
Gestión de documentos en entornos de desarrollo de software: normativa y aplicabilidad de soluciones *open source*

Document management in software development activities: standards and feasibility of open source solutions

Ricardo EITO-BRUN

Universidad Carlos III de Madrid, c/ Madrid, 126, 28903 Getafe (Madrid), reito@bib.uc3m.es

Resumen

El desarrollo de software es una de las actividades donde la eficiente gestión de la información no sólo se traduce en ventajas para suministradores y clientes. En muchos casos, los métodos y las normas aplicables para el diseño y desarrollo de estas soluciones técnicas constituyen requisitos para los proveedores, que deben aplicar pautas para la identificación y gestión de los documentos. A esto se suma la necesidad de gestionar la evolución de los documentos, y de distintos items que —aunque incluidos normalmente en documentos de naturaleza textual— exigen una gestión independiente, como sucede con requisitos de distinta naturaleza, pruebas del software, elementos de diseño o el propio hardware en el que se desarrollará, verificará y ejecutará el programa. Todos estos elementos —junto con otros— constituyen de forma directa o indirecta la configuración del software, entendiendo como tal el conjunto de programas, datos, documentos y la infraestructura técnica que permiten su desarrollo y ejecución. La gestión eficiente de estos elementos constituye uno de los principales retos para los profesionales de la información que desempeñan su actividad en estas organizaciones. Una de las dificultades tiene como origen el hecho de que los actuales sistemas de gestión de documentos deben ser adaptados para satisfacer las necesidades que impone la normativa aplicable en estos entornos. En el artículo se presentan las dificultades más frecuentes en la gestión de los diferentes elementos informativos y documentales que constituyen la configuración de una solución informática y la adaptación desarrollada a partir de un sistema de gestión documental y de contenidos *open source* para gestionar la configuración de proyectos de desarrollo de software.

Palabras clave: Gestión documental. Gestión de contenidos. Código abierto. Desarrollo de software. Gestión de la configuración del software.

1. Introducción

Los proyectos de desarrollo de software exigen la elaboración y gestión de un importante número de documentos. Los documentos o entrega-

Abstract

Software development is one activity that requires an efficient management of information. In this context, following well-defined information management practices do not only provide advantages for providers and customers; in most of the cases, there is a need to be compliant with different standards that enforce organizations to follow specific rules and practices for the identification and management of documents. In addition, organizations involved in software development activities must manage the complex evolution of documents and other items that, although usually endorsed in textual documents, require a separate, specific management. This happens with software requirements, design elements or the hardware platform on which the software will be executed. All these items, among others, constitute the configuration of the software. This term refers to the set of programs, data, documents and technical infrastructure that permits its development and execution. Efficient management of these elements is one of the main challenges for information professionals working in these scenarios, and one of the difficulties is probably due to the need to adapt currently available EDMS (Electronic Document Management Systems) to these situations and to the applicable standards. The author presents the main issues regarding the management of the different information items and documents in software configuration and the customisation completed in an open source document and content management system to manage configuration items in software development projects.

Keywords: Document management. Open source. Software development. Configuration management.

bles reflejan las especificaciones del sistema, requisitos, diseño, casos de prueba asociados, etc., así como los planes que se utilizarán para comprobar que el trabajo se completa según lo previsto. Una gestión efectiva de esta documen-

tación es imprescindible para asegurar la integridad de la información que comparten los equipos de trabajo. Por otra parte, el dinamismo característico de estos desarrollos exige normalmente la actualización regular de los documentos, con lo que se hace evidente la necesidad de disponer de mecanismos que garanticen el acceso las versiones aprobadas de los documentos.

2. El desarrollo de aplicaciones informáticas como un proceso de elaboración de conocimiento

El punto de partida de un desarrollo de una aplicación informática es la resolución de un problema o el desarrollo de mejoras en el funcionamiento de una organización o en los servicios que presta a sus usuarios.

La identificación y descripción del problema u oportunidad debe hacerse de forma precisa, para establecer el alcance del trabajo e identificando con precisión en qué consiste la solución técnica que se quiere implementar. Esto debe quedar plasmado en unos documentos, normalmente escritos en lenguaje natural, que sirvan como referencia al equipo de trabajo en las actividades sucesivas: elaboración de requisitos, descripción de la arquitectura técnica, estimación de costes, planificación y diseño de pruebas, etc.

