

por Nacho Iborra

Entornos de Desarrollo

Bloque 1

Tema 4: Especificación de los requisitos de una aplicación

1.4.1. Inicio de un proyecto software

Cuando una empresa de desarrollo toma la decisión de iniciar un proyecto software, hay que tener en cuenta factores como: ¿de dónde parte la idea? ¿quién va a ser el encargado o jefe del proyecto?

1.4.1.1. Origen del proyecto

En cuanto al origen de la idea, ésta puede venir de:

- La propia empresa, con sus desarrollos internos. En este caso, la empresa hace de su propio cliente
- Una entidad externa que solicita el producto, en cuyo caso la empresa hace de suministrador de software

En ambos casos, se necesita tener una imagen clara del producto a realizar, y valorar la viabilidad del proyecto. Esto es más fácil en el caso de entidades externas, ya que el contrato con el cliente se puede ajustar a lo que la empresa necesita para que el producto le sea rentable. En el caso de desarrollos internos, al no tener fuente de ingresos externa, se debe valorar si merece la pena dicho desarrollo.

1.4.1.2. El jefe del proyecto

En cuanto al jefe del proyecto, a la hora de seleccionarlo hay que tener en cuenta algunas cuestiones, tales como:

- El futuro jefe de proyecto debería intervenir en el propio proceso de decisión de emprender el proyecto
- La persona elegida debe tener un buen nivel en cuanto al negocio de que trata el proyecto, relaciones interpersonales, competencias personales y gestión de grupos.

Es el jefe del proyecto quien debe, inicialmente, realizar tareas como:

- Identificar las posibles áreas de riesgo del proyecto
- Establecer presupuestos, calendarios y asignaciones de tareas
- Dar soporte y proporcionar técnicas de comunicación entre los miembros del equipo

- Establecer cómo interactuar con el cliente

1.4.2. Recogida de información

Como parte del proceso de decisión de emprender un proyecto software, está la comunicación con el cliente para averiguar qué quiere exactamente. En este proceso, tenemos dos inconvenientes:

- Los desarrolladores no saben (y es difícil que lleguen a saber con exactitud) qué quiere el cliente
- Los clientes no saben qué información relevante proporcionar para el desarrollo de la aplicación

Para intentar paliar estos aspectos, hay que:

- Identificar las fuentes de información del cliente que puedan resultar relevantes
- Realizar las preguntas apropiadas para comprender las necesidades del cliente
- Confirmar con el cliente lo que parece haberse comprendido de los requisitos
- Sintetizar los requisitos obtenidos en un documento llamado especificación de requisitos del sistema (ERS).

1.4.2.1. Técnicas de recogida de información

Existen diversas formas de obtener información del cliente para definir los requisitos y valorar así la viabilidad del proyecto. Algunas de las más empleadas (y no necesariamente excluyentes) son:

- **Entrevista:** quizá sea la más empleada, aunque para que sea exitosa requiere que el analista tenga gran preparación y experiencia
- **Desarrollo conjunto de aplicaciones (JAD):** clientes y analistas se reúnen conjuntamente para trabajar en las características que debe tener el software. De esta forma, el cliente se involucra más en el proyecto
- **Observación:** consiste en analizar "in situ" (en la empresa cliente) cómo funciona la organización y cómo encaja en ella el producto que se quiere desarrollar
- **Estudio de documentación:** estudio de los documentos oficiales de la empresa cliente que describan su funcionamiento
- **Cuestionarios:** útiles para recoger información de muchas personas en poco tiempo. También es útil cuando los trabajadores con los que necesitamos comunicarnos están muy dispersos geográficamente
- **Brainstorming o tormenta de ideas:** se utiliza cuando no están claras las necesidades, de forma que podemos identificar así un primer conjunto básico de requisitos.

Veremos ahora con algo más de detalle algunas estas técnicas.

1.4.2.2. Las entrevistas

Una entrevista es una forma sistemática de obtener información de otra persona, mediante una conversación estructurada. Este aspecto es importante, ya que una entrevista no consiste simplemente en hacer preguntas, sino que la estructura y orden de las mismas, y la forma en que se plantea la conversación, determinan en gran parte la información que se obtiene.

Una buena entrevista debe organizarse en:

- **Preparación de la entrevista**, donde nos documentaremos sobre la situación del cliente, identificaremos a qué personas entrevistar, y prepararemos el objetivo y contenido de la(s) entrevista(s), incluyendo hora y lugar. En cuanto a las personas a las que entrevistar, se suele empezar por los directivos y terminar en los empleados, y a veces se realiza alguna "entrevista de cortesía" a algún jefe de sección que, aunque no vaya a aportar nada nuevo, facilitará el camino para obtener entrevistas a los empleados a su cargo.
- **Realización de la entrevista**. Esta fase es el núcleo de la entrevista, y en ella se distinguen tres etapas:
 - **Apertura o presentación**, donde nos presentamos al cliente, le informamos de la razón de la entrevista y de la mecánica a seguir. Así el entrevistado tiene una primera impresión y se genera un ambiente más confortable
 - **Desarrollo de la entrevista** en sí. Al principio puede haber preguntas más abiertas, para luego ir a otras más específicas (que se contesten con un "sí" o un "no", o tengan respuestas muy concretas). Es aconsejable utilizar una terminología adecuada a nuestro interlocutor (no usar tecnicismos con quien no los entiende), y repetir las respuestas dadas para verificar que quedan claras y era eso lo que se quería decir.
 - **Finalización de la entrevista**, donde se hace un resumen de la misma, se agradece el tiempo dedicado al entrevistado, y se deja abierta la posibilidad de volver a contactar con él.

1.4.2.3. El desarrollo conjunto de aplicaciones (JAD)

Esta otra técnica de recolección de información promueve, como hemos dicho, la cooperación entre el cliente y el analista. Consiste en una serie de reuniones o workshops de unos 2 a 4 días de duración cada una, donde participan varios usuarios de la empresa cliente junto con varios analistas.

De esta forma, se evita tener que corregir después posibles errores o faltas de información de las entrevistas, y facilita que el cliente comprenda mejor el sistema que se está siguiendo para desarrollar su producto.

Cada uno de los workshops consta de las siguientes etapas:

- **Preparación**, donde el jefe del proyecto selecciona a los participantes a intervenir (analistas y clientes), recaba información sobre el sistema a construir y organiza la

reunión (lugar, fecha, materiales...)

- La **sesión JAD** propiamente dicha, que consta de varias reuniones con una temática bien estructurada. Al final de la sesión se tiene un documento de especificación de requisitos que debe ser aprobado por los asistentes.
- **Documentación**, a partir del documento de especificación de la sesión JAD, se perfilan algunos detalles y se le da un formato final adecuado.

Sin embargo, no es una técnica que se utilice mucho, ya que requiere una mayor organización, e implica romper el ritmo de trabajo del cliente para acudir a los workshops cada cierto tiempo.

Ejercicios propuestos:

1.4.2.1. Responde a las siguientes afirmaciones indicando si son verdaderas o falsas, y razonando la respuesta:

- Las entrevistas son la forma más recomendable de obtener información para un proyecto software, en general
- Mediante JAD se pretende obtener de forma rápida una especificación de requisitos para proyectos donde intervienen diferentes departamentos o secciones de la empresa cliente
- Los cuestionarios son útiles para recabar información del cliente cuando sus componentes están dispersos geográficamente
- Mediante únicamente la observación *in situ* o el estudio de documentación se podría obtener una ERS muy completa

1.4.3. Especificación de requisitos

A partir de las necesidades recogidas del cliente, se debe llegar a una definición de los requisitos necesarios, tanto a nivel de hardware como a nivel de software. Así, se elabora, tras la recogida de información, un documento llamado **especificación de requisitos del sistema (ERS)**, que describe con el mayor detalle posible lo que el futuro producto debe hacer.

Antes de seguir, conviene concretar qué entendemos por requisito: un **requisito** es una condición que necesita el cliente para resolver un problema o lograr un objetivo. De esta forma, se definen las restricciones sobre lo que hará el sistema.

Por ejemplo, un agente de bolsa que nos pida una aplicación para invertir, puede que nos indique que el tiempo de respuesta de la aplicación sea rápido para no perder la oportunidad de comprar o vender acciones a un determinado valor. Esa rapidez de respuesta sería un requisito de la aplicación a desarrollar.

