



USAL
UNIVERSIDAD
DEL SALVADOR

Facultad de Ingeniería

Seminario de Integración Profesional

Especificación de Requerimientos

Lic. Christian López Pasarón



Análisis de requisitos en el desarrollo de software

- Es una tarea crucial en el ciclo de vida del desarrollo de software.
- Los requisitos son las necesidades del producto que se debe desarrollar.
- En la fase de análisis se deben identificar y documentar claramente estas necesidades.
- El resultado de esta fase es un documento de especificación de requisitos en el que se describe lo que el futuro sistema debe hacer.
- El análisis de requisitos implica tanto una actividad de análisis como de síntesis.



Análisis de requisitos en el desarrollo de software

- Los requisitos pueden ser definidos como las condiciones o capacidades que necesita el usuario para resolver un problema o conseguir un objetivo determinado.
- La determinación de los requisitos requiere la participación activa de los usuarios y los analistas para llegar a un acuerdo en las necesidades del nuevo sistema.
- El documento de especificación de requisitos debe ser legible por el cliente para evitar malentendidos y asegurar su participación activa en la extracción de dichos requisitos.



Modelado de la futura aplicación

- Basándose en los requisitos, se procederá al modelado de la futura aplicación.
- Se pueden utilizar diferentes tipos de metodologías para el modelado, tales como la metodología orientada a objetos.
- La metodología orientada a objetos utiliza el UML para representar diagramas (casos de uso) que permiten definir el sistema desde el punto de vista del usuario.
- Los diagramas establecen las relaciones entre el futuro sistema y su entorno en forma de acciones del usuario y reacciones del sistema.



Objetivos de la ERS

La Especificación de Requisitos de Software (ERS) tiene como principales objetivos:

- Ayudar a los clientes a describir claramente lo que se desea obtener mediante un determinado software
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente
- Servir de base para desarrollos de estándares de ERS particulares para cada organización



Ventajas de una buena ERS

Una buena especificación de requisitos software ofrece una serie de ventajas, entre las que destacan:

- Contrato claro entre cliente y desarrolladores
- Reducción del esfuerzo en el desarrollo
- Buena base para la estimación de costos y planificación
- Punto de referencia para procesos de verificación y validación
- Base para la identificación de posibles mejoras en los procesos analizados



Contenido de una ERS

Una ERS forma parte de la documentación asociada al software que se está desarrollando, por tanto debe definir correctamente todos los requerimientos, pero no más de los necesarios. Esta documentación no debería describir ningún detalle de diseño, modo de implementación o gestión del proyecto, ya que los requisitos se deben describir de forma que el usuario pueda entenderlos. Al mismo tiempo, se da una mayor flexibilidad a los desarrolladores para la implementación.

Lenguaje utilizado

El grado y el lenguaje utilizado para la documentación de los requisitos estarán en función del nivel que el usuario tenga para entender dichas especificaciones. Es importante que la documentación sea clara y concisa, evitando tecnicismos y términos que puedan generar confusiones. La ERS debe ser entendible tanto para el cliente como para los desarrolladores.



Características de una buena ERS

Las características deseables para una buena especificación de requisitos software que se indican en el IEEE son [Chalmeta, 2000][Piattini, 1996]:

- Correcta
- No ambigua
- Completa
- Verificable
- Consistente
- Clasificada
- Modificable
- Trazable
- Utilizable durante las tareas de mantenimiento y uso



Características de una buena ERS

- **CORRECCIÓN:** si y sólo si todo requisito que figura en ella refleja alguna necesidad real. La corrección de la ERS implica que el sistema implementado será el sistema deseado.
- **AMBIGÜEDAD:** La ambigüedad en la ers de software puede generar interpretaciones erróneas y aumentar los costos del proyecto.
 - Cada requisito debe tener una única interpretación
 - Utilizar términos únicos y claros para describir las características del producto final.
 - El lenguaje natural puede ser muy ambiguo. Los lenguajes formales son menos ambiguos, pero pueden ser más difíciles de aprender

Todos los clientes tienen el mismo campo de control

- 1.- ¿Todos tienen el mismo valor en el campo de control?
- 2.- ¿Todos los campos de control tienen el mismo formato?
- 3.- ¿Un campo de control se usa para todos los clientes?