La visión inicial de la solución a implementar se dividirá en una serie de requisitos, definidos como “una condición o capacidad necesaria para un usuario para resolver un problema o lograr un objetivo, (2) una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, norma, especificación u cualquier otro documento para el que se exija cumplimiento (3), una representación documentada de una condición o capacidad tal y como se define según (1) o (2)” (IEEE).

Tomando como punto de partida esta especificación de requisitos, el equipo de trabajo procederá a completar el diseño del sistema. En una primera etapa, se establece el diseño de alto nivel —o arquitectura—, que será elaborado posteriormente con un mayor nivel de precisión —diseño detallado. La descripción o documento de diseño recogerá esta información, normalmente formulada en un lenguaje formal, y que incluirá los distintos componentes del sistema, su lógica de control, formatos de entrada y salida de datos, modelos de datos y las descripciones de los algoritmos.

La elaboración del diseño a partir de las especificaciones implica un proceso donde el conoci-

miento del problema expresado inicialmente en forma de requisitos se reformula usando lenguajes especializados que facilitarán la implementación en un lenguaje de programación determinado.

De esta forma, en cada etapa del ciclo de desarrollo se producen elaboraciones más detalladas del conocimiento capturado inicialmente sobre un problema o situación real o deseable. Esta representación se irá precisando hasta permitir su implementación en un sistema informático. Un aspecto importante en esta actividad de representación del conocimiento es asegurar que no se producen pérdidas entre las distintas representaciones. Con esto queremos decir que la representación inicial de un sistema en un conjunto de requisitos debe ser completa y corresponder con la realidad y la problemática a la que se quiere dar respuesta. En el siguiente paso (elaboración del diseño), se debe garantizar que todos los elementos identificados en éste son suficientes para satisfacer los requisitos anteriores. En la codificación, se debe asegurar que la implementación que se hace en uno u otro lenguaje cubre la totalidad del diseño que se ha aprobado. Para garantizar que estos niveles de representación son coherentes y que no se pierde información al pasar de uno a otro, se recurre a distintas actividades y técnicas, como la trazabilidad y las inspecciones.

La trazabilidad indica en qué medida una relación puede establecerse entre dos o más productos generados en el proceso de desarrollo, especialmente aquellos que tengan entre sí una relación de “predecesor-sucesor” en el orden de actividades que deben completarse. También se define como el grado en el que un producto del desarrollo de software establece su razón de existencia, o dicho de otra forma, en qué medida está justificada su realización y son evidentes y justificados los motivos que han dado origen a su creación. La trazabilidad también permite evitar la sobre-especificación del software, un problema frecuente que se produce cuando se desarrollan o implementan componentes, módulos o funciones adicionales a los exigidos. La trazabilidad se plasma en un tipo de entregable adicional llamado “matrices de traza”, en los que se establece la relación entre los distintos tipos de ítems (requisitos, elementos de diseño, archivos de código fuente, pruebas, etc.) y que se adjuntan a los documentos producidos durante el proceso de desarrollo.

3. Gestión de documentos y registros

Los registros sirven como evidencia de que se han completado ciertas actividades durante el

ciclo de desarrollo. Según la recomendación técnica ISO 90003:2004, apartado 4.2.4, “los registros deben establecerse y mantenerse para proporcionar evidencia de la conformidad de los requisitos [...], deben permanecer legibles, fácilmente localizables y recuperables”. También se establece la necesidad de disponer de “un procedimiento documentado para definir los controles necesarios para la identificación, almacenamiento, protección, recuperación, tiempo de retención y disposición de estos registros”.

Algunos registros habituales en los proyectos informáticos son los resultados de pruebas, informes de problemas, informes de evaluación y auditorías, registros de inspecciones y revisiones completadas sobre documentos y código, peticiones de cambios en requisitos, etc. Otro tipo de registros mencionados en la ISO 90003:2004 son los cambios en recursos, estimaciones de coste y esfuerzo, justificaciones de la selección de herramientas, tecnologías o suministradores, actas de reunión, registros de versiones de software y licencias para el uso del software. No debemos olvidar los resultados de actividades de verificación completados automáticamente, como los de estándares de diseño o codificación.

Desde el punto de vista del tratamiento documental, la principal característica de los requisitos es que éstos no se pueden modificar —ni intencionada ni accidentalmente— y que deben conservarse durante el tiempo que exija el contrato entre el suministrador y el cliente.

