

Seminario 2. Requisitos no funcionales

Ingeniería de Requisitos – Curso 2023 / 2024

Departamento de Informática

Universidad de Oviedo

¿Qué son los requisitos no funcionales?

Son **restricciones** a las funcionalidades que ofrece el sistema.

Detallan **cómo debe funcionar** un sistema en vez de sus funcionalidades (requisitos funcionales).

¿Por qué son importantes?

Ignorar los requisitos no funcionales en esta fase, supone identificar el error y remediarlo en la fase de implementación, donde será mucho **más costoso**.

Se deben definir en términos **precisos**.

Ejemplo de requisito mal definido

Se **identifica** el siguiente requisito no funcional:

RNF1. El sistema debe tener un tiempo de respuesta aceptable.

Al definirlo de forma **imprecisa**, el sistema tiene un tiempo de respuesta de 15 segundos en la fase de desarrollo, pero el sistema necesitaba que fuera de 5 segundos para que no se formasen colas.

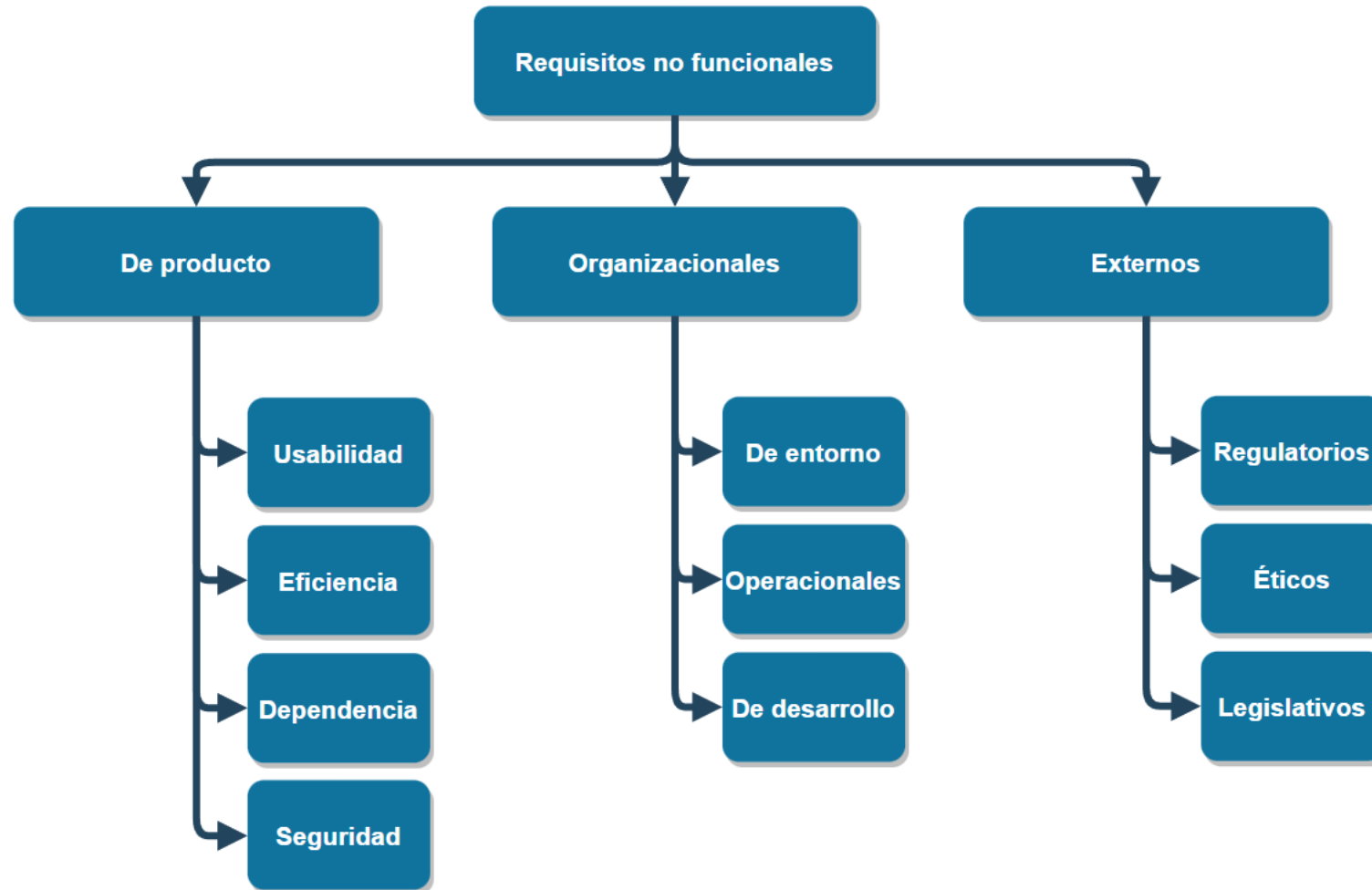
Resultado: El usuario final decide no utilizar el sistema y el proyecto fracasa.

