

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

[4. Captura de requisitos](#)

[4.1 El responsable del usuario](#)

[4.2 El responsable de TI](#)

[4.3 El responsable financiero](#)

[5. Métodos de captura](#)

[5.1 Extracción de requisitos desde la información proporcionada por el cliente](#)

[5.2 Entrevista con el Cliente](#)

[5.3 Observación del usuario](#)

[5.4 Observación de sistemas semejantes](#)

[5.5 Consulta a los expertos](#)

[5.6 Obtención de métricas representativas](#)

[5.7 Uso de prototipos](#)

[5.8 Requisitos autoimpuestos](#)

[6. Árboles de decisión y plantillas de requisitos](#)

[7. Métodos de documentación](#)

[7.1 Lista de requisitos](#)

[7.2 Priorización y categorización de requisitos](#)

[7.3 Matriz de trazabilidad](#)

[7.4 Control de versiones](#)

[7.5 Documento final](#)

[7.6 Validación de requisitos](#)

[7.7 Aplicaciones de gestión de requisitos](#)

[7.8 Un ejemplo: REM](#)

[8. Agrupación de requisitos](#)

[9. Casos de uso](#)

[10. Conclusión](#)

Resumen

Cuando vamos a abordar el desarrollo de un proyecto, ya sea de ingeniería de software o de cualquier campo, una de sus primeras etapas es la definición de los requisitos del proyecto.

Para ello debemos saber qué son los requisitos, cómo se capturan, cómo se analizan y cómo se realiza la definición final de requisitos del proyecto.

En este artículo vamos a realizar una introducción a la gestión de requisitos para el desarrollo de un sistema informático. Vamos a comenzar definiendo lo que son los requisitos: Luego veremos diferentes técnicas de captura de requisitos. También dedicaremos un apartado a la organización de los requisitos. Otro apartado que vamos a incluir es el dedicado a los Casos de Uso, ampliamente usados como complemento a la toma de requisitos del sistema.

1. Introducción a la Gestión de Requisitos

Los requisitos de un sistema son los aspectos que el sistema desarrollado debe cumplir.

Surgen de las necesidades del cliente, de las limitaciones del entorno donde se va a implantar o de la propia gestión de la información que debe realizar el sistema.

Normalmente van a representar valores que debe cumplir como mínimo o como máximo a cada uno de los aspectos desarrollados del sistema. Los requisitos sirven para acotar la funcionalidad o la construcción del sistema suponiendo límites al diseño del sistema y enumerando todas las funcionalidades que debe cubrir el sistema.

2. Los primeros requisitos: objetivo y alcance

Todo proyecto tiene dos requisitos principales, que son el objetivo y el alcance. El objetivo define lo que pretende realizar el proyecto. Si se trata de un desarrollo de software/hardware, define lo que va a hacer el producto final. Normalmente este punto suele ser el primero de todo plan de proyecto, ya que es lo que queremos vender a nuestro cliente.

El alcance es el segundo gran requisito de todo proyecto, ya que pone límites a lo que vamos

2010-04-22
[AppWidget Android: Ejemplo usando BroadcastReceiver y Localización](#)

2010-04-20
[Facelets en JSF 2: sistema de plantillas y componentes por composición.](#)

2010-04-19
[DbVisualizer free version.](#)

2010-04-09
[Session Timeout en RichFaces, con el soporte de Jboss Seam.](#)

2010-04-08
[Jetspeed-2 de Apache Software Foundation](#)

2010-04-07
[Primeros pasos con Balsamiq Mockups](#)

2010-03-18
[Revisando los ejemplos de Cocos2d para iPhone.](#)

2010-03-16
[Organización de eventos con StageHQ](#)

2010-03-15
[Retrasar la carga de Javascript con jQuery.getScript\(\).](#)

2010-03-15
[Optimización de páginas web con Page Speed.](#)

2010-03-09
[JSF 2 ya está aquí !!! The JSF Return, ahora más sencillo que nunca !!!](#)

2010-03-08
[Instalación de tus programas en tu iPhone.](#)

2010-03-04
[Sacar Release de un proyecto con Maven](#)

2010-03-03
[Instalación de Subversion y Apache en Ubuntu](#)

2010-03-03
[Cómo instalar la JDK de SUN en Fedora Linux](#)

2010-03-02
[Creando un botón de compra de Paypal con datos cifrados](#)

2010-03-01
[Creación de un plugin de tipo hook en Liferay](#)

2010-03-01
[ScrumCards de Autentia en Android](#)

a desarrollar. Nos dice hasta dónde va a llegar el sistema que vamos a desarrollar para nuestro cliente, y cuáles son las necesidades generales que va a cubrir nuestro sistema. Normalmente el alcance se define dentro del plan de proyecto, tras el objetivo. Una vez que en nuestro poyecto hemos definido las líneas maestras del mismo, como los estudios de viabilidad o la planificación general, puede comenzar la primera parte del desarrollo del poyecto: la toma de requisitos.

