



Programación Avanzada

Requerimientos de Software

[Contenido]

- n Especificación de Requerimientos
- n Tipos de Requerimientos
- n Requerimientos Funcionales
- n Casos de Uso

[Especificación de Requerimientos]

- n La Especificación de Requerimientos es un insumo fundamental en el desarrollo de software:
 - i Es la principal fuente de información a partir de la cual se diseña, implementa y prueba el sistema
 - i Es uno de los aspectos más delicados de un proyecto:
 - n Es algo complejo de obtener
 - n De su correctitud depende el éxito del proyecto

[Especificación de Req. (2)]

- n Representa un “contrato” con el usuario
- n No se genera por completo al inicio del proyecto, sino incrementalmente
- n Suele presentarse como la agregación de diferentes artefactos

[Tipos de Requerimientos]

- n Un requerimiento es una condición o capacidad que un sistema debe cumplir
- n Requerimiento **No Funcional**:
 - i Expresa una propiedad o cualidad que el sistema debe presentar
 - i También restricciones físicas sobre los funcionales
- n Requerimiento **Funcional**:
 - i Expresa una acción que debe ser capaz de realizar el sistema
 - i Especifica comportamiento de entrada/salida

[Requerimientos No Funcionales]

- n Los requerimientos no funcionales suelen referir a:
 - i **Usabilidad:** factores humanos, ayuda, documentación
 - i **Confiabilidad:** frecuencia de fallas, tiempo de recuperación
 - i **Performance:** tiempo de respuesta, tasa de procesamiento, precisión, capacidad de carga
 - i **Soportabilidad:** adaptabilidad, mantenibilidad, configurabilidad, internacionalización

[Requerimientos No Funcionales]

n y más:

- i **Interfaces:** restricciones en la comunicación con sistemas externos
- i **Restricciones:** en el uso de
 - n Sistemas o paquetes existentes
 - n Plataformas
 - n Lenguajes de programación
 - n Ambientes de desarrollo
 - n Herramientas (sistemas de bases de datos, middleware, etc.)

[Requerimientos Funcionales]

- n Los requerimientos funcionales se expresaban en términos de “funciones del sistema”
- n Una función del sistema es algo puntual que el sistema debe hacer
- n Técnica básica: Si X es una función del sistema, entonces la frase “*El sistema debe hacer X*” tiene que tener sentido

[Requerimientos Funcionales (2)]

- n Esta herramienta funcionó bien mientras:
 - i Las funcionalidades de los sistemas eran sencillas
 - i El modelo de uso de los usuarios se limitaba a una interacción simple y puntual con el sistema
- n Perdió su efectividad cuando las interacciones a soportar se volvieron más complejas tomando períodos de tiempo más prolongados
- n Nuevo enfoque: **Casos de Uso**

[Casos de Uso]

- n El enfoque de casos de uso está basado en la noción de **actor**
- n Un actor es un agente externo (humano o no) que interactúa directamente con el sistema
- n Un caso de uso narra la historia completa (junto a todas sus variantes) de un conjunto de actores mientras usan el sistema:
 - i La historia termina cuando uno de los actores (el principal) logra su objetivo y obtiene un resultado de valor

[Casos de Uso (2)]

- n Los casos de uso son la herramienta más aplicada para la especificación de requerimientos funcionales
- n Por ser expresados textualmente resultan simples de comprender (hasta para personal no-técnico)
- n Por estar orientados a los objetivos de los actores (y al camino hacia su obtención):
 - i Son intuitivos
 - i Propician la completitud de especificación

[Casos de Uso (3)]

- n Un caso de uso se compone de:
 - i **Nombre** que identifica al caso de uso
 - i **Actores** participantes en el caso de uso
 - i **Sinopsis** que describe brevemente su objetivo
 - i **Curso típico de eventos** que narra la “historia” más común de los actores durante el uso del sistema
 - i **Cursos alternativos de eventos** que narran las variantes de uso del sistema

[Casos de Uso (4)]

- n Los casos de uso no suelen especificarse con todo detalle de una sola vez
- n Esto se realiza en forma gradual y posterior a la identificación de actores:
 - i Los actores son más fáciles de identificar y sus necesidades son las que dan lugar a los casos de uso
- n Formas posibles de un caso de uso:
 - i Identificado o detectado: solo su nombre y actores participantes
 - i Especificado en alto nivel: se incorpora una sinopsis
 - i Especificado en forma expandida: se incorpora la “historia” de uso y sus variantes

[Casos de Uso (5)]

n Método básico (variable según el avance):

1. Detectar actores
2. Identificar algunos casos de uso
(detectando objetivos y necesidades de actores)
3. Especificarlos en alto nivel
4. Examinarlos y expandir algunos de ellos
5. ...

n A medida que se avanza en el desarrollo se detectan nuevos casos de uso y se especifican otros ya detectados

[Casos de Uso (6)]

- n Sobre el principio del proyecto se tiende a buscar y especificar los casos de uso más importantes
- n Los casos de uso se usan además como criterio de partición del problema en un proceso iterativo e incremental:
 - i En una iteración se desarrolla “uno a la vez”
 - i Los incrementos no refieren a “partes” físicas sino a conjuntos de funcionalidades

[Ejemplo (Formato Expandido)]

Caso de Uso: Realizar una compra

Actores: Cajero

Sinopsis: Un cliente llega a la caja con artículos para comprar. El cajero registra los artículos y recibe el pago. Al finalizar, el Cliente se retira con los artículos.

Escenario Típico:

1. El Cliente llega a la caja con artículos para comprar.
2. El Cajero comienza una nueva venta.
3. El Cajero ingresa el identificador del artículo.
4. El Sistema registra el artículo y presenta su descripción, precio y subtotal.
El Cajero repite los pasos 3 y 4 hasta terminar los artículos.
5. El Sistema presenta el total con los impuestos incluidos.
6. ...

[¿Qué sigue después?]

- n Una vez detectado y especificado el conjunto inicial de casos de uso:

El equipo de desarrollo está listo para analizarlos, diseñar una solución para ellos e implementarlos

- n Mientras tanto el equipo de analistas avanza en la detección y especificación de otros casos de uso