

Modelo de Requisitos

Definición

- Conjunto de requisitos
 - ◆ Especificación de lo **que** se debería implementar
 - **No cómo** se debería implementar
- Tipos
 - ◆ Requisitos funcionales
 - Comportamiento que el sistema debería ofrecer
 - ◆ Requisitos no funcionales
 - Propiedad específica o restricción del sistema
- Creación
 - ◆ Texto o herramienta de ingeniería de requisitos

Requisitos bien formados

- ¿Cómo listar los requisitos?
 - ◆ Identificador único (<id>)
 - Letras que indiquen si es funcional o no
 - Número
 - ◆ Nombre del sistema (<sistema>)
 - ◆ Palabra clase (*debería*)
 - ◆ Declaración de función <funcionar>

<id> El <sistema> debería <funcionar>

Requisitos funcionales

■ Definición

- ◆ Afirmación sobre lo que debería hacer el sistema

■ Ejemplos

- ◆ **RF1** El sistema del cajero *debería* comprobar la validez de la tarjeta insertada
- ◆ **RF2** El sistema del cajero *debería* validar el número PIN facilitado por el cliente
- ◆ **RF3** El sistema del cajero *debería* dispensar no más de 250 euros para una misma tarjeta en un periodo de 24 horas

Requisitos no funcionales

■ Definición

- ◆ Especifican o restringen como se debería implementar el sistema

■ Ejemplos

- ◆ **RNF1** El sistema del cajero *debería* ser escrito en C++
- ◆ **RNF2** El sistema del cajero *debería* comunicarse con el banco utilizando un cifrado de 256 bits
- ◆ **RNF3** El sistema del cajero *debería* validar una tarjeta en tres segundos o menos
- ◆ **RNF4** El sistema del cajero *debería* validar un PIN en tres segundos o menos.

Organizar los requisitos

■ Objetivo

- ◆ Trabajar con un conjunto amplio de requisitos en dominios más pequeños y manejables.

■ Usar taxonomías

- ◆ Jerarquía de tipos de requisitos
 - Depende del tipo de sistema
- ◆ Funcionales no funcionales
 - La más sencilla
- ◆ Usar dos o tres niveles es suficiente

Organizar los requisitos

■ Ejemplo

- ◆ Requisitos funcionales
 - Clientes
 - Productos
 - Pedidos
 - Pago
- ◆ Requisitos no funcionales
 - Rendimiento
 - Capacidad
 - Disponibilidad
 - Seguimiento de estándares
 - Seguridad

Atributos de los requisitos

- Objetivo
 - ◆ Capturar información adicional sobre los requisitos
- ¿Cómo se define?
 - ◆ Nombre descriptivo
 - FechaVencimiento,
 - ◆ Valor
 - Fecha en la que debe estar listo un requisito
- ¿Cuáles? ¿Cuántos?
 - ◆ Dependerá de la naturaleza y necesidades del proyecto, así como de los tipos de requisitos

Atributos de los requisitos

■ Atributo prioridad

- ◆ Prioridad del requisito relativa al resto de requisitos
- ◆ Esquema MoSoCoW
 - **Must have**
 - Requisito fundamental para el sistema
 - **Should have**
 - Requisito importante que se puede omitir
 - **Could have**
 - Requisitos que son opcionales (se realizan si hay tiempo)
 - **Want to have**
 - Requisitos que pueden esperar para versiones posteriores

Encontrar requisitos

- ¿Dónde? → **contexto**
 - ◆ Usuarios directos del sistema
 - ◆ Otros grupos de decisión
 - ◆ Otros sistemas con los que interactúa el sistema
 - ◆ Dispositivos hardware con los que interactúa
 - ◆ Restricciones legales y regulatorias
 - ◆ Restricciones técnicas
 - ◆ Objetivos de negocio

Encontrar requisitos

■ Técnicas

◆ Entrevistas

- Forma más directa de recopilar la información
- Paciencia, saber escuchar, preguntas abiertas, no ideas preconcebidas.

◆ Cuestionarios

- Suplemento a las entrevistas
- Conseguir respuestas a preguntar cerradas

◆ Workshop de requisitos

- Forma más eficiente de capturar requisitos
- Tormenta de ideas

Encontrar requisitos

■ Cuidado con

◆ Eliminación

- La información se filtra
- *“Utilizan el sistema para sacar libros”*

◆ Distorsión

- La información se modifica
- *“Los usuarios no pueden sacar otro libro hasta que todos los libros retrasados han sido devueltos”*

◆ Generalización

- La información se resume en reglas
- *“Todo el mundo debe tener una tarjeta para sacar libros”*