



REQUERIMIENTOS NO FUNCIONALES

Desarrollo de software I

Descripción breve

El presente documento muestra un resumen del libro requerimientos no funcionales

Martín Josué Andrade Salazar | 219737144
Maestría en ingeniería de software

¿Qué son los requerimientos no funcionales?

En el área de requisitos de software, el término requisitos no funcionales se ha utilizado para referirse a preocupaciones no relacionadas con la funcionalidad del software. Sin embargo, diferentes autores se caracterizan esta diferencia en definiciones informales y desiguales. Por ejemplo:

- a) Describir los aspectos no conductuales de un sistema, capturando las propiedades y restricciones bajo las cuales un sistema debe operar.
- b) Los atributos generales requeridos del sistema, incluyendo portabilidad, confiabilidad, eficiencia, ingeniería humana, capacidad de prueba, comprensibilidad y modificabilidad.
- c) Requisitos que no refieren específicamente a la funcionalidad de un sistema. Imponen restricciones externas que el producto debe cumplir.
- d) Requisitos globales sobre su desarrollo u costo operativo, desempeño, confiabilidad, mantenