

Requerimientos y Prototipado

Ingeniería de Software y Negocios Digitales

Aug 2024

Tipos de requerimientos

Importancia de los requerimientos en el ciclo de vida del software

- **Los requerimientos definen qué debe hacer el sistema, cómo debe comportarse, y las restricciones bajo las cuales operará. Un buen análisis de requerimientos es clave para evitar malentendidos, sobrecostos y retrasos en el proyecto.**
- “Permitir que los usuarios puedan actualizar su avatar”.
- “Realizar un app para concertar citas”.
- “El website debe soportar 1,000 usuarios concurrentes en hora pico”.

Importancia de los requerimientos en el ciclo de vida del software

- **<https://aws.amazon.com/es/what-is/sdlc/>**



Clasificación General de los Requerimientos

Funcionales: Describen lo que el sistema debe hacer.

No Funcionales: Describen cómo el sistema debe funcionar.

Requerimientos funcionales

Los requerimientos funcionales son las funciones o características que el sistema debe tener para cumplir con su propósito.

Características:

- Específicos y detallados.
- Describen acciones que el sistema debe ser capaz de realizar.

Requerimientos funcionales

Ejemplos:

- "El sistema debe permitir a los usuarios registrarse con una dirección de correo electrónico y una contraseña."
- "El sistema debe generar un reporte mensual de ventas."

Tipos de requerimientos no funcionales

- **Rendimiento:** Tiempo de respuesta, capacidad de procesamiento.
- **Seguridad:** Autenticación, autorización, encriptación.
- **Usabilidad:** Facilidad de uso, accesibilidad.
- **Mantenibilidad:** Facilidad para realizar modificaciones y actualizaciones.
- **Portabilidad:** Capacidad para funcionar en diferentes entornos.

Requerimientos no funcionales

- "El sistema debe poder manejar 1000 usuarios simultáneamente."
- "El sistema debe estar disponible el 99.9% del tiempo."

Impacto en el Proyecto

Un sistema con bajos requerimientos de rendimiento o seguridad puede cumplir sus funciones pero fracasar en la práctica.

Diferencias entre requerimientos funcionales y no funcionales

- **Requerimientos Funcionales** son concretos, describen acciones específicas que el sistema debe realizar.
- **Requerimientos No Funcionales** son más abstractos y describen atributos de calidad que definen cómo esas funciones deben ejecutarse.

Diferencias entre requerimientos funcionales y no funcionales

- **Ejemplos Prácticos**

- Un sistema de banca en línea:
 - Funcional: "El sistema debe permitir transferencias entre cuentas."
 - No Funcional: "La transferencia debe completarse en menos de 2 segundos."

Relación y Complemento:

Los requerimientos no funcionales garantizan que los funcionales se realicen con el nivel de calidad necesario.

Caso de estudio:

**Proyecto de la Aplicación de Rastreo
COVID-19 del NHS (NHS COVID-19
App) en el Reino Unido.**

Caso de estudio

- **Fuente:** <https://www.theguardian.com/world/2020/nov/02/fault-in-nhs-covid-app-meant-thousands-at-risk-did-not-quarantine>
- Contexto del Proyecto: En marzo de 2020, al comienzo de la pandemia de COVID-19, el Servicio Nacional de Salud del Reino Unido (NHS) inició el desarrollo de una aplicación para rastrear contactos de personas contagiadas con COVID-19. La intención era utilizar tecnología de rastreo por Bluetooth para identificar y alertar a personas que habían estado en contacto cercano con alguien que dio positivo por COVID-19.

Problemas de requerimientos

- **Elección de una Arquitectura Centralizada:**

- Inicialmente, los desarrolladores del NHS optaron por una arquitectura centralizada para la aplicación. Esto significaba que los datos de contacto de los usuarios se almacenarían en un servidor central, lo cual generó preocupaciones sobre la privacidad y la seguridad
- Además, la arquitectura centralizada no era compatible con las APIs de rastreo de Apple y Google, lo que resultó en problemas técnicos, como la incapacidad de la aplicación para funcionar correctamente en segundo plano en dispositivos iOS.

Problemas de requerimientos

- **Falta de Enfoque en la Experiencia del Usuario:**

- Los requerimientos de la aplicación no abordaron adecuadamente la experiencia del usuario ni consideraron los diferentes modelos de teléfonos y sistemas operativos que serían utilizados.
- Esto resultó en una baja adopción de la aplicación, ya que muchos usuarios la encontraron difícil de usar o no confiaban en la seguridad de sus datos.

Problemas de requerimientos

- **Cambio de Requerimientos Tardío:**

- Después de varios meses de problemas y críticas, el NHS decidió cambiar a una arquitectura descentralizada que sí era compatible con las APIs de Apple y Google. Este cambio de requerimientos se produjo demasiado tarde en el ciclo de desarrollo, lo que llevó a retrasos significativos en el lanzamiento de una versión funcional de la aplicación.

Consecuencias

- El cambio tardío a una arquitectura descentralizada y la falta de consideración inicial por la privacidad y la experiencia del usuario causaron un retraso en la implementación de una herramienta crucial en la lucha contra la pandemia. La versión inicial de la aplicación fue ampliamente criticada y se consideró un fracaso, lo que llevó al desarrollo de una nueva versión que se lanzó en septiembre de 2020, varios meses después de lo planeado originalmente.

Lecciones aprendidas

- Este caso destaca la importancia de alinear los requerimientos técnicos con las expectativas y necesidades de los usuarios, así como la necesidad de ser flexible pero riguroso en la gestión de cambios en los requerimientos. Un enfoque temprano en la privacidad y la interoperabilidad técnica podría haber evitado muchos de los problemas que enfrentó el proyecto de la aplicación del NHS.

Ejercicio

Tarea en clase: Identificar y especificar requerimientos funcionales y no funcionales para un sistema de reservas en línea para un restaurante.