos con riesgos, casos de uso con clases, casos de uso con pruebas...

Esto permite que cuando se desee reutilizar un activo de software, las propias herramientas CASE sugieran qué datos y elementos son también susceptibles de ser reutilizados y, por supuesto, trazados en el nuevo proyecto.

Por último, ni que decir tiene, la importancia del control de este proceso, es decir, sabes de cuántos activos se dispone, cuáles de ellos se reutilizan más, quién ha reutilizado cada activo, ante una modificación (por evolución o por anomalía) de un activo, notificar automáticamente a todos los desarrolladores que lo han utilizado...

3 UNIDAD DE REUTILIZACIÓN DEL CONOCIMIENTO

Una vez que sabemos qué se considera por reutilización en el Siglo XXI, vamos a ver qué es una Unidad de Reutilización de Conocimiento en una organización dedicada al desarrollo de software; cuál es su misión, estructura y sus relaciones con otros departamentos.

3.1 Su misión

Comencemos el viaje previo a la institucionalización de una Unidad de Reutilización de Software por la definición de los objetivos y misión del mismo. Estos objetivos, como es lógico, deben ir en consonancia con los objetivos de la organización y a la descripción dada sobre reutilización de software.

Misión:

- Gestionar de forma adecuada el conocimiento que se genera en el seno de la organización, inicialmente en el ámbito del Software
- Permitir una reutilización sistemática de los activos de software gestionados

3.2 Pros y contras de la reutilización de activos

La perfecta gestión de activos es, y ha sido siempre, el objetivo más importante a conseguir, ya que constituye la actividad principal de las organizaciones. La reutilización sistemática no es una misión tan clara en las organizaciones del Siglo XX, pero sí que se torna vital en las del Siglo XXI ya que permite:

- » Incrementar la calidad del software desarrollado
- » Reducir el Time To Market
- » Un ahorro considerable de costes
- » Aporta una metodología cuando se carece de ella

Sin embargo, hay que tener en cuenta que acometiendo actividades de reutilización sin una correcta planificación y control, se puede incurrir en ciertos riesgos:

- » Grandes inversiones (financieras y de otros tipos) iniciales que no dan lugar a su retorno
- » Errores tácticos que descomponen tanto la estructura general de la organización como sus procesos de desarrollo
- » Las actividades de reutilización son difíciles de medir

3.2.1 Obstáculos a la reutilización

En las culturas organizativas actuales hay una serie de obstáculos que dificultan la adhesión a políticas activas y sistemáticas de reutilización de software. Se detallan a continuación algunos de ellos:

- » Síndrome de 'No inventado aquí': la sociedad competitiva que nos ha tocado vivir pone trabas a utilizar cualquier cosa que no haya sido creada por nosotros.

- Siempre creemos que lo nuestro será mejor, por lo que en lugar de reutilizar, muchas veces preferimos comenzar de nuevo el desarrollo de cualquier sistema
- » Síndrome 'Recuerda chaval, está prohibido copiar': parece que esta frase nos marca incluso desde cuando estudiábamos primaria. Esto nos puede hacer ver que copiar es malo, cuando en realidad es el eje fundamental para la evolución
 - » Procesos de desarrollo de software no maduros: muchas organizaciones carecen de un proceso de desarrollo de software maduro, por lo que les asusta abordar cualquier iniciativa como la de la reutilización. En realidad, una política correcta de reutilización debería, por un lado, ser flexible para poder adaptarse a organizaciones con procesos maduros; y, por otro lado, dotar de cierto orden en aquellas organizaciones que se encuentren sumidas en el caos
 - » Inversión inicial: como la mayoría de las cosas buenas de la vida, adoptar políticas de reutilización requieren una inversión inicial
 - » Adaptación de las herramientas actuales: sí, es cierto, de nada vale un plan perfecto para la reutilización de software si sus herramientas no lo soportan