3. Tipos de requisitos

Un requisito es una necesidad del cliente que debe ser cubierta por el sistema que le vamos a ofrecer. Los requisitos van a delimitar cómo quiere el cliente que se comporte el sistema, que información tiene que manejar y cómo la debe procesar y presentar. En este artículo vamos a repasar algunos de los tipos más importantes de requisitos, organizándolos por tipos sin entrar en un nivel de detalle muy alto para cada uno de ellos.

3.1 Requisitos funcionales

En este apartado vamos a recoger los requisitos que afectan directamente a la funcionalidad principal del sistema. Normalmente esta funcionalidad describe los procesos de negocio a los que se destina el sistema.

Requisitos de Actores

Son todos los requisitos que afectan a los diferentes actores del sistema, es decir aquellas personas o sistemas externos que pueden interactuar con el sistema, que van a proporcionarle la información de entrada y van a recibir la información de salida del sistema. En este apartado recogeremos requisitos por ejemplo de :

- Usuarios. Son los responsables de interactuar con el sistema. Pueden ser usuarios físicos o sistemas externos.
- Grupos. Permite realizar conjuntos de usuarios con características comunes, para simplificar las reglas de interacción entre los usuarios y el sistema.
- Perfiles. Permite agrupar todas las características que distinguen a un usuario o grupo.
- Papeles (roles),. Permite asignar grupos de funcionalidades a los usuarios y grupos, permitiendo diferenciar el comportamiento de los usuarios con el sistema según la actividad o proceso a realizar en el mismo.

Requisitos de Interfaz

En este apartado aparecen reflejados todos los requisitos que definen la forma de enviar la información a procesar por los usuarios al sistema, y la forma de recibir la respuesta del sistema por el usuario. Entre ellos podemos distinguir

- **Medios de interacción** (Aplicación de escritorio, páginas Web, Sistemas de Kiosko, etc)
- **Pantallas**, formularios y demás elementos de la interfaz de usuario
- **Mensajes** intercambiados y protocolos de comunicación, fundamentales para describir las interacciones entre sistemas.
- **Movimiento** por la interfaz de usuario del sistema
- **CLAMB**. Pantallas de gestión normalizadas: consultas, listados, altas, modificaciones y bajas.
- **Ayudas** a los usuarios. Debemos recoger el nivel de ayuda y formato de dicha ayuda que proporcionará el sistema a los diferentes tipos de usuarios.
- **Informes**, documentos, archivos y datos en general generados por el sistema.
- **Internacionalización**. Aquí hay que distinguir entre la internacionalización de la interfaz y la internacionalización de la información contenida en el sistema.
- **Accesibilidad**. Define los requisitos de adaptación de un sistema para su uso por usuario con diferentes tipos y grados de discapacidad.
- **Libros de estilo** y cosmética del sistema. Definen los diferentes estándares, estilos, formatos de pantallas, formatos de la información, formatos de salida de los documentos que serán tomados como guía para el desarrollo del sistema.

2010-02-25
[Creando la baraja de SCRUM de Autentia como aplicación para Android](#)

2010-02-25
[Instalar CentOS en Virtualbox con NetInstall](#)

2010-02-22
[Expresiones CRON](#)

2010-02-19
[Cómo utilizar el DataStore de Google App Engine con JDO](#)

2010-02-19
[Recursos Freeware](#)

2010-02-17
[Plugin de mejora de graficos para JMeter](#)

2010-02-17
[Cómo utilizar el datastore de Google App Engine con su API de nivel inferior](#)

2010-02-16
[Aprendiendo Objective-C desarrollando para nuestro Iphone 3Gs](#)

2010-02-11
[Introducción a JCL.](#)

2010-02-09
[Creando la Baraja de SCRUM de Autentia como aplicación para el iPhone 3G.](#)

2010-02-08
[Cómo generar versiones imprimibles de páginas web](#)

2010-02-04
[Como cambiar el tamaño de las fuentes en Xcode \(el entorno de desarrollo para Mac e iPhone\)](#)