Posible corrección: *RNF1. El sistema debe tener un tiempo de respuesta de un máximo de 5 segundos.*



Brainstorming

Clasificación según Sommerville



Requisitos de producto

Estos requisitos especifican cómo debe comportarse el software.

- Requisitos de usabilidad
- Requisitos de eficiencia
- Requisitos de dependencia (disponibilidad, mantenibilidad, fiabilidad, seguridad industrial, integridad)
- Requisitos de seguridad

Clasificar requisitos de producto (1)

RNF1. La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.

USABILIDAD

RNF2. Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema.

SEGURIDAD

RNF3. El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente accederlo.

DEPENDENCIA

RNF4. El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.

EFICIENCIA

Clasificar requisitos de producto (2)

RNF5. El tiempo para iniciar o reiniciar el sistema no podrá ser mayor a 5 minutos.

DEPENDENCIA

RNF6. La probabilidad de fallo del Sistema no podrá ser mayor a 0,05%.

DEPENDENCIA

RNF7. Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

EFICIENCIA

RNF8. El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.

USABILIDAD

Requisitos organizacionales

Estos requisitos derivan de políticas de la empresa cliente y de la empresa desarrolladora.

- Requisitos de entorno
- Requisitos operacionales
- Requisitos de desarrollo

Requisitos externos

Requisitos derivados de factores externos al sistema, por ejemplo, requisitos necesarios para que el sistema sea aprobado por un organismo regulador.

- Requisitos regulatorios
- Requisitos éticos
- Requisitos legislativos

Clasificar requisitos: Organizacional vs Externos

RNF9. La metodología de desarrollo de software será Behaviour Driven Development (BDD) apoyada en Cucumber.

ORGANIZACIONAL

RNF10. El nuevo sistema y sus procedimientos de mantenimiento de datos deben cumplir con las leyes y reglamentos de protección de datos médicos.

EXTERNO

RNF11. Las pruebas de software se ejecutarán utilizando Selenium y Ruby como herramienta.

ORGANIZACIONAL

Atributos de calidad

- Disponibilidad
- Fiabilidad
- Extensibilidad
- Escalabilidad
- Mantenibilidad
- Seguridad
- Usabilidad

Ejercicio: Identificar requisitos no funcionales

Perteneces al equipo de análisis de requisitos de un sistema de mensajería instantánea. En dicho software, se deben permitir tener chats entre 2 usuarios, chats en grupos y difusiones.

Objetivos de la aplicación:

- Mensajería en tiempo real
- Autenticación y autorización de los usuarios
- Seguridad y privacidad de los datos
- Recuperación de los datos perdidos
- Amigable para los usuarios.

Identifica requisitos no funcionales que creas que debe tener el software descrito.

Bibliografía

Ian Sommerville. 2010. Software Engineering (9th. ed.).
Addison-Wesley Publishing Company, USA.