1.4.3.1. Tipos de requisitos

Distinguimos principalmente tres tipos de requisitos:

- **Del sistema:** se refiere a los componentes físicos (hardware) que serán necesarios para implantar la aplicación, como por ejemplo número de ordenadores donde implantarlo, procesador, espacio en disco, etc.
- **Funcionales:** indican lo que hace o se espera que haga el sistema. Aquí se incluyen los procesos que se espera que haga la aplicación, las entradas que recibirá, las salidas esperadas, las interacciones con el usuario, los datos que deben mantenerse, etc. Por ejemplo, que la aplicación nos pida que nos identifiquemos antes de entrar, o que periódicamente nos envíe e-mails con estadísticas, son ejemplos de requisitos funcionales
- **No funcionales:** indican lo bien o mal que el sistema cumple los requisitos funcionales. Aquí hablaríamos de criterios de rendimiento, seguridad, volumen de datos soportado, etc. Este tipo de requisitos son menos críticos que los funcionales, ya que no provocan que el sistema deje de funcionar, sino que degradan su funcionamiento (lo ralentizan, lo hacen más inseguro, etc.)
 - Dentro de este tipo de requisitos, un grupo particular son los requisitos de facilidad de uso, que miden el acoplamiento entre el sistema y los usuarios que lo utilizan. Para lograr una buena facilidad de uso, conviene analizar a los usuarios, ver sus características (nivel de destreza, entorno de trabajo, etc.), las tareas que realizarán en la aplicación, la usabilidad de las pantallas (por ejemplo, que una ventana no tenga demasiados botones), e incluso factores situacionales (por ejemplo, usabilidad para personas zurdas).

Ejercicios propuestos:

1.4.3.1. Lee el siguiente enunciado sobre el desarrollo de un producto software, e identifica los requisitos de sistema, funcionales y no funcionales: *Un blog tiene tres tipos de usuarios: administradores, editores y visitantes. Cualquiera de ellos debe loguearse para entrar a la aplicación. Los administradores pueden dar de alta a otros usuarios, los editores pueden publicar posts, y los visitantes pueden comentar los posts de los editores. Se espera que el blog tenga una gran afluencia, por lo que la disponibilidad del servicio debe de ser alta. También se necesitarán tener varios equipos servidores para balancear la carga y repartirse los accesos de los clientes, evitando la saturación. Por otra parte, al no contar con editores demasiado experimentados, se desea que la interfaz de usuario para que éstos puedan publicar los posts sea lo más sencilla posible. Toda la información se almacenará en una base de datos MySQL, con copias de respaldo diarias. Las contraseñas de los usuarios estarán encriptadas para evitar que se puedan descifrar.*

1.4.3.2. A partir de la siguiente propuesta de software a desarrollar, identifica los requisitos de sistema, funcionales y no funcionales. *Un centro cultural de un municipio se dedica al préstamo de dos tipos de materiales de préstamo: discos y libros. Necesitamos una aplicación que nos permita dar de alta materiales de ambos tipos. Por*

otra parte, a este centro cultural acuden clientes, que se identificarán en la aplicación con un login y password. Después, deben poder buscar los materiales por tipo (disco o libro) y/o título del material. Una vez seleccionado el material, deben poderlo pedir, o reservar si no está disponible. Los clientes pueden tener hasta 5 materiales en préstamo simultáneamente. Al ser un municipio pequeño, no se espera que el sistema esté muy solicitado, y basta con un simple equipo y una pequeña base de datos para almacenar la información. En el caso de que un cliente no recuerde su contraseña, se le volverá a enviar al correo que indicó al darse de alta en el sistema.

1.4.3.3. Indica de qué tipo es cada uno de los siguientes requisitos (de sistema, funcional o no funcional):

- La aplicación debe guardar los datos antes de cerrarse
- El servidor de base de datos debe estar separado del de la aplicación
- Si no hay conexión con el servidor remoto, se deberá mostrar un mensaje de error y cerrar la aplicación
- El tiempo de respuesta deseable no debe exceder los 10 milisegundos
- Se debe reservar el suficiente ancho de banda para enviar el vídeo en streaming en tiempo real
- Se deben verificar que los datos introducidos por el usuario (nombre, DNI y teléfono) son correctos