Características de una buena ERS

- Completitud: Una ERS es completa si:
 - Incluye todos los requisitos significativos del software (relacionados con la funcionalidad, ejecución, diseño, atributos de calidad o interfaces externas).
 - Existe una definición de respuestas a todas las posibles entradas, tanto válidas como inválidas, en todas las posibles situaciones.
 - Cumple con el estándar utilizado. Si hay alguna parte del estándar que no se utiliza, se debe razonar el porqué no se ha utilizado.
 - Aparecen etiquetadas todas las figuras, tablas, diagramas, etc, así como definidos todos los términos y unidades de medida empleados.

La ERS debe ser siempre completa, aunque en ocasiones esto no será posible. Por ejemplo si todavía no se han determinado los formatos de los informes finales o por cualquier razón se está esperando la publicación de una ley o un reglamento sobre impuestos.



Características de una buena ERS

- Verificabilidad: Un requisito se dice que es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina puedan chequear que el software satisface dicho requerimiento.

No verificables:

El producto debería funcionar bien

El producto debería tener una buena interfaz de usuario

Verificable:

La salida se suministra dentro de los 20 segundos siguientes al evento en el 60% de las veces, y en los 30 segundos siguientes en el 100%



Características de una buena ERS

- Consistencia: Una ERS es consistente si y sólo si ningún conjunto de requisitos descritos en ella son contradictorios o entran en conflicto. Se pueden dar tres casos:
 - Requisitos que describen el mismo objeto real utilizando distintos términos.
 - Las características especificadas de objetos reales. Un requisito establece que todas las luces son verdes y otro que son azules.
 - Conflicto lógico o temporal entre dos acciones determinadas. Se llega a un punto en el que dos acciones serían perfectamente válidas (¿sumar o multiplicar?)



Características de una buena ERS

- Clasificación: No todos los requisitos son igual de importantes.

Lo ideal es el establecimiento de prioridades, de modo que la implementación de un requisito de menor prioridad no emplee excesivos recursos.

Los requisitos pueden clasificarse por diversos criterios:

- Importancia: Pueden ser esenciales, condicionales u opcionales.
- Asiduidad: Que tanto se utiliza esa funcionalidad
- Estabilidad: Cambios que pueden afectar al requisito



Características de una buena ERS

- Modificabilidad: Una ERS es modificable si cualquier cambio puede realizarse de manera fácil, completa y consistente. Para ello es deseable:
 - Tener una organización coherente y fácil de usar en la que aparezca el índice o una tabla de contenidos fácilmente accesible.
 - Evitar la redundancia, es decir que no aparezca un mismo requisito en más de un lugar de la ERS. No es un error, pero si se tiene que modificar alguna cosa será mucho más cómodo si no tenemos que buscar el mismo requisito en varios lugares.



Características de una buena ERS

- Trazabilidad: Una ERS es explorable o trazable si:
 - El origen de cada requerimiento es claro tanto hacia atrás (origen que puede ser un documento, una persona etc.) como hacia delante (componentes del sistema que realizan dicho requisito).
 - Cuando un requisito de la ERS representa un desglose o una derivación de otro requisito, se debe facilitar tanto las referencias hacia atrás como hacia adelante en el ciclo de vida.
 - Las referencias hacia delante de la ERS son especialmente importantes para el mantenimiento del software. Cuando el código y los documentos son modificados, es esencial poder comparar el conjunto total de requisitos que puedan verse afectados por estas modificaciones



Características de una buena ERS

- Utilizable durante las tareas de mantenimiento y uso: En la ERS también se deben tener en cuenta las necesidades de mantenimiento.
 - El personal que no ha intervenido directamente en el desarrollo debe ser capaz de encargarse de su mantenimiento.
 - Actúa a modo de plano de la aplicación, permitiendo incluso modificaciones que no requieran un cambio en el diseño.
 - En ocasiones, el equipo de desarrollo supone unos conocimientos que el personal que se encargue del mantenimiento no tiene por qué tener. Por esta razón es necesaria una correcta documentación de las funciones, ya que si no se conoce en detalle su origen, difícilmente podrán ser modificadas.