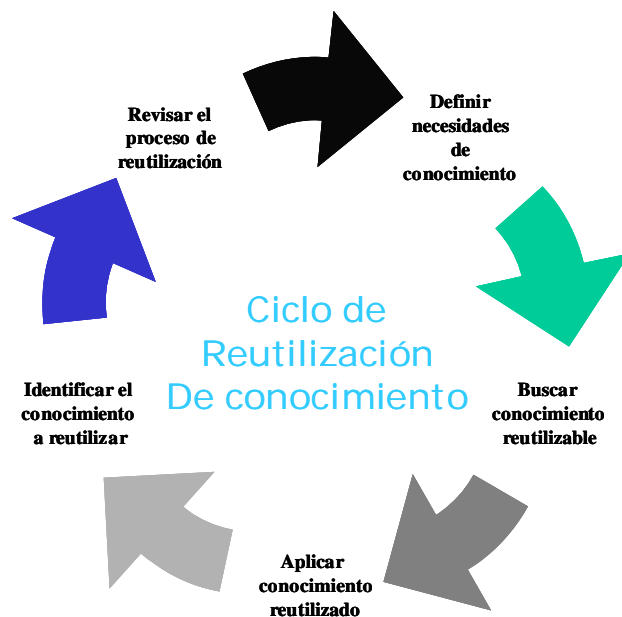
Descubra gracias a los servicios de The Reuse Company la mejor forma de evitar mitos, mejorar sus procesos y adecuarlos a la reutilización de software, reducir inversiones iniciales, acercar retornos de inversión y dotar a su gente de las herramientas adecuadas. Y todo esto a bajo coste.

3.2.2 ¿Cuál es el conocimiento a tratar?

Tanto para este tipo de organizaciones, como para cualquier organización, conocimiento es cualquier activo que se crea, transmite o almacena dentro de la misma. Por esto, conocimiento en un tipo de organización que desarrolla SW es un código fuente, pero es mucho más un diseño, un análisis, una especificación de requisitos, un conjunto de pruebas, unas métricas, un documento de un usuario final, un e-mail, tu página web, la página web de tus clientes, las páginas web de tus competidores, páginas web de la administración pública, bases de datos sobre patentes, artículos científicos y un largo etcétera. Si no somos capaces de gestionar toda esta información, estamos perdiendo una oportunidad valiosísima de negocio.

3.2.3 ¿Cuál es el ciclo de vida de este conocimiento?

El ciclo de vida del conocimiento, para The Reuse Company, es el siguiente:



Antes de describir las diferentes fases que forman el ciclo de reutilización de conocimiento, creemos esencial definir que dicho ciclo se aplica directamente imbricado en el proceso de desarrollo de Software (SDP) de cada proyecto. Es decir, nuestro ciclo debe ser incorporado al desarrollo de un proyecto de software concreto. Por este motivo, la reutilización que proponemos se realiza SIN MODIFICAR el proceso clásico, centrado en el proyecto, del desarrollo de sistemas informáticos.

Este ciclo comienza por la fase de Definición de necesidades de conocimiento. Esta fase es la única que NO se efectúa al amparo de un proyecto de software concreto sino que se desarrolla independientemente de ellos, y se centra en enmarcar el resto del proceso de reutilización. Es decir, hay que definir el dominio a cubrir, si se va a efectuar todo el proceso en un mismo dominio o, por el contrario, si es conveniente crear más de un dominio; cuáles son las fronteras de estos dominios. También es importante determinar cuáles son los tipos de activos que formarán parte de estos dominios, teniendo en cuenta que a posteriori se pueden incluir más activos en el sistema.

La palabra clave es precisamente dominio. Todas las actividades de conocimiento se estructuran alrededor de un dominio, que para The Reuse Company no es más que un repositorio donde:

- » Se almacena el conocimiento que se posee sobre el dominio en una estructura ontológica
- » Se almacenan los activos de conocimiento

Durante la fase de Búsqueda de activos reutilizables, ya realizable dentro del desarrollo de un proyecto de software, se partirá de una necesidad clara de conocimiento, así como de la estructura de repositorio y herramientas CASE que soportan la reutilización implantadas en la fase anterior, y se hará una actividad de prospección que dé lugar a localizar activos candidatos a ser reutilizados en el nuevo proyecto SW a desarrollar. Debido a que la visión de reutilización propuesta pretende desarrollar SW mediante el clásico proceso de desarrollo de Software centrado en el proyecto y sus diferentes fases, The Reuse Company considera activos reutilizables todos los diferentes elementos que se necesitan para dicho desarrollo (requisitos, riesgos, pruebas, diagramas de UML, Atributos, métodos, estimaciones, etc.) y por lo tanto, un nuevo proyecto de software desarrollado mediante la metodología propuesta deberá buscar requisitos, riesgos, diagramas UML, etc... reutilizables desde proyectos anteriores.

Obviamente, inmediatamente después de implantar una Unidad de Reutilización como el que propone The Reuse Company, el repositorio puede no poseer almacenado ningún activo de proyectos anteriores (salvo que se haya decidido realizar una indización previa con proyectos anteriores ya realizados), por lo que el primer proyecto desarrollado NO podrá reutilizar nada. Desde el punto de vista de The Reuse Company esta aproximación es la que representa nuestra fortaleza: No se necesita invertir por adelantado para reutilizar, sino realizar el proceso incremental, y a coste cero. Además, según vaya evolucionando de forma incremental nuestro dominio, el repositorio también lo irá haciendo de forma incremental por lo que no serán necesarias enormes inversiones para mantener actualizado el dominio.

