

Fundamentos de Ingeniería del Software



Capítulo 3. Análisis de Requisitos

Cap 3. Análisis de Requisitos Estructura



1. Actividades iniciales.
2. Técnicas de recogida de la información.
3. Requisitos y análisis de requisitos.
4. Actividades generales de análisis de requisitos.
5. Documentos de especificación de requisitos.
6. Análisis estructurado.
7. Introducción a los casos de uso.
8. Prototipado.

Cap 3. Análisis de Requisitos

Bibliografía



- De los puntos 1 a 5...
 - (Piattini et al. 96) (Piattini et al. 04)
 - Cap. 6 y Cap. 7 (aptdos. 7.1 y 7.2, este último no con tanto nivel de detalle)

1. Actividades iniciales

Análisis de necesidades y estudio de viabilidad:

Decisión de emprender el proyecto



Recoger información sobre el proyecto
(Directivos nivel alto/medio)

(Técnicas
recogida
información)



Informe de necesidades



Estudio de la viabilidad del proyecto
(Análisis de factibilidad)

Estudio de viabilidad



- Exige bastante experiencia
- Enumerar alternativas.
- Evaluación de las alternativas:
 - Económico (¿Los beneficios compensan los costes?)
 - Técnico (¿Se encuentra disponible la tecnología necesaria?)
 - Legal (¿Se atenta contra alguna ley o reglamento? p.e. LOPD, Ley Orgánica de Protección de Datos)
 - Operativo (¿Puede coordinarse con los métodos ya existentes? ¿Encaja en la filosofía de la empresa?)
- Es posible que después de analizar la viabilidad del proyecto, se desestime. El dinero que ya se ha invertido en el análisis de viabilidad no debería condicionar esta decisión.
- Si no se desestima,
 - Especificación detallada de la alternativa seleccionada.
 - Definición del plan inicial del proyecto.

Estudio viabilidad - Alternativas



- Comprar un producto software comercial, ya construido, que cumpla los requisitos marcados (COTS, *Commercial Off-The-Shelf*)
- Desarrollar el producto internamente.
- Desarrollarlo de forma externa mediante un contrato (*outsourcing*).
- Automatizar parcialmente el sistema, para reducir gastos.

Plan tentativo del proyecto



- Identifica:
 - Áreas de riesgo
 - Presupuestos, calendarios, planes de trabajo del personal y asignación de tareas.
 - Soporte necesario para el equipo del proyecto.
 - Técnicas de comunicación entre los componentes del proyecto.
 - Forma de interactuar con el cliente.

2. *Técnicas de recogida de información*



- Entrevistas
- JAD (*Joint Application Design*)
- Prototipado
- Observación
- Estudio de documentación
- Cuestionarios
- Tormenta de ideas (*brainstorming*)
- ...

JAD - Desarrollo conjunto de aplicaciones

Conjunto de reuniones usuarios/analistas:

2 - 4 días

Dinámica de grupos

Se comienza con un doc. de trabajo, y se discute



Al final del JAD



Doc. de requisitos (aprobado)

Entrevistas vs. JAD



Entrevistas:

- Requieren mucho tiempo (prepararlas, hacerlas, y elaborar conjunto coherente de requisitos a partir de diferentes entrevistados).
- Más difícil detectar errores (sólo analista revisa).

JAD:

- Participación más profunda usuarios (se identifican con el sist.)
- Más difícil llevar a la práctica.
- Requiere más organización.
- Empíricamente: ahorro tiempo ↑↑, satisfacción usuarios ↑↑

3. Requisitos y análisis de requisitos

REQUISITO:

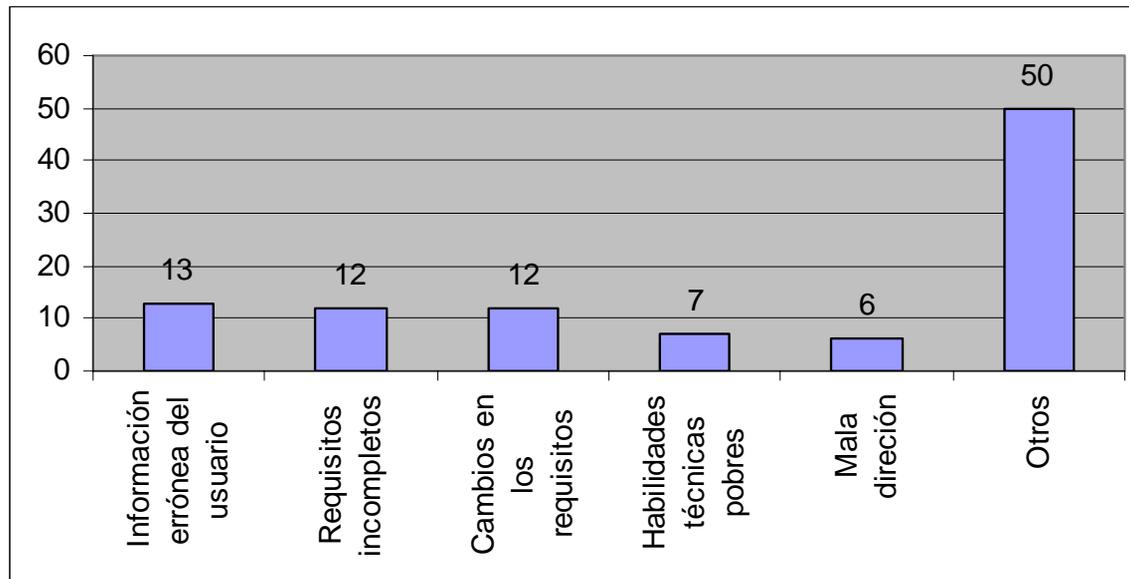
- Condiciones que debe cumplir un sistema para satisfacer un contrato, una norma o una especificación.
- Condición o capacidad que necesita el usuario para poder resolver un problema o conseguir un beneficio determinado.

