Tema 2: Conceptos Básicos de la Ingeniería de Requisitos

Ingeniería de Requisitos

Raquel Martínez España

Grado en Ingeniería Informática

Raquel Martínez España- Tlf: (+34) 968 27 86 57

Objetivos

Este tema tiene como objetivos principales:

- Presentar el concepto de requisito de un sistema software y explicar las diferentes formas de expresar requisitos.
- Establecer los conceptos de requisitos del usuario y del sistema y el por qué se deben escribir de diferentes formas.
- Diferenciar entre requisitos funcionales y no funcionales.

Concepto de Requisito

Definiciones

Antes de profundizar en los temas que se tratan en este capítulo, es preciso definir formalmente lo que se entiende por “requisito software'".

Según el glosario de términos estándar del IEEE podemos definir el concepto de “requisito” desde dos enfoques: cómo condición o capacidad y cómo su representación. Así tendríamos las siguientes definiciones:

- *“Una condición o capacidad que debe ser cumplida, o poseída, por un sistema o componente de sistema, para satisfacer un contrato, estándar, especificación u otros documentos impuestos formalmente”*
- *“Una representación documentada de una condición o capacidad relativa a los puntos anteriores”*

De acuerdo a la guía SWEBOK 2004, un requisito software es la propiedad que un software desarrollado o adaptado debe tener para resolver un problema concreto.

Otra conocida definición es la aportada por Sommerville y Sawler en 1997:

“Una especificación de qué se debería implementar. Son descripciones de cómo se debe comportar el sistema, o de un atributo o propiedad del sistema. Puede ser una restricción en el proceso de desarrollo de un sistema.”

Partiendo de esta definición, podemos dar una definición de la Ingeniería de Requisitos basada en el concepto de requisito:

> *La Ingeniería de Requisitos comprende las actividades de desarrollo de software (y SI) relacionadas con la gestión y definición de requisitos para sistemas nuevos o actuales.*

Raquel Martínez España- Tlf: (+34) 968 27 86 57

De esta forma, el documento en el cual se especifican formalmente las características deseadas que el sistema software debe cumplir, los requisitos, se denomina Documento de Especificación de Requisitos Software, ERS, o SRS de sus siglas en inglés Software Requirements Specification.

En general, los requisitos especifican qué se supone que debe hacer un sistema, pero normalmente no intentan expresar cómo lograr estas funciones.

Por ejemplo, el siguiente podría ser un requisito para una aplicación de contabilidad:

- *R1) "El sistema debe permitir al usuario ver su saldo"*

Sin embargo, por normal general, R2 no será un requisito:

- *R2) "El estado de la cuenta del cliente se almacenará en una tabla llamada saldoCliente en una base de datos MySQL"*

Podría darse el caso de un sistema en el que por razones de compatibilidad con otros sistemas ya existentes tuviera que existir una tabla en MySQL con ese nombre. En ese caso, R2 sí sería un requisito válido de la aplicación.

Así, por normal general, los requisitos no pretenden dar detalles de diseño, implementación o pruebas, ni información relativa a la planificación o necesidades del proyecto. Estos elementos suelen llamarse "especificaciones suplementarias".

Por ejemplo, normalmente, las siguientes afirmaciones no serán requisitos:

- Detalles de diseño: "La aplicación deberá diseñarse usando un patrón Modelo-Vista-Controlador"
- Detalles de implementación: "Los datos se almacenarán en una BBDD Oracle"
- Detalles de las pruebas: "Las pruebas se realizarán utilizando jUnit"
- Planificación del proyecto: "La primera fase deberá estar entregada el 10 de Octubre"
- Necesidades del proyecto: "El presupuesto ascenderá a 10.000€"

Objetivos de los Requisitos

Los requisitos no sólo nos permiten determinar qué debe hacer el sistema, si no que pretenden conseguir otros beneficios. De acuerdo con J. W. Bracket (Software Requirements, 1990), los requisitos tienen como objetivos los siguientes puntos:

- Alcanzar un acuerdo entre clientes, productores de sw y usuarios, sobre lo que hay que producir.

Raquel Martínez España- Tlf: (+34) 968 27 86 57

- Proporcionar la base para el diseño de software.
- Servir como soporte para la verificación y validación de los productos obtenidos.
- Orientar a potenciales compradores de sw (clientes) sobre la definición de nuestros productos.

Características de los Requisitos

Existe numerosa bibliografía acerca de las propiedades deseables de los requisitos. Según el estándar IEEE-830, los requisitos deben ser:

- Correctos: Tanto el cliente como el desarrollador deben revisarlos para asegurar que no tienen errores.
- Consistentes: Dos requisitos son inconsistentes cuando es imposible satisfacerlos simultáneamente.
- Completos: El conjunto de requisitos está completo si todos los estados posibles, cambios de estado, entradas, productos y restricciones están descritos en alguno de los requisitos.
- Realistas: Todos los requisitos deben ser revisados para asegurar que son posibles.
- Necesarios: Los requisitos deben ser revisados para conservar sólo aquellos que inciden directamente en la resolución del problema del cliente.
- Verificables: Se deben poder preparar pruebas que demuestren que se han cumplido los requisitos.
- Rastreables: ¿Se puede rastrear cada función del sistema hasta el conjunto de requisitos que la establece?

Problemas comunes de los Requisitos

Los problemas más comunes en los requisitos son: ambigüedad, imprecisión, inexactitud e inconsistencia (D. A. Stokes, Requirements analysis, 1990).

Las dificultades de comunicación o malentendidos entre desarrolladores y usuarios, y entre los propios miembros del equipo de desarrolladores, pueden provocar que los requisitos no reflejen las necesidades reales del cliente, o que sean inconsistentes o incompletos.