Una vez aquí, cuando se desee construir la solución a un nuevo problema, se podrá, en cualquier momento (ver sección Metodología donde se describe IRM) acceder al repositorio para localizar, bien conocimiento ontológico, o bien activos similares creados en el pasado.

La fase de aplicación del conocimiento reutilizable al nuevo proyecto es eminentemente específica de nuestra visión de reutilización. Incorporar activos de proyectos anteriores, o propios del dominio dentro de un nuevo proyecto NO es una tarea inmediata, aunque sí puede ser automatizada. Las herramientas CASE de reutilización han de ser capaces de incorporar este conocimiento al nuevo desarrollo e integrarlo con el resto de activos del mismo.

Teniendo en cuenta todo lo anterior, la fase de Identificar el conocimiento a reutilizar es tremendamente importante para soportar la metodología incremental de reutilización que propone The Reuse Company. A esta fase se llega generalmente cuando el proyecto ha concluido, aunque para proyectos largos pueden existir otras aproximaciones, y supone la etapa en la que todos los activos utilizados en el desarrollo del proyecto, y que son candidatos a ser reutilizados en nuevos proyectos (requisitos, diagramas UML, modelos de datos, esquemas XML, Documentación, etc..) son procesados e incorporados al repositorio. Para esta fase la metodología IRM (Incremental Reuse Method) propone:

- » Sub-Fase de debriefing: durante esta fase se debe aportar cierta información a los artefactos que se utilizará en futuros proyectos y no en el presente proyecto (que ya se considera inicialmente cerrado). Ejemplos de esta información pueden ser los datos de cierre de ciertas métricas estimadas al inicio del proyecto; o bien la determinación del grado de bondad de las estrategias y actividades desplegadas para la mitigación de nuestros riesgos
- » Sub-Fase de indización: durante esta fase se tratan los activos importantes del proyecto para introducirlos al repositorio (dominio). Debido a que las herramientas utilizadas son capaces de localizar conocimiento dentro de bastas cantidades de información, no es necesario decidir qué activos son tratados y cuáles no, y la totalidad de cada proyecto puede formar parte del repositorio

Hay que volver a destacar que, a diferencia de otras aproximaciones a la reutilización de software, donde resulta necesario realizar un complejo y caro proceso de Ingeniería de Dominios, en la metodología propuesta por The Reuse Company no es imprescindible comenzar el proceso de reutilización con todos los activos identificados y bien conocidos por todos los miembros de la organización, sino que estos activos pueden ser incorporados progresivamente, y las herramientas serán capaces de localizarlos en todo momento.

Además de la aproximación basada en la reutilización de activos creados ad-hoc en proyectos anteriores, The Reuse Company propone, como cierre de esta fase, la aplicación de técnicas de

clasificación, clustering y reconocimiento de patrones, para intentar realizar de forma automática el proceso de ingeniería de Dominios puro (Commonality/Variability) pero también de forma incremental. Esto significa que se deben intentar localizar activos comunes, que reiteradamente se repiten en gran parte de los proyectos informáticos de la organización, para realizar un proceso mas profundo de construcción de activos reutilizables (por ejemplo localizar patrones de requisitos con los que se repiten, patrones UML a partir de partes comunes de los diagramas UML, etc.) . Las herramientas informáticas de The Reuse Company se encargan de realizar este proceso de forma automática.

Por último, llega el momento de Revisar el proceso de reutilización. En este momento deben tenerse en cuenta métricas que nos ayuden a medir:

- » Si el proceso de reutilización está suficientemente extendido
- » Cuáles son los tipos de artefactos que más suelen reutilizarse
- » Si la reutilización acorta tiempos de desarrollo e incrementa calidad
- » ...

Esto nos llevará a determinar cómo puede mejorarse el proceso de reutilización.

3.2.4 ¿Qué es una ontología y cómo se crea?