Respecto a las prácticas de gestión de documentos, el apartado 4.2.3 de ISO/IEC 90003:2004 establece unas prácticas mínimas, entre ellas “aprobar los documentos en cuanto a su adecuación antes de su emisión, asegurar que se identifican los cambios y el estado de revisión de los documentos, que las versiones pertinentes se encuentran disponibles en los puntos de uso, y que los documentos permanecen legibles y fácilmente identificables”. El control de los documentos de origen externo y el control de su distribución forman parte de las prácticas fijadas en esta recomendación.

Todo esto nos conduce a la necesidad de disponer de mecanismos que garanticen la correcta distribución de versiones autorizadas de los documentos y el estricto control de los registros (ECSS-M-50B, ISO/IECTR 9294).

4. Documentos, datos e ítems

Muchos de los documentos que se generan en el desarrollo de un sistema informático tienen un carácter modular. Estos documentos recogen

datos correspondientes a elementos o ítems gestionables de forma independiente. Por ejemplo, una especificación de requisitos recoge datos independientes de requisitos individuales, cada uno de ellos con sus propias características: descripción, título, versión, estado, responsable, etc.

Lo mismo sucede con las especificaciones de pruebas y los documentos de diseño.

Esto hace que muchos documentos se generen —normalmente usando métodos automatizados— a partir de bases de datos estructuradas o modelos creados con aplicaciones especializadas donde se mantienen y gestionan los ítems. El documento es una forma de informar del estado de todos ellos y de facilitar la distribución de esta información.

Desde la perspectiva de la gestión documental, precisamos disponer de un mecanismo capaz de generar documentos susceptibles de ser impresos a partir de información estructurada, pero también se nos obliga a considerar la forma más eficiente de gestionar y almacenar la información estructurada de estos ítems.

5. Relaciones entre ítems

La trazabilidad a la que antes nos hemos referido establece relaciones entre los distintos ítems y documentos. A parte de las relaciones inherentes a los distintos niveles de representación del conocimiento a la que antes nos referíamos, entre documentos y registros se establecen otras relaciones. Por ejemplo, una solicitud de cambio debe relacionarse con aquel requisito que pretende modificar; a su vez, como resultado del análisis de impacto de dicho cambio, se identificarán cambios en elementos de diseño o en archivos de código fuente que tendrán que actualizarse si el cambio se aprueba. Otra relación se da entre los documentos y los registros de revisión generados antes de su publicación, o entre los archivos de código fuente y los registros de inspección. Los informes de problemas también deben relacionarse con la versión de la aplicación en la que se identificaron, y en la que se resolvieron, y así un largo etcétera.

Mantener actualizadas estas relaciones entre ítems es crítico para la integridad y la validez de la información. Por ejemplo, si un requisito se ve afectado por una o más peticiones de cambio, debe ser posible saber en cualquier momento qué peticiones de cambio existen, su estado (aprobadas, rechazadas, pendientes) e impacto. Si no se ofrece una visibilidad de estas relacio-

nes, el riesgo de trabajar con información obsoleta es evidente.

Una forma de presentar la organización de los documentos correspondientes a una aplicación informática o sistema es mediante su organización en torno a un “árbol de producto”. En esta aproximación, el producto o sistema se descompone en las partes que lo constituyen. Para cada una de estas partes, se identifican aquellos documentos que contienen información relevante (especificaciones, diseño, pruebas, informes de problemas, etc.) Existe así una relación adicional entre las partes de la aplicación y los documentos que se refieren a ellas. Cada evolución de las partes podrá estar relacionada con versiones diferentes de los mismos documentos, o incluso con conjuntos de documentos diferentes. Se trata de otra relación que debemos poder gestionar, y que nos conduce al concepto de “línea de referencia”.

6. Líneas de referencia

El glosario IEEE la define como:

(1) una especificación o producto que ha sido revisado formalmente y para el cual se ha alcanzado un acuerdo, y que en consecuencia sirve como base para futuros desarrollos, y que puede ser cambiado sólo a través de procedimientos formales de control de cambios; (2) un documento o conjunto de documentos formalmente designados y fijados en un momento determinado del ciclo de vida de un elemento de configuración.