En muchas ocasiones, resulta complicado obtener información de algunos usuarios, pues muchos no saben (o no desean) revelar todos los detalles sobre el funcionamiento de un sistema existente. En otras ocasiones la persona que obtiene los requisitos no tiene la formación ni la experiencia necesaria para ello, o desconoce el dominio del problema al que se enfrenta y no comprende la jerga en que se expresan los usuarios, etc. Esto puede provocar problemas de imprecisión de los requisitos. Un requisito ambiguo puede ser interpretado de diferentes formas por desarrolladores y usuarios. Por ejemplo, imaginemos el siguiente requisito:

Raquel Martínez España- Tlf: (+34) 968 27 86 57

R1: "El valor de precisión será alto"

Este requisito puede ser interpretado de distintas formas dependiendo de lo que cada persona considere como "alto" en el contexto y situación del sistema.

Además, la complejidad inherente al software, es otro de los motivos fundamentales que dan lugar a generar requisitos inconsistentes y/o incompletos. La determinación exacta de los requisitos de un sistema que se adivina complejo no es, en la mayoría de los casos, tarea sencilla y de hecho, en la práctica este objetivo es imposible de cumplir al 100% en un documento de requisitos de complejidad media o grande.

Otro aspecto importante es el relacionado con la naturaleza cambiante de los requisitos. La aparición de nuevos requisitos durante el proceso de desarrollo, o el hecho de que muchos de ellos se vean alterados a lo largo del mismo puede ocasionar problemas en el proceso de ingeniería de requisitos. Más adelante estudiaremos como paliar este problema con una adecuada gestión de los requisitos.

Por último, y como veíamos al estudiar la definición de requisito, un requisito debe ser independiente del diseño y la implementación. Sin embargo, que sólo se admitan requisitos que limiten esta frontera no siempre es posible, por ejemplo, es posible que la arquitectura del sistema pueda ser diseñada para estructurar los requisitos, o que el sistema deba interaccionar con otros sistemas que generan requisitos de diseño del software o incluso que el uso de un diseño específico sea un requisito de dominio.

Tipos de Requisitos

Existen diversas taxonomías para clasificar los requisitos de acuerdo a distintos criterios. Una de las formas más habituales de clasificación es la establecida por Sommerville, según la cual los requisitos se clasifican en funcionales y no funcionales, o como requerimientos del dominio.

- Continuar lectura en Sommerville: pp. 109 – 116

La anterior clasificación atiende principalmente a la naturaleza de la característica del sistema deseada que especifica.

Sin embargo, de acuerdo al nivel de abstracción o detalle podemos clasificar los requisitos en requisitos de usuario y requisitos de sistema y del software.

- Continuar lectura en Sommerville: pp. 116 – 123

Otra clasificación bastante utilizada es aquella definida por Rombach (H. D. Rombach, Software Specifications: A Framework, 1990) y Brackett (J. W. Brackett. Software Requirements, 1990), en la que se establece la clasificación en base a la audiencia a la que está dirigido el requisito, es decir,

Raquel Martínez España- Tlf: (+34) 968 27 86 57

las personas que deben ser capaces de entenderlo. En general, se pueden distinguir dos tipos de audiencia, los clientes y usuarios, que no tienen por qué tener formación en ingeniería del software, y los desarrolladores de software.

De esta forma, el análisis de requisitos se divide en dos niveles. El primer nivel, “Requisitos de Cliente” o “Requisitos C”, documenta los deseos y necesidades del cliente y se expresa en lenguaje comprensible por él. El segundo nivel, “Requisitos de Desarrollador” o “Requisitos D” tienen como audiencia principal la comunidad del desarrollador, tienen una forma más estructurada y específica y un carácter mucho más técnico. En el caso de que la audiencia prevista esté formada por desarrolladores de software, los requisitos suelen expresarse mediante un modelo, normalmente utilizando técnicas estructuradas, orientadas a objetos o formales.

- Requisitos C: Continuar lectura en Braude: pp. 137– 141
- Requisitos D: Continuar lectura en Braude: pp. 182 – 187 y 200 – 205

Raquel Martínez España- Tlf: (+34) 968 27 86 57

Puntos clave

En este apartado se muestran aquellos puntos más importantes que el alumno debería tener claros al finalizar este capítulo:

- Concepto de requisito.
- Tipos de requisitos:

- Requisitos funcionales y no funcionales
    - Los requisitos funcionales: son declaraciones de los servicios que el sistema debe proporcionar o son descripciones de cómo se deben llevar a cabo algunos cálculos.
    - Los requisitos no funcionales restringen el sistema en desarrollo y el proceso de desarrollo que se debe utilizar.
        - Producto, organizaciones o externo.
        - Propiedades emergentes del sistema
    - Los requisites del dominio: son requisitos funcionales que se derivan de las características del dominio de aplicación.

- Requisitos de Usuario y Sistema
    - Los requisitos del usuario:
        - Personas relacionadas con la utilización y obtención del sistema.
        - Lenguaje natural, tablas y diagramas fáciles de entender.
    - Los requisitos del sistema:
        - Funciones que debe proporcionar el sistema.
        - Especificación precisa.
        - Reducir ambigüedad: formulario estructurado del lenguaje natural complementado con tablas y modelos del sistema.

- Requisitos Cliente y Desarrollador
    - Requisitos Cliente (C):
        - Principalmente para el cliente
        - Especificación clara: lenguaje natural, casos de uso, interfaces
    - Requisitos Desarrollador (D):
        - Orientados al Diseñador/ Desarrollador
        - También para el cliente
        - Detallan requisitos C

- Diferentes organizaciones según su uso.