Como ya se ha comentado, una ontología es la aproximación utilizada por The Reuse Company para almacenar conocimiento en un repositorio. Puede verse una ontología como el conjunto de²:

- » Los conceptos claves del dominio tratado, junto con sus sinónimos, siglas...
- » Las relaciones lógicas que puedan definirse entre dichos conceptos: conceptos relacionados, conceptos jerárquicos (padre e hijo)
- » Otro tipo de relaciones propias del dominio: por ejemplo, en el dominio de las finanzas, la relación de comprar
- » Las reglas que rigen el dominio

Una vez que sabemos qué es una ontología, surge la eterna pregunta de ¿cómo lo construyo? ¿Es caro construir una ontología? The Reuse Company pone a su disposición un conjunto de herramientas y una metodología que le permitirán, de forma automatizada, la creación de ontologías en un tiempo record. Utilizando herramientas de procesamiento del lenguaje natural (PLN) sobre sus propios documentos u otros relevantes, The Reuse Company reduce considerablemente el tiempo que sus expertos deberían dedicar a la construcción de la ontología³.

² The Reuse Company tiene a su disposición un conjunto de documentos que describen más en detalle el concepto de ontología. Visite <http://www.reusecompany.com>

³ Más información sobre la metodología de construcción de ontologías en <http://www.reusecompany.com>

3.3 Su estructura

La estructura de la Unidad de Reutilización de Conocimiento no es complicada, y puede adaptarse al tamaño de cualquier organización. En organizaciones pequeñas, es decir, pequeñas PYMES formadas por pocos empleados, todos los roles de esta Unidad pueden ser jugados por una única persona, o incluso realizar un proceso de “outsourcing” de dichos roles a The Reuse Company. En organizaciones de mayor tamaño, podría tenderse incluso a una persona diferente por cada uno de los roles que se describirán a continuación.

3.3.1 Arquitecto de dominios

Como ya se ha comentado, las ontologías forman el núcleo del repositorio (dominio). Por ello, se requiere un rol con capacidad de liderar la construcción y evolución de las ontologías. No necesariamente será un experto en el dominio a construir, sino más bien un experto en la construcción de dominios.

Por otro lado, determinará qué activos serán los utilizados para la construcción de las ontologías.

Este rol es uno de los más susceptibles de ser contratado a The Reuse Company, donde se dispone de formación en las metodologías y herramientas precisas para construir las ontologías.

3.3.2 Experto de dominio

Se encargará, junto con el arquitecto de dominios, de la construcción de la ontología. Será entrevistado por éste y hará uso de las herramientas de trabajo colaborativo necesarias para la construcción y evolución de las ontologías.

En la fase de construcción inicial de la ontología pueden utilizarse varios expertos de dominio, sin embargo, una vez construida la ontología, la dependencia de este rol se minimiza, siendo incluso suficiente con una única persona para la evolución del mismo.

En caso de que se desee abordar un gran proyecto, o bien un proyecto cuyo ámbito de conocimiento esté ligeramente fuera de las fronteras del dominio ya construido, una persona que cumpla con este rol debería verse involucrado en el proyecto. Es decir, abandonaría temporalmente la unidad para dedicarse a ese nuevo proyecto.

3.3.3 Catalogador

Es el rol encargado de incluir activos finalizados en el repositorio. Por lo tanto, se encargará de la fase de indización y de dar soporte al responsable de cada proyecto en la fase de debriefing.

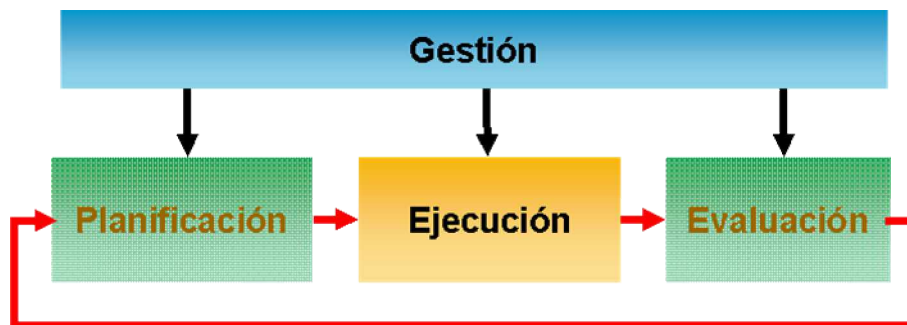
Asimismo, da soporte al resto de usuarios del sistema en la fase de recuperación y reutilización del conocimiento.

3.3.4 Jefe de Unidad – Reuse Manager

Su misión es la de coordinar al resto de roles de la Unidad. Asimismo, es el rol ideal para la comunicación con la dirección y la determinación de los objetivos; es decir, será el encargado de las actividades de gestión y de planificación (ver sección Sus actividades principales). También dentro de la actividad de Evaluación, se encargará de definir y corregir en su caso las métricas con las que se mide el proceso.