2010-02-04
[Primeros pasos con Enterprise Architect y UML 2.x](#)

2010-02-04
[Creación de un componente JSF, basádonos en un plugin de jQuery, con el soporte de RichFaces.](#)

2009-02-03
[Sincronizando el Mail de Mac con Gmail, el correo de Google](#)

2010-02-03
[Integración de jQuery en RichFaces.](#)

2010-02-02
[AjaxSingle: el partialSubmit de RichFaces.](#)

2010-02-01

Requisitos de Procesamiento

En este apartado incluiremos todos los requisitos que permiten realizar las principales funciones del sistema. Son aquellos requisitos que indican qué hacer con los datos de entrada, cómo procesarlos y generar datos de salida. Indican los requisitos que hay que aplicar a las funciones y procesos internos. Hay que definir qué datos hay que tratar y mediante qué procesos se van a tratar. Normalmente para una aplicación de gestión se recogen los requisitos que definen la lógica de negocio.

Requisitos de Persistencia

En este apartado se recogen los requisitos que afectan a la información que se debe persistir en el sistema, es decir la información que se debe guardar entre diferentes ejecuciones del sistema. Normalmente tendremos los requisitos que nos permitirán construir el modelo de datos del sistema. Normalmente para una aplicación de gestión se recogen los requisitos que van a definir las entidades, o información de negocio que se debe persistir.

Requisitos de Gestión y Administración

Estos requisitos recogen todas las funciones que son necesarias para gestionar s el sistema, por ejemplo la gestión de usuarios, gestión de la configuración del sistema y otras funciones del sistema que se apartan de la función principal del sistema.

3,2 Requisitos no funcionales

Aquí se recogen todos los requisitos del sistema que no representan la funcionalidad principal del sistema, sino que fijan condiciones para realizar dicha funcionalidad.

Requisitos de Disponibilidad

En este apartado se incluyen los requisitos que definen la disponibilidad del sistema, el tiempo que debe estar operativo, así como el comportamiento del sistema en caso de fallos. Entre ellas podemos enumerar:

- § Tiempo total de disponibilidad (en %anual)
- § Tiempo medio entre fallos
- § Tolerancia a fallos en el sistema o en su acceso
- § Tolerancia a fallos de su base de datos, si la tiene
- § Tolerancia a fallos del sistema de ficheros, si lo tiene
- § Tolerancia a fallos de otros sistemas o comunicación con sistemas externos
- § Operativas que deben estar disponibles en caso de fallos de alguna de las partes del sistema

Requisitos de Rendimiento

En este apartado se recogen todos los requisitos de rendimiento del sistema, lo que se denominan métricas de rendimiento, pues suelen ser fácilmente medibles. Entre ellos podemos sugerir los siguientes:

- § Velocidad de las peticiones al sistema (número de peticiones que debe responder en cierto tiempo)
- § Tiempo medio de respuesta por tipo de petición, que sería el tiempo máximo (en media) que debería tardar el sistema en contestar a una petición
- § Velocidad en la comunicación con el sistema
- § Velocidad en la gestión de la interfaz del usuario

Requisitos de Calidad

Aquí recogemos los requisitos que afectan a la gestión de la Calidad en el desarrollo del proyecto. Entre ellos podemos destacar:

- § Normativas y procedimientos de gestión del proyecto
- § Normativas y procedimientos de desarrollo
- § Normativas y procedimientos de documentación

[Introducción a RichFaces.](#)

2010-01-29
[Transformación de mensajes en SOA con OpenESB](#)

2010-01-26
[JMeter. Uso de funciones.](#)

2010-01-18
[Autenticando los usuarios de Sonar contra un LDAP](#)

2010-01-18
[Introducción a jQuery UI.](#)

2010-01-18
[jQuery: cómo crear nuestros propios plugins.](#)

2010-01-18
[Cómo consumir un servicio web RESTful con el soporte de Ajax y JSON de jQuery.](#)

2010-01-18
[Introducción a jQuery.](#)

2010-01-17
[Introducción a Tapestry 5](#)

2010-01-14
[JMeter. Gestión de usuarios](#)

Últimas ofertas de empleo

2010-04-28
[Comercial - Compras - CORDOBA.](#)

2010-04-25
[Otras Sin catalogar - MADRID.](#)

2010-04-25
[Atención a cliente - Call Center - MADRID.](#)

2010-04-21
[Comercial - Ventas - MADRID.](#)

2009-07-31
[T. Información - Operador \(día / noche\) - BARCELONA.](#)

Anuncios Google

- § Normativas y Procedimientos de generación de entregables
- § Normativas de calidad del Cliente
- § Pruebas de Certificación de Calidad que deben superar los entregables.