AR:

- “El proceso de estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hw. o de sw.”
- “El proceso de estudio y refinamiento de dichos requisitos” [IEEE Std. 610, Glosario estándar de términos en ingeniería del software]

Importancia del análisis de requisitos

- Los problemas con los requisitos constituyen la principal fuente de problemas (37%)



Factores del coste en proyectos software reales
(Standish94, <http://www.standishgroup.com/chaos/toc.php>)

Importancia del análisis de requisitos (II)



- En diciembre de 1997, un estudio de 500 directores de proyectos en tecnologías de la información (en RU y EE UU), informó que el 76% había estado involucrado en algún proyecto que había fracasado totalmente
(*Computer Industry Daily*, December 12, 1997)

⇒ La causa del fracaso más frecuentemente nombrada era "requisitos de usuario cambiantes"

Requisitos funcionales y no funcionales (Sommerville 2004)

- **Requisitos funcionales:** describen la funcionalidad o los servicios que se espera que el sistema proveerá, sus entradas y salidas, excepciones, etc.

Ejemplos:

- 1.- "El usuario deberá tener la posibilidad de buscar en el conjunto inicial de la base de datos o seleccionar un subconjunto de ella."
- 2.- "El sistema deberá ofrecer un explorador (*browser*) para que el usuario lea documentos en el almacén de documentos."

- **Requisitos no funcionales:** se refieren a las propiedades emergentes del sistema como la fiabilidad, el tiempo de respuesta, la capacidad de almacenamiento, la capacidad de los dispositivos de entrada/salida, y la representación de datos que se utiliza en las interfaces del sistema.

Ejemplos:

- 1.- "Será necesario que la comunicación requerida entre el APSE y el usuario se pueda expresar utilizando el conjunto de caracteres estándar de ADA."
- 2.- "El proceso de desarrollo del sistema y los documentos a entregar estarán sujetos al proceso y a los productos a entregar definidos en XYZCo-SP-STAN-95."
- 3.- "El sistema no deberá revelar a sus operadores información personal alguna de los clientes excepto su nombre y número de referencia."

Más sobre los requisitos funcionales



- Deberían definir las acciones fundamentales que tienen que tener lugar en el software, aceptando y procesando las entradas y su procesamiento y generación de salidas
 - Pruebas de validez en las entradas
 - Secuencia exacta de operaciones
 - Respuestas a situaciones anormales, incluyendo: desbordamientos, facilidades de comunicación, manejo de errores y recuperación
 - Efecto de los parámetros
 - Relaciones de salidas a entradas, incluyendo secuencias de entrada/salida y fórmulas para la conversión entre entrada y salida
- Puede ser apropiado partir los requisitos funcionales dentro de subfunciones o subprocesos. Esto no implica que el diseño de software tenga que ser partido de esa forma.
- Son generalmente listados como sentencias del tipo “deberá”, comenzando con “El sistema deberá...”.

¿Son funcionales o no funcionales?



- “Los participantes tendrán que ser mayores de edad”
- “Los participantes tendrán que residir en la península”
- “El valor de las pujas será mostrado en euros”
- “El IVA aplicado a las pujas ganadoras será del 7%”
- “La lista de resultados estará preparada para ser impresa en un folio tamaño A4”
- “El sistema lanzará una excepción en caso de que un usuario quiera cargar un importe mayor que el saldo de la cuenta”
- “Los accesos a la BD deberán usar el estándar SQL-92”
- “Los registros de la BD no deben ocupar más de 4 Kb”
- “El sistema deberá ser capaz de interactuar con 100 usuarios concurrentes”

4. *Actividades generales de análisis de requisitos*



- **Extracción** (*elicitación*) de requisitos: se descubren, revelan, articulan y comprenden los requisitos, usando técnicas de recogida de información.
- **Análisis** de requisitos: razonamiento sobre los requisitos, se descubren inconsistencias, etc.
- **Especificación** de requisitos: redacción o registro de los requisitos, en lenguaje natural o usando alguna técnica
- **Validación** de los requisitos: los clientes o usuarios comprueban que los requisitos son correctos: válidos, consistentes y completos

5. Documentos de especificación de requisitos

Después de realizar el informe de necesidades y de dar luz verde al proyecto, se crea el **SyRS** (***System Requirements Specification***) (si procede) y el **SRS** (***Software Requirements Specification***) o **ERS** (***Especificación de Requisitos del Software***)