3.4 Sus actividades principales

En realidad, las actividades principales del proceso de gestión de la reutilización no distan nada de las de otros procesos, sin embargo hay que personalizarlas:



3.4.1 Gestión

Hay que recordar que son actividades que van a requerir una inversión inicial (aunque pequeña, y basada esencialmente en la adquisición de herramientas informáticas, la creación de un dominio mínimo y algo de formación) pero que afortunadamente tienen un retorno de inversión a corto-medio plazo. Por lo tanto, para su correcto éxito se requiere que la dirección de la organización se involucre, sin ello cualquier planificación está abocada al más rotundo de los fracasos.

Por ello debe definirse la figura del sponsor de la reutilización; debe ser una persona perteneciente al equipo de dirección, involucrado de forma directa en el éxito de la iniciativa.

Por otro lado, se deben determinar unos objetivos estratégicos realistas y en consonancia con los de la organización, deben ser susceptibles de ser medidos y, obviamente, deben medirse para determinar el éxito del proceso, lanzar las correspondientes alarmas en su caso...

También debe preverse la interacción con otros procesos de la organización.

Por último, recordar que tanto el sponsor como el resto de roles, deberán quizá trabajar en base a incentivos (siempre que vaya en consonancia con la organización). Adaptar estos incentivos al éxito de las políticas de reutilización garantizará que éstas lleguen a buen puerto.

3.4.2 Planificación

La planificación es el paso previo a la ejecución. Durante esta fase se deben marcar objetivos concretos y de corto recorrido, debe marcarse cuál debe ser la estructura de la unidad de reutilización de conocimiento, marcar los roles del mismo y marcar cómo han de ser utilizados los fondos proporcionados por la gerencia.

Hasta aquí podríamos ver la fase de planificación de cualquier unidad actual. Sin embargo, también debemos acometer otras tareas tales como:

- » Establecer cuál es la forma de medir el éxito o el fracaso del proyecto de reutilización de conocimiento. Qué variables van a ser tenidas en cuenta, cómo se van a cuantificar, cuándo y por quién
- » Definir concretamente el proceso a llevar a cabo
- » Tener en cuenta aspectos tecnológicos, es decir, cuáles son las herramientas más adecuadas para acometer este reto

3.4.3 Ejecución

Si las dos fases anteriores han tenido éxito, esta es prácticamente la fase más sencilla. Aquí debemos acometer proyectos haciendo uso de la reutilización, pero también debemos llevar a cabo actividades que no pertenecen a ningún proyecto, sino que servirán como marco a todos ellos.

Se necesita cuantificar todas las variables definidas en la fase de planificación. Esta es la única forma de:

- » Tener alerta temprana ante problemas emergentes
- » Poder indicar a la gerencia cuál el grado de éxito del plan
- » Acometer el sistema de incentivos
- » ...

3.4.4 Evaluación

Durante esta fase se evalúan los datos obtenidos en la fase anterior, lo que nos va a permitir, fundamentalmente, mejorar nuestros procesos, por ello, podría considerarse tanto o más importante que las anteriores.

3.5 Relación con otros departamentos

De lo comentado hasta este momento puede deducirse fácilmente cuál va a ser la vinculación de esta unidad con el resto de departamentos de la organización.

El Jefe de la Unidad de Reutilización de Conocimiento debe tener una estrecha colaboración con la Gerencia de la compañía. No en vano los objetivos de esta unidad van en consonancia con los objetivos de la organización. Asimismo, y con el afán de minimizar riesgos, la Gerencia ha de ser informada con bastante frecuencia sobre los hitos cumplidos, retornos de inversión y resto de métricas cultivadas durante los proyectos.

Por supuesto, si la compañía ya posee un Departamento tradicional de Documentación, la relación con este grupo de personas es crucial, incluso debería plantearse la posibilidad de fusionar a ambos grupos.

Por último, y ya que se trata de organizaciones dedicadas al desarrollo de software, la Unidad de Reutilización de Conocimiento tiene una clarísima e íntima relación con dicho Departamento de Desarrollo (incluso algunas organizaciones los desean integrados).