Requisitos de Almacenamiento

Aquí se recogen todos los requisitos que especifican el cómo, dónde y cuándo guardar los datos persistentes del sistema, así como la capacidad del sistema de almacenamiento de los mismos, su seguridad, su fiabilidad, su protección contra fallos o intento de acceso no autorizado y su política de respaldo.

Requisitos de Trazabilidad

Aquí se recogen los requisitos de trazabilidad de todas las acciones del sistema, como pueden ser:

- § Acciones de las que se deben guardar trazas en el sistema
- § Formato y medio de almacenamiento de cada tipo de trazas
- § Gestión de la caducidad de las trazas y de sus históricos
- § Control y medio de acceso a las trazas del sistema.

Requisitos de Seguridad

Aquí se recogen todos los requisitos relativos a la seguridad del sistema, como pueden ser:

- § Control de acceso al sistema y autenticación de usuarios
- § Políticas de usuarios y contraseñas, si las hubiere.
- § Desactivación de usuarios.
- § Control y auditoría de las acciones de los usuarios
- § Políticas de gestión de la seguridad y de los elementos y funcionalidades del sistema.
- § Métodos de agrupación de usuarios, y de permisos. Esquemas de administración y almacenamiento de la seguridad
- § Gestión de los roles de los usuarios, si hubiese.
- § Medidas de protección del sistema frente a ataques externos
- § Normativas y protocolos de seguridad que debe cumplir el sistema
- § Auditorías de seguridad y alarmas.

Requisitos de Escalabilidad

Aquí se recogen los requisitos de capacidad del sistema y cómo se debe poder ampliar si es necesario. Si el sistema es utilizado por múltiples usuarios simultáneos, debe disponer de un plan para redimensionar el sistema al crecer el número de usuarios.

Requisitos Legales y Normativas

En este apartado se recogen los requisitos legales que debe cumplir el sistema, es decir, toda la normativa legal que aplica al sistema, las restricciones legales de su uso y las normativas de gestión de la información confidencial. También se incluyen los requisitos de destrucción de información confidencial al final del ciclo de vida de la misma.

3.3 Requisitos de sistema

En este apartado se recogen todos los requisitos que debe cumplir el sistema, independientemente de la funcionalidad que debe cubrir.

Requisitos de Arquitectura

Estos requisitos definen arquitectura y componentes del sistema, cuando es un sistema creado a partir de varios, modúlalos.

Requisitos de Software

En este apartado recogemos todos los requisitos de software que se aplicarán al sistema, como pueden ser:

Sistemas operativos de los diferentes módulos que forman el sistema, incluyendo versiones

y actualizaciones
Software adicional necesario en el sistema
Requisitos de actualización del software de base del sistema

Requisitos de Hardware

Aquí se recogen todos los requisitos de hardware de todos los componentes del sistema. Entre ellos tipos de servidores, configuración de los mismos, tipos de unidades de disco y configuración, sistemas de backup, etc.

Requisitos de Comunicaciones

Aquí recogemos los requisitos necesarios para interconectar nuestro sistema, tanto interiormente, si es complejo, como exteriormente. Definiremos los parámetros de red requeridos, topología, tipos de enlace, anchos de banda, etc.

Requisitos de Integración

En este apartado recogeremos los requisitos de integración de nuestro sistema con otros sistemas externos o del cliente. Debemos incluir los protocolos que se deben soportar, los servicios que nos proporcionan o que debemos proporcionar, las aplicaciones con las que debemos interactuar, etc.

Requisitos de Contingencia

Aquí enunciaremos los requisitos que debe cumplir nuestro sistema en caso de contingencia, y que servirán de base para desarrollar el plan de contingencia del sistema.

4. Captura de requisitos

Una vez que ya sabemos lo que necesitamos recoger, hay que realizar un proceso de captura de requisitos. Los requisitos nos los van a proporcionar diferentes figuras del cliente, ya que debemos satisfacer las demandas de todos ellos.

4.1 El responsable del usuario

El responsable del usuario es la figura que conoce los requisitos de negocio del sistema a implementar. Sabe qué demanda debemos cubrir, luego conoce el “QUÉ” debemos hacer.

4.2 El responsable de TI

El responsable de TI del cliente es el que conoce el estado actual de la tecnología dentro de las instalaciones del cliente, y hacia dónde quiere enfocar su mejora. Es el que nos va a responder al “CÓMO” realizar el sistema.