Podemos ver una línea de referencia como una instantánea del conjunto de documentos y datos tomada en un momento dado, normalmente, tras la revisión formal de una entrega (final o intermedia). Los documentos que conforman esa instantánea constituirán un punto de referencia para la futura evolución del software.

En cualquier momento, la organización o el equipo de proyecto debe poder informar del estado de los elementos de configuración que constituyen el sistema y de su documentación asociada. La elaboración de estos informes resulta extremadamente laboriosa si no se dispone de un sistema automatizado.

Un caso particular de los informes de configuración son aquellos que no se corresponden con informes predefinidos en algún tipo de estándar—sino que se solicitan *ad hoc*—, pero que precisan extraer información de los documentos o de los distintos ítems que conforman las especificaciones o el diseño de la aplicación informática.

7. Adecuación de los sistemas de gestión documental al operativa del desarrollo software

En líneas generales, los sistemas de gestión documental ofrecen una serie de funciones directamente aplicables a un entorno orientado al desarrollo de software. Aspectos como el control de versiones de la documentación, evitar la distribución de copias obsoletas o no autorizadas, o la gestión de estados pueden aplicarse directamente para garantizar la integridad de la información. Las funciones habituales para la recuperación de información textual también son ventajosas en este contexto.

Pero el uso de un sistema de gestión de documentos en este contexto exige un esfuerzo de adaptación, ya que las funcionalidades de estos sistemas no cubren todas las necesidades antes descritas.

8. Necesidades de adaptación de un SGED

Estas pueden resumirse en:

- a) Gestión de las relaciones entre los documentos y las partes del sistema identificadas en un árbol de producto.
- b) Gestión de las versiones del sistema (como conjunto), y de cada una de sus partes.
- c) Gestión de tipos de relaciones entre documentos y registros.
- d) Capacidad de establecer líneas de referencia.
- e) Registro y mantenimiento de información para ítems tipo requisitos, casos de prueba o elementos de diseño.
- f) Capacidad de mantener y registrar la trazabilidad entre elementos, y generar informes a partir de esta información.
- g) Capacidad de generar documentos e informes a partir de los ítems modulares registrados en el sistema.
- h) Capacidad de obtener (o importar) información de ítems creados con otras aplicaciones de modelado, como los sistemas de gestión de requisitos o las aplicaciones para el diseño orientado a objetos.

Si se comparan los puntos anteriores con las funciones características de los sistemas de gestión documental disponibles, encontramos importantes diferencias. Para resolver estas diferencias, el autor ha desarrollado una adaptación del software libre Alfresco, que ofrece módulos para la gestión documental, de conte-

nidos y de documentos de archivo. La flexibilidad de este software ha permitido resolver satisfactoriamente la problemática característica de la gestión de información en proyectos de desarrollo software. En los siguientes apartados se describe la aproximación seguida en la aproximación propuesta.

8.1. Gestión del árbol de producto

El árbol de producto consiste en una estructura relacional donde se dan de alta las partes o componentes del sistema. Es por lo tanto un modelo de registro formado por datos estructurados, y que se caracteriza por la necesidad de relacionar entre sí dichos registros. Además, debemos poder relacionar y vincular cada una de sus partes con distintos documentos y registros.

El árbol de producto no puede gestionarse como un documento, ya que su principal característica es la modularidad e independencia de cada una de sus partes. Para resolver esto, se ha mantenido la información de las partes en una lista controlada, con un conjunto de valores posibles, y se ha incluido un campo en el perfil de los documentos para relacionar el documento con las partes del árbol de producto. El uso de códigos mnemotécnicos —representativos de la jerarquía del árbol de producto— garantiza la coherencia de la información y facilita la generación de informes.

8.2. Gestión de las versiones del sistema y de las partes que lo integran

El sistema queda representado por el ítem raíz del árbol de producto, que se desglosaría en distintas partes. El modelo de registro utilizado para gestionar la información de estas partes es común, siendo uno de sus atributos la versión. La relación entre cada documento y las partes del árbol de producto es en realidad la relación entre `<documento>+<version>` y `<componente del árbol de producto>+<versión>`.