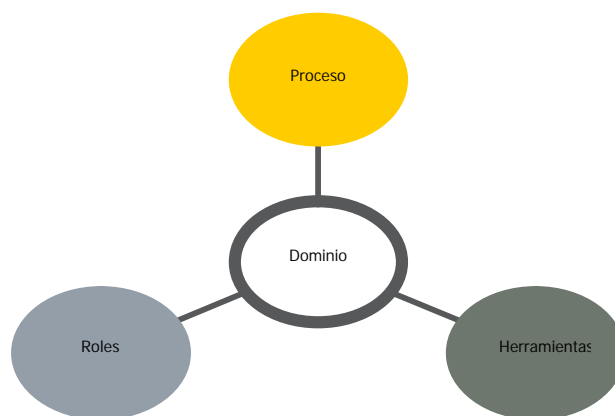
4 HERRAMIENTAS Y METODOLOGÍA

4.1 Metodología de reutilización: IRM

4.1.1 Introducción

La metodología IRM (Incremental Reuse Method) es la única que da un soporte amplio a la reutilización de software que nosotros proponemos. Sus pilares fundamentales son los siguientes:

- » Dominio: como ya se ha comentado, todas las actividades de Gestión y Reutilización de Conocimiento de The Reuse Company se fundamentan en un dominio de conocimiento basado en una ontología
- » Roles: los roles ya comentados son vitales para poner en marcha esta metodología y la Unidad de Reutilización de Conocimiento
- » Mínimas modificaciones al Proceso de Desarrollo de Software clásico: como es lógico, necesitamos modificar el proceso de desarrollo para orientarlo a la reutilización pero manteniendo la orientación hacia el "Proyecto" y su gestión mediante el SDP clásico. La idea es modificarlo de la forma más ligera posible, para conseguir que la curva de aprendizaje se minimice.
- » Herramientas de reutilización: The Reuse Company posee un conjunto de herramientas informáticas que posibilitan todo el proceso. Su descripción mas profunda se detalla en la próxima sección



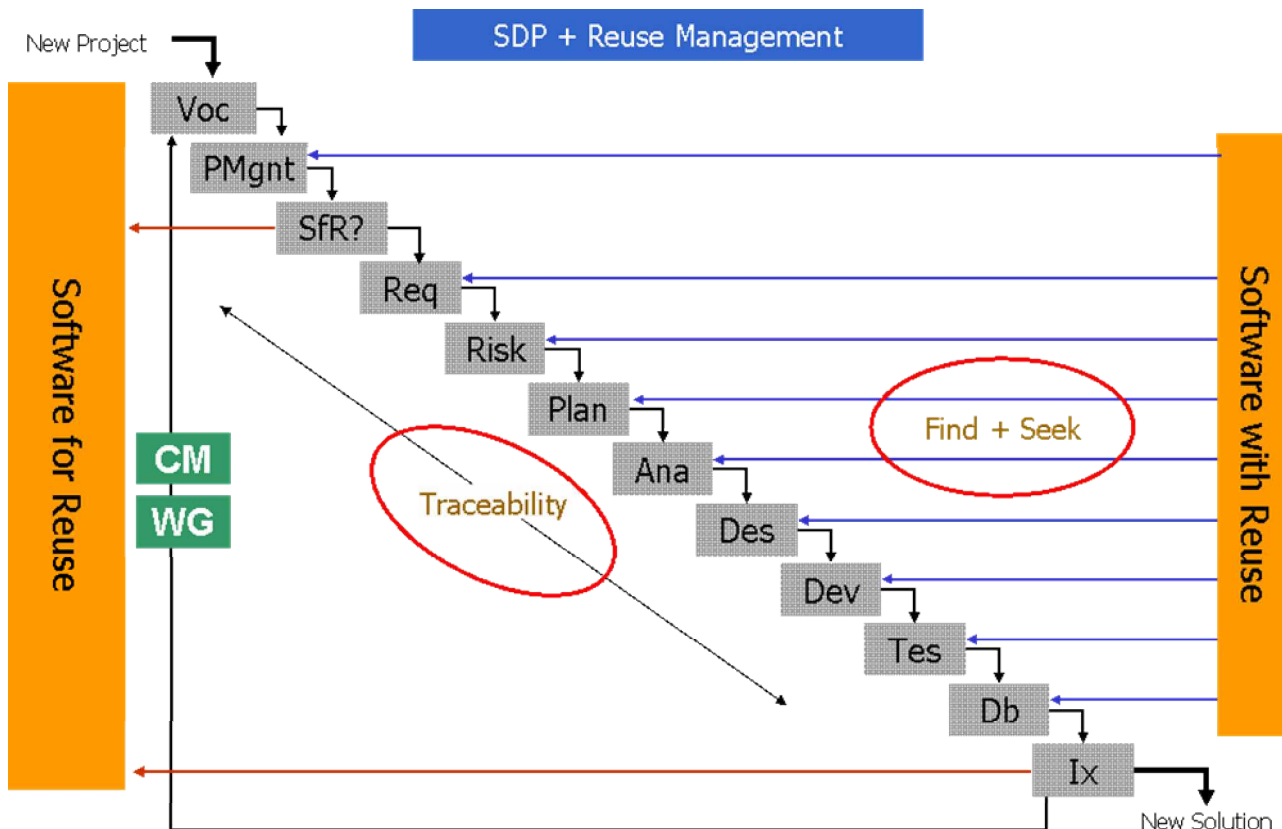
Este trébol, sin cualquiera de sus hojas y, por supuesto, sin su corazón, carecería de toda lógica y pondría en peligro el éxito de las actividades de reutilización. Pasemos a ver la hoja del proceso, la única no vista en el resto del documento.