4.3 El responsable financiero

El responsable financiero del cliente es la figura que conoce cuánto se puede gastar para implementar el sistema y cuándo puede desembolsar dicha cantidad. Él por tanto conoce el “CUÁNTO” y el “CÚANDO” se podrá abordar el desarrollo.

5. Métodos de captura

Una vez que sabemos lo que queremos buscar, es decir, los requisitos que nos hace falta encontrar, hay que utilizar diferentes técnicas para recoger dichos requisitos.

5.1 Extracción de requisitos desde la información proporcionada por el cliente

Normalmente el cliente comenzará proporcionándonos algo de información dentro de la primera petición. Dicha información puede tener un carácter heterogéneo, mezclando datos actuales de la empresa con documentos que recogen la idea de lo que necesitan.

5.2 Entrevista con el Cliente

Una vez analizada la documentación inicial, se debe comenzar la ronda de entrevistas con el cliente. Para ello seleccionaremos interlocutores válidos en cada una de las tres áreas: TI,

financiera y de negocio, para ir obteniendo la lista de requisitos del sistema. De cada reunión deberíamos crear un documento de acta de reunión, y un documento global de requisitos del sistema, que servirá de referencia para todo el desarrollo del sistema.

5.3 Observación del usuario

Es muy importante observar al futuro usuario del sistema en su propio entorno para saber qué tareas realiza actualmente y cuáles debe cubrir el sistema a desarrollar. Cada organización realiza tareas de una forma particular, algunas veces buenas, otras mejorables, pero suele ser muy complicado cambiar por completo la forma de trabajar de la organización. Por ello normalmente el sistema se adapta a la organización, y no la organización al sistema.

Esta observación se hace especialmente importante y obligatoria si desconocemos el negocio del cliente o la operativa habitual para este tipo de usuarios. Si no se hace esta observación pueden pasarse detalles por alto que luego serán difíciles de implementar. Y como dice el doctor House: “El paciente siempre miente”. Con esto quiero decir que el usuario te va a decir siempre que es todo muy sencillo y se le van a “olvidar” esas casuísticas que luego serán muy difíciles de implementar.

5.4 Observación de sistemas semejantes

Otra técnica de extracción de requisitos es realizar el mismo proceso en otro entorno diferente, quizás en otra organización a la que tengamos acceso. Probablemente no sea extrapolable directamente los requisitos obtenidos, pero sí nos puede dar una buena base para saber qué tenemos que buscar y obtener una colección de requisitos de partida que permita agilizar la toma de requisitos.

5.5 Consulta a los expertos

Otra técnica de toma de requisitos es realizar una entrevista a un experto en la materia. Podemos identificar dos tipos de expertos: los que conocen la parte funcional del sistema y los que conocen la parte técnica del sistema. De los expertos podremos obtener no sólo requisitos, sino también un conjunto de recomendaciones y buenas prácticas que nos evitarán problemas en nuestro desarrollo. También nos permitirán descubrir problemas y casuísticas ocultas, que al tenerlos en cuenta nos evitarán luego retrasos en los desarrollos.

5.6 Obtención de métricas representativas

Todo sistema desarrollado debería funcionar (bueno, eso es en teoría) y además hacerlo con un rendimiento adecuado. Para conocer ese rendimiento “adecuado” debemos tomar métricas del rendimiento del sistema que queremos desarrollar, bien tomando métricas de sistemas antiguos, si estamos sustituyendo a un sistema ya existente, o de sistemas semejantes.

5.7 Uso de prototipos

Otra técnica de captura de requisitos es la creación de prototipos del sistema. Los prototipos son sistemas teóricos o desarrollados que nos permiten presentar las ideas de lo que queremos realizar al cliente para su validación. De la observación del prototipo se obtendrán nuevos requisitos para completar la lista.

Hay varios tipos de prototipos:

- § **Prototipos teóricos.** Serían diseños en papel (documento, presentación, páginas Web o similar) en la que se explica mediante pantallas y diagramas la futura funcionalidad del sistema. El desarrollo de las herramientas de creación de documentación actuales (procesadores de texto, herramientas de presentación, herramientas de edición de páginas Web) permite hacer prototipos de sistemas informáticos fácilmente, que al menos nos darán una idea de cómo quedará el sistema.