8.3. Gestión de información estructurada y no estructurada

Muchos sistemas de gestión de documentos se orientan exclusivamente a la gestión de información no estructurada. Aunque todos ellos incorporan la posibilidad de definir un perfil (conjunto de propiedades descriptivas e identificativas de un documento), los sistemas no suelen permitir crear registros estructurados. Esta función está más próxima a la de los sistemas de gestión de contenidos (una de cuyas características es la posibilidad de definir plantillas formadas por varios campos de longitud variable que

se toman como punto de partida para crear páginas). No obstante, las plantillas características de los sistemas de gestión de contenidos están formadas normalmente por campos de contenido textual, y no encontramos la capacidad de trabajar con otros tipos de datos (fecha, numérico, autonumérico, etc.)

En el caso de la aplicación seleccionada, su módulo de gestión de contenidos permite crear plantillas a partir de esquemas XML, lo que ha nos ha permitido resolver satisfactoriamente esta situación. Se han creado plantillas para la gestión de requisitos, problemas, elementos de diseño (clases y componentes), casos y procedimientos de prueba.

8.4. Líneas de referencia

El concepto de línea de referencia también resulta difícil de implementar en muchos sistemas. Una posibilidad es añadir un campo al perfil del documento que identifique la línea de referencia a la que pertenece cada versión. Pero esta solución no resulta óptima en cuanto una misma versión puede formar parte de distintas líneas de referencia. Pensamos que la mejor aproximación es la generación de un informe que se genera automáticamente y que mantiene la relación entre las líneas de referencia (identificada por un código, fecha de creación, motivo, estado de aprobación) y las versiones de los documentos que forman parte de ella.

La implementación de la línea de referencia considera un caso especial: el de los documentos que recogen datos sobre ítems tipo requisitos, casos de prueba o elementos de diseño.

En este caso, la aproximación adoptada consiste en generar en primer lugar el documento que recoge la información actualizada de dichos ítems, y en la línea de referencia hacer referencia a ese documento en lugar de hacerlo a ítems individuales. Ante la posibilidad de que se generen líneas de referencia erróneas, se ha incorporado la posibilidad de revisar, anular y regenerar líneas de referencia.

8.5. Gestión de relaciones entre tipos de documentos y registros

En este caso, por una parte debemos poder indicar qué relaciones son aceptables desde un punto de vista conceptual y tipificarlas, asignándoles un identificador y nombre descriptivo. Alfresco permite definir distintos tipos de relaciones entre tipos de documentos, por lo que la configuración de esta función ha resultado directa. El sistema también permite —desde el punto de vista del usuario final— establecer relaciones

con facilidad, mostrarlas y activarlas mediante hiperenlaces. Se asegura la integridad de las relaciones, permitiendo únicamente aquellas que estén declaradas como válidas por el administrador.

8.6. Trazabilidad entre ítems y capacidad de generar informes

La trazabilidad es un caso particular de las relaciones. Cada tipo de traza (entre requisitos y elementos de diseño, requisitos y casos de prueba, etc.) debe establecerse como un tipo de relación. La trazabilidad incorpora la generación de informes a partir de estas relaciones y la bidireccionalidad (una vez creada la relación entre los ítems A y B, debemos poder ir tanto de A a B como de B a A).

Esta capacidad se ha implementado mediante el Application Programming Interface (API) facilitado por el programa, que permite leer y procesar los registros almacenados en el sistema.

8.7. Capacidad de importar datos

Finalmente, el sistema se ha adaptado para volcar datos sobre modelos de diseño y requisitos generados con otras aplicaciones especializadas. Concretamente, se ha implementado una interfaz con el formato XMI, ampliamente utilizado para el intercambio de diagramas y modelos basados en el estándar UML.

9. Conclusiones

La adaptación del software de Alfresco ha permitido resolver satisfactoriamente la problemática habitual en un entorno de gestión de documentos en el contexto del desarrollo de software y sistemas informáticos. Las adaptaciones realizadas exigen una combinación de capacidades de gestión documental, de documentos de archivo y de gestión de contenidos para satisfacer los requisitos de estas situaciones.

Referencias

- ECSS-M-50B (2007). Space Project Management, Information and Documentation Management. Noordwijk: ECSS Secretariat
- IEEE Std 610.12-1990 (2002). IEEE Standard Glossary of Software Engineering Terminology. Los Alamitos, CA: IEEE.
- ISO/IECTR 9294 (2005). Information technology. Guidelines for the management of software documentation. 2ª ed. Geneve: ISO/IEC
- UNE-ISO/IEC 90003 (2005). Guía de aplicación de la ISO 9001:2000 al software. Madrid: AENOR.