4.1.2 El proceso

The Reuse Company ofrece Incremental Reuse Method (IRM) como metodología para soportar la reutilización en una organización que desarrolla SW.

La idea de la metodología IRM es la de:

1. Poder adaptarse a cualquier metodología en caso de ya tenerla
2. Dotar de marco metodológico a aquellas organizaciones que aún no lo posean



A pesar de su visión descendente, se trata de un ciclo de vida en espiral, basado en otros bien conocidos como por ejemplo el Proceso Unificado [UNIF]. Correspondencia con las actividades tradicionales:

Acrónimo en diagrama	Nombre de actividad
Voc	Gestión de vocabulario (dominio)
PMgmt	Gestión del proyecto. Esta tarea se mantiene durante todo el proyecto
SfR	Software for Reuse (ver abajo)
Req	Gestión de Requisitos
Risk	Gestión de Riesgos
Est	Estimación
Ana	Análisis
Des	Diseño
Dev	Desarrollo
Tes	Pruebas
DB	Post-mortem o Debriefing (ver abajo)
IX	Indexación

Voc – Construcción del dominio ontológico

Durante esta fase se construirá la estructura ontológica que potenciará todas las futuras actividades de reutilización. Por esta razón, se propone una pequeña inversión inicial previa al desarrollo de ningún proyecto y que consta de:

- » Una ontología clásica: donde se incluyen los conceptos del dominio, sus relaciones y las reglas y axiomas que marcan el dinamismo del dominio
- » Patrones verticales: donde se deberían considerar tanto patrones de análisis, como patrones de requisitos
- » Grandes componentes reutilizables⁴: si se desea abordar la reutilización mediante la vía de construcción de frameworks, es un buen momento para diseñarlos y construirlos

Posteriormente, el resto de proyectos que se inicien basados en este mismo dominio, no tienen más que hacer alguna revisión, si se considera necesario, a dicho dominio ontológico.

SfR – Software for Reuse

Durante esta fase del proyecto de software concreto se determina por el equipo de proyecto, guiados por el Reuse Manager, si puede resultar interesante diseñar específicamente algún componente del proyecto "for Reuse". Esto significaría la realización de unas especificaciones, diseño, codificación, pruebas, documentación, etc diferentes de las que se realizarían si no fuera concebido como reutilizable. Esta fase está pensada para que no se tenga que depender del reconocimiento automático de posibles patrones de reutilización futuros.

⁴ Esta fase es completamente opcional

DB – Post-mortem o Debriefing

La fase de post-mortem o debriefing se lleva a cabo justo tras la finalización tradicional de cada proyecto. Durante esta fase se aportan datos, no quizá importantes para el proyecto en sí, sino importantes para el control del proceso, y para dotar de mayor información a los futuros reutilizadores de este proyecto.

IX - Indexación

Durante esta fase, el Catalogador se encarga de introducir los activos del proyecto dentro del repositorio. A partir de este momento, el proyecto está a disposición de cualquier otro miembro del equipo de desarrollo.

Esta fase suele acometerse al finalizar cada proyecto, sin embargo, en proyectos extensos puede crearse un pequeño repositorio propio del proyecto y la fase de indexación (siempre al repositorio auxiliar del proyecto) podría efectuarse regularmente.

Dispone de más documentación sobre IRM en nuestra página Web <http://www.reusecompany.com>.

4.2 Herramientas de reutilización

Si planifica su reutilización, la gestiona y utiliza metodologías orientadas a la reutilización de software, necesita que sus herramientas estén también orientadas a la reutilización de software.

4.2.1 swREUSER

Esta herramienta representa un entorno CASE completo orientado a la reutilización de software. Es decir, además de dar cobertura a áreas tales como:

- » Gestión de requisitos
- » Planificación
- » Gestión de riesgos
- » Estimación
- » Modelado con UML
- » Patrones de diseño
- » Métricas
- » Generación de código
- » Gestión de pruebas
- » ...