- § **Prototipos no funcionales o de “cartón piedra”.** Son maquetas del sistema que muestra algunos datos fijos, y la interfaz de usuario. Están a medio camino entre un prototipo teórico y un prototipo funcional. Si el sistema incluye una interfaz de usuario avanzada se puede hacer un prototipo no funcional con

alguna herramienta de desarrollo rápido que nos permita validar los requisitos funcionales de interfaz de usuario y los principales procesos de negocio.

§ **Prototipos funcionales.** Son sistemas que ya incluyen una cierta funcionalidad, normalmente desarrollados sobre una arquitectura más sencilla. Pueden servirnos para probar el futuro comportamiento del sistema o para realizar pruebas de concepto, en las que se pruebe si una cierta solución técnica es posible.

Es importante entender que los prototipos son descartables, es decir, no formarán parte del futuro sistema. Esto es así pues el prototipo debe ser realizado en el menor tiempo posible y con el menor coste (dos requisitos para el prototipo que no tienen porqué estar presentes en el sistema final).

5.8 Requisitos autoimpuestos

Este tema es bastante delicado. Los requisitos autoimpuestos son los requisitos que se añaden a la lista de requisitos sin que el cliente los haya solicitado. Normalmente son requisitos que restringen la construcción del sistema. Por ejemplo podemos incluir en este tipo de requisitos:

§ Impuestos por la organización. Se introducen por el modo de entender los desarrollos de la organización. Pueden ser requisitos que afecten a la metodología de desarrollo, a la gestión de la calidad o incluso a la tecnología.

§ Impuestos por el conocimiento del equipo de desarrollo. Pueden aparecer requisitos debido a las carencias técnicas del equipo que realiza el proyecto.

§ Impuestos por la costumbre. Son requisitos que se introducen sin ser contrastados, normalmente debidos al modo de trabajo de los desarrolladores. En ocasiones se añaden para amoldar el desarrollo de un proyecto a lo que interesa al equipo de desarrollo. Prima el interés personal sobre el interés del proyecto.

La detección de este tipo de requisitos es bastante complicada. Para ayudar a ello, cada requisito añadido a la lista de requisitos debería tener la referencia del documento, acta o anotación en la que quedó reflejada su introducción.

6. Árboles de decisión y plantillas de requisitos

Para realizar una captura eficiente de requisitos debemos llevar preparada una posible lista de requisitos a capturar, es decir, hacer los deberes antes de iniciar el proceso de captura.

Para ello podemos utilizar un par de técnicas que nos ayudarán a realizar este proceso:

§ Plantillas de requisitos. Si vamos a repetir el proceso de toma de requisitos con una cierta frecuencia conviene preparar una plantilla de requisitos, que nos facilite la tarea de la toma de requisitos. En esta plantilla podemos recoger los requisitos que hemos dado como ejemplo o los que sean necesarios para realizar la toma de requisitos.

§ Árboles de decisión. A medida que el proceso de toma de requisitos se hace más habitual, participando en más proyectos, conviene utilizar árboles de decisión que ayude a los nuevos integrantes del equipo a realizar la toma de requisitos de manera más rápida y fiable. El árbol de decisión se irá montando poco a poco con las diferentes características comunes y particulares de los proyectos en los que participemos.

7. Métodos de documentación

Como siempre, todo el proceso de toma de requisitos hay que documentarlo. Vamos a exponer algunos de los métodos de gestión de los requisitos más habituales:

7.1 Lista de requisitos

Es el método más simple. Consiste en crear una lista ordenada de requisitos, en la que asignaremos a cada requisito un identificador único. Se puede hacer a mano, pero es difícil controlarla.

Para la creación inicial de las listas podemos utilizar algún sistema que nos permita rápidamente gestionar los requisitos. Un ejemplo podría ser una simple hoja Excel o un

esquema de Word o similares. Si el documento no va a ser complejo, podemos comenzar por aquí, pero pronto nos daremos cuenta de que no es el mejor sistema, ya que por ejemplo, la trazabilidad y la gestión de las versiones pronto se complica. Nos obliga a realizar un montón de tareas a mano, muy repetitivas y propensas a errores. Aun así, muchas organizaciones utilizan este medio para presentar a sus clientes los requisitos de grandes proyectos...

7.2 Priorización y categorización de requisitos

Después de obtener la lista global de requisitos hay que clasificarlos y priorizarlos, ya que fijarán la importancia en su futuro desarrollo.

