Contents

Módulo 3: Requisitos de Calidad del Software - Clase 2: Tipos de Requisitos de Calidad

1. Objetivos Específicos de la Clase:

- Identificar y clasificar los diferentes tipos de requisitos de calidad: funcionales, no funcionales, de rendimiento, de seguridad, de usabilidad y de mantenibilidad.
- Comprender las características y la importancia de cada tipo de requisito de calidad.
- Distinguir entre requisitos funcionales y no funcionales, proporcionando ejemplos concretos.
- Aprender a definir requisitos de calidad específicos y medibles para cada tipo.

2. Contenido Teórico Detallado:

Los requisitos de calidad, como vimos en la clase anterior, son fundamentales para el éxito de cualquier proyecto de software. No todos los requisitos son iguales, y clasificarlos nos ayuda a gestionarlos de manera más efectiva. Generalmente, se dividen en dos categorías principales: funcionales y no funcionales.

- **Requisitos Funcionales:** Describen *qué* debe hacer el sistema. Definen las funcionalidades o características específicas que el software debe proporcionar. Se centran en las acciones y resultados que el usuario espera del sistema.
  - **Ejemplos:**
    - Un sistema de comercio electrónico debe permitir a los usuarios registrarse y crear una cuenta.
    - Un sistema de gestión de inventario debe permitir añadir, modificar y eliminar productos.
    - Un sistema de biblioteca debe permitir a los usuarios buscar libros por título, autor o ISBN.
  - **Requisitos No Funcionales:** Describen *cómo* debe comportarse el sistema. Se centran en las cualidades o atributos del software, como el rendimiento, la seguridad, la usabilidad, la confiabilidad, la escalabilidad y la mantenibilidad. A menudo, los requisitos no funcionales imponen restricciones sobre el diseño del sistema.
  - **Tipos de Requisitos No Funcionales:**
    - **Rendimiento:** Se refiere a la velocidad, eficiencia y capacidad de respuesta del sistema.
      - *Ejemplo:* El sistema debe responder a las solicitudes de búsqueda en menos de 3 segundos. El sistema debe soportar 1000 usuarios concurrentes.
    - **Seguridad:** Se refiere a la protección de datos y la prevención de accesos no autorizados.
      - *Ejemplo:* El sistema debe requerir autenticación con contraseña para acceder a la información personal. Los datos sensibles deben ser encriptados durante el almacenamiento y la transmisión.
    - **Usabilidad:** Se refiere a la facilidad de uso y la experiencia del usuario.
      - *Ejemplo:* La interfaz de usuario debe ser intuitiva y fácil de aprender. Todas las funciones importantes deben ser accesibles con un máximo de tres clics.
    - **Confiabilidad:** Se refiere a la probabilidad de que el sistema funcione correctamente durante un período de tiempo específico.
      - *Ejemplo:* El sistema debe tener una disponibilidad del 99.9%. El tiempo medio entre fallos (MTBF) debe ser de al menos 6 meses.
    - **Escalabilidad:** Se refiere a la capacidad del sistema para manejar un aumento en la carga de trabajo o el número de usuarios.
      - *Ejemplo:* El sistema debe poder escalar para soportar un aumento del 50% en el número de usuarios sin afectar significativamente el rendimiento.
    - **Mantenibilidad:** Se refiere a la facilidad con la que el sistema puede ser modificado, corregido o adaptado a nuevos requisitos.
      - *Ejemplo:* El código del sistema debe estar bien documentado y organizado para facilitar las modificaciones futuras. El sistema debe ser modular para facilitar la adición de nuevas funcionalidades.

Es crucial entender que la priorización y definición clara de estos requisitos son vitales. Un conflicto común es entre rendimiento y seguridad: medidas de seguridad más robustas a menudo implican una disminución en el rendimiento del sistema. La clave está en encontrar el equilibrio adecuado para cada aplicación.

3. Ejemplos o Casos de Estudio:

- **Caso de Estudio: Desarrollo de una Aplicación de Banca Móvil:**
  - **Requisitos Funcionales:**
    - Permitir a los usuarios ver el saldo de sus cuentas.
    - Permitir a los usuarios transferir fondos entre cuentas.
    - Permitir a los usuarios pagar facturas.
    - Permitir a los usuarios depositar cheques mediante una foto.
  - **Requisitos No Funcionales:**
    - **Seguridad:** La aplicación debe utilizar encriptación de extremo a extremo para proteger las transacciones. Debe requerir autenticación de dos factores.
    - **Rendimiento:** La aplicación debe cargar el saldo de la cuenta en menos de 2 segundos.
    - **Usabilidad:** La interfaz debe ser fácil de navegar y entender para usuarios de todas las edades.
    - **Disponibilidad:** La aplicación debe estar disponible 24/7, excepto por breves períodos de mantenimiento programado.

4. Problemas Prácticos o Ejercicios con Soluciones:

- **Ejercicio 1:** Para un sistema de gestión de un hospital, identifique al menos tres requisitos funcionales y tres requisitos no funcionales (uno de rendimiento, uno de seguridad y uno de usabilidad).
  - **Solución:**
    - **Funcionales:**
      - El sistema debe permitir registrar nuevos pacientes.
      - El sistema debe permitir programar citas médicas.
      - El sistema debe permitir generar informes de actividad.
    - **No Funcionales:**
      - *Rendimiento:* El sistema debe permitir acceder a la información del paciente en menos de 1 segundo.
      - *Seguridad:* El acceso a los registros de los pacientes debe estar restringido a personal autorizado.
      - *Usabilidad:* La interfaz del sistema debe ser compatible con pantallas táctiles.
- **Ejercicio 2:** Clasifique los siguientes requisitos en funcionales o no funcionales:
  - El sistema debe enviar un correo electrónico de confirmación después de cada compra.
  - El sistema debe ser compatible con los navegadores Chrome, Firefox y Safari.
  - El sistema debe realizar una copia de seguridad de la base de datos diariamente.
  - El sistema debe generar un informe mensual de ventas.
  - **Solución:**
    - Funcionales: El sistema debe enviar un correo electrónico de confirmación después de cada compra. El sistema debe generar un informe mensual de ventas.
    - No Funcionales: El sistema debe ser compatible con los navegadores Chrome, Firefox y Safari. El sistema debe realizar una copia de seguridad de la base de datos diariamente.

5. Materiales Complementarios Recomendados:

- **Libros:**
  - "Software Engineering" de Ian Sommerville (capítulos sobre requisitos).

  - ”Requirements Engineering: From System Goals to UML Models to Software Specifications” de Axel van Lamsweerde.
- **Artículos:** Buscar artículos académicos sobre ”requisitos no funcionales” y ”clasificación de requisitos de software” en bases de datos como IEEE Xplore o ACM Digital Library.
- **Recursos en Línea:**
  - Tutoriales de ingeniería de requisitos en YouTube.
  - Artículos de blog sobre mejores prácticas en la definición de requisitos de software.