También da soporte a trabajo colaborativo, traza completa entre cualquier elemento del ciclo de vida, generación de documentación,... Sin embargo, su principal novedad radica en sus aproximaciones a reutilización de software. En este sentido:

- » Permite el acceso a la información ontológica
- » Permite el trasvase de conocimiento desde la capa ontológica hasta las capas de modelado de software
- » Permite incluir información de post-mortem sobre los activos de software
- » Permite clasificar los activos construidos dentro del repositorio

- » Permite el acceso y la reutilización de activos que ya estuviesen previamente almacenados en el repositorio

4.2.2 Reutilización de modelos Rational Rose

Si por cualquier motivo decidiese seguir utilizando Rational Rose como herramienta de modelado, swREUSER le permite poder reutilizar también este tipo de modelos.

4.2.3 REUSE Server

Esta herramienta forma el núcleo de la reutilización de software. En su seno residirán las ontologías construidas, así como también constituirá el repositorio donde se ubicarán los diferentes proyectos finalizados o en vías de finalización.

Todo el resto de herramientas orientadas a la reutilización de software trabajarán contra dicho repositorio.

4.2.4 domainREUSER

Esta herramienta le permite la creación manual y mantenimiento de una estructura ontológica que potencie las actividades de reutilización de software. Si desea automatizar este proceso, no dude en ponerse en contacto con The Reuse Company y descubra cómo construir ontologías de calidad, en un tiempo record y por una fracción del coste que siempre soñó.

Entre sus principales características destacan:

- » Gestión de conceptos: sus notas, estado, sus relaciones y sus sinónimos
- » Triple vista sincronizada: jerárquica, alfabética y gráfica
- » Soporte a la poli-jerarquía
- » Diferentes opciones de importación y exportación
- » Acceso web a la ontología

Puede encontrar demos y documentación sobre estos y otros productos en nuestra Web <http://www.reusecompany.com>.

5 QUÉ ES LA 'UNIDAD DE REUTILIZACIÓN DE CONOCIMIENTO POR 15.000€' KU15K

Unidad de Reutilización de Conocimiento por 15.000 € es todo lo comentado hasta este momento. Unidad de Reutilización de Conocimiento por 15.000 € es:

- » Manuales y artículos prácticos: sobre la gestión, planificación, ejecución y evaluación
- » Metodología de trabajo: formación y documentación al respecto
- » Construcción de la Ontología: The Reuse Company construirá una pequeña ontología relativa a tu propia área de negocio. Esta ontología debería crecer, bien con el trabajo directo de la nueva Unidad de Conocimiento de la organización, o bien como una colaboración con The Reuse Company
- » Herramientas: orientadas a la reutilización de cualquier activo de software
- » Soporte: en las herramientas y en la formalización de la Unidad

Y todo esto por tan solo 15.000 €⁵.

⁵ Gastos de soporte no incluidos

6 REFERENCIAS

- [CHAO] <http://www.standishgroup.com>
- [FOWL] "Analysis Patterns: Reusable Object Models". Martin Fowler. Addison-Wesley Professional, 1996
- [GAMM] "Design Patterns: Elements of Reusable Object-Oriented Software". Gamma et al. Addison-Wesley Professional, 1995
- [HURT] "Generación de Patrones de Requisitos en una Herramienta CASE: Aplicación al Common Criteria". Omar Hurtado, Juan Llorens, Gonzalo Génova, José Fuentes. 8° Workshop Iberoamericano de Ingeniería de Requisitos y Ambientes de Software-IDEAS '2005, 2-6 mayo 2005, Universidad Técnica Federico Santa María, Valparaíso, Chile.
- [IEEE] IEEE Std 610.12-1990. IEEE Standard Glossary of Software Engineering Terminology
- [JACO] "Software Reuse: Architecture, Process and Organization for Business Success". Ivar Jacobson. Addison-Wesley Professional, 1997
- [ONTO] <http://www.reusecompany.com/docs/Tecnología%20REUSERS.pdf>
- [PRIE] "A Faceted Approach to Building Ontologies". Rubén Prieto-Díaz
- [UNIF] "The Rational Unified Process: An Introduction". Philippe Kruchten. Addison-Wesley Professional, 2000

dTinf. The Reuse Company
Avda. Rey Juan Carlos I, 98, 4ºD
28916 Leganés, MADRID
SPAIN

| www.reusecompany.com | +34 91 680 90 22 | contact@reusecompany.com |





The Reuse Company, The Way to Reuse Knowledge and The Way to Reuse Software are trademarks of DTINF S.L. Copyright 2006.