7.3 Matriz de trazabilidad

La matriz de trazabilidad es una técnica que liga los requisitos a los diferentes elementos del desarrollo. Por ejemplo podemos hacer una matriz de trazabilidad de cada requisito con los casos de uso en los que aparece. Dado que la matriz es muy grande, conviene hacer vistas de la misma si queremos incluirla en los documentos de análisis

Se pueden trazar varios tipos de matrices de trazabilidad. Algunos ejemplos pueden ser:

- Requisitos frente a fases de desarrollo del sistema
- Requisitos frente a Actores
- Requisitos frente a Casos de Uso
- Requisitos frente a entidades del sistema
- Requisitos frente a procesos de negocio
- Requisitos frente a interfaces del sistema
- Requisitos frente a módulos

Hay que tener en cuenta que las matrices de trazabilidad suelen tener un gran tamaño, por lo que conviene partirlas en matrices más pequeña, usando la agrupación de requisitos.

7.4 Control de versiones

Como todo documento del proyecto, el listado de requisitos debe estar controlado mediante algún sistema de control de versiones. Lo ideal es que elijamos un sistema que nos dé la siguiente información:

- Versiones del documento, con posibilidad de recuperar fácilmente versiones anteriores
- Ver las diferencias entre versiones
- Identificar las diferencias mediante un resumen o cuadro de modificaciones

7.5 Documento final

Debemos elegir un mecanismo para generar el documento final que será entregado al cliente. Este documento fija por contrato las funcionalidades que debe cubrir el sistema a desarrollar, por lo que es conveniente que cualquier alteración sobre el mismo quede documentada.

Debemos prestar especial importancia a:

- Formato del documento. Debe ser un formato estandarizado y de amplia difusión. Un ejemplo puede ser el formato PDF, pues es fácilmente generable y legible desde cualquier plataforma.
- Versión y control del documento. Ya que es habitual el intercambio de documentos con el cliente hasta la obtención de una versión definitiva es muy importante detallar la versión del documento y la lista de cambios entre versiones del mismo.
- Control de distribución. Debemos asegurarnos de que quedan reflejados en el documento la lista de interesados del mismo, así como la lista de personas que deben recibir copia del mismo.
- Control de aprobación. Debemos incluir también la lista de verificaciones y aprobaciones del documento. Lo ideal es que el documento fuese firmado. A estas alturas debería ser ya posible la firma digital de los mismos.
- El documento debe estar depositado en un sistema de gestión de la

documentación, de manera que quede accesible y claramente identificado por los interesados.

7.6 Validación de requisitos

El último paso en la gestión de requisitos es la validación de los mismos por el cliente. Una vez validados, los requisitos están cerrados y listos para ser enviados al equipo de análisis y desarrollo.

7.7 Aplicaciones de gestión de requisitos

Dado que la gestión de requisitos es un proceso crítico y complejo, ya que define contractualmente lo que nos comprometemos a desarrollar, la gestión de los mismos se suele hacer con ayuda de aplicaciones específicas para la gestión de requisitos.

7.8 Un ejemplo: REM

Un ejemplo de aplicación de gestión de requisitos es el desarrollo REM, que está cubierto en otro de los tutoriales ya publicados (ver <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=REM>) Esta herramienta permite hacer una gestión bastante sencilla de los requisitos de un proyecto, generando el documento final. Dado que utiliza XML como formato interno, se pueden usar sistemas de control de versiones típicos de código fuente para almacenarlos y compararlos.

8. Agrupación de requisitos

Antes de cerrar la etapa de toma de requisitos debemos realizar un análisis de los mismos. En esta etapa organizaremos y repasaremos los requisitos capturados, hasta obtener la lista final de requisitos. En esta etapa debemos organizar los requisitos según diferentes criterios.

- **Agrupación por alcance y fases.** Si el alcance del proyecto es muy amplio conviene dividir el proyecto en fases, limitando el alcance de cada fase hasta cubrir el alcance final. En este caso la primera organización de los requisitos es identificar a qué fase afecta cada uno.
- **Agrupación funcional.** Otra agrupación importante es la agrupación funcional, en la que agrupamos requisitos que tienen una relación funcional directa, por ejemplo según el tipo de datos que van a manejar o según los procesos de negocio que van a cubrir.
- **Agrupación modular.** En este caso primero dividimos el sistema en módulos y luego identificamos a qué módulo afecta cada requisito. Esta división permite detectar el alcance de cada requisito, delimitando las partes del sistema a las que afecta.
- **Agrupación por producto.** Si el sistema se compone de productos separados, los requisitos pueden quedar también asociados a alguno de los productos, limitando también su alcance, como en el caso anterior.

9. Casos de uso

El uso de los casos de uso, ideados por Ivar Jacobson, es una buena herramienta para la captura y el análisis de requisitos. Los casos de uso permiten definir de forma sencilla el comportamiento del sistema desde el punto de vista de interacción con el usuario u otros sistemas.

Un caso de uso es una descripción textual de la secuencia de interacción de los usuarios (actores) con el sistema. Al escribir el caso de uso se considera al sistema como una “caja negra”, ya que no nos importa lo que sucede dentro sino las posibles interacciones entre el sistema y el usuario. Los casos de uso son documentos de texto breves (normalmente caben en una página) en la que se va detallando los posibles caminos de interacción entre el usuario y el sistema.

Los casos de uso los explicaremos en detalle en la segunda parte de este artículo, debido a su extensión.

10. Conclusión

En este artículo hemos dado un repaso a las técnicas de gestión de requisitos de un sistema,

su descripción, su captura y su gestión. También hemos visto cómo y de dónde obtener los requisitos de nuestro sistema. Poco a poco iremos perfeccionando nuestras técnicas de captura de requisitos, de manera que en el desarrollo de nuestros proyectos no nos llevemos sorpresas. También hemos visto los casos de uso, una técnica muy usada para generar y validar requisitos del sistema para la parte de interfaz del mismo. Con este tutorial hemos ampliado el anterior tutorial dedicado a la herramienta de gestión de requisitos REM:

¿Qué te ha parecido el tutorial? Déjanos saber tu opinión y ¡vota!

Muy malo

Malo

Regular

Bueno

Muy bueno

Votar

(Sólo para usuarios registrados)

» Regístrate y accede a esta y otras ventajas «

Anímate y coméntanos lo que pienses sobre este tutorial

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» Regístrate y accede a esta y otras ventajas «

Autor

Mensaje de usuario registrado

- Puedes inscribirte en nuestro servicio de notificaciones [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAITrabajo en XING [haciendo clic aquí](#).
- [Añadir a favoritos Technorati.](#)



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)



Recuerda

[Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Formación en nuevas tecnologías

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	Valoración	Votos	Pdf
Gestión de los requisitos	Cuando vamos a abordar el desarrollo de un proyecto, ya sea de ingeniería de software o de cualquier campo, una de sus primeras etapas es la definición de los requisitos del proyecto.	2010-05-05	22	-	-	
Aprendiendo Objective-C desarrollando para nuestro iPhone 3Gs	En este tutorial veremos que aunque el lenguaje y entorno para el iPhone puedan sernos totalmente nuevos hay decenas de posibles combinaciones con las aplicaciones empresariales que habitualmente nos piden.	2010-02-16	1964	-	-	
Creando la Baraja de SCRUM de Autentia como aplicación para el iPhone 3G.	En este tutorial, se me ha ocurrido que podría hacer una pequeña aplicación útil: el pasar a iPhone la baraja de estimación que utilizamos en nuestras reuniones Scrum	2010-02-09	1501	-	-	
JavaBean Datasource Ireport	La particularidad del caso que nos ocupa, es conseguir que la fuente de datos del informe sea una lista de JavaBeans y no una consulta definida previamente en el informe.	2009-12-14	2736	Bueno	1	
Instalar OpenESB 2.1 e Introducción	En este tutorial veremos como descargar e instalar OpenESB y explicaremos sus funcionalidades	2009-12-03	2863	-	-	
Cómo conseguir que Subversion avise a Hudson para lanzar una build	En este tutorial vamos a ver como configurar Subversion para que sea este el que avise a Hudson cada vez que hay un commit, y así se lance la build.	2009-10-27	3330	-	-	
Cómo instalar Hudson en Apache Tomcat	Instalar Hudson en Apache Tomcat	2009-10-26	3915	Muy bueno	1	
Enlazar Bugzilla con MavenChangesPlugin	En este tutorial veremos como enlazar Bugzilla con MavenChangesPlugin	2009-09-11	1473	-	-	
Release Bugzilla Maven Plugin	En este tutorial vamos a mostrar como automatizar un conjunto de acciones que hay que hacer siempre en los sistemas de gestión de incidencias, tales como dar de alta una nueva versión del producto, cerrar las incidencias que soluciona la nueva versión, et	2009-09-11	1824	-	-	
Ordenación por cantidades en informe cruzado	Nico nos explica en ese tutorial cómo lograr ordenar por cantidades en informes cruzados usando JasperReports e iReport	2009-08-26	2567	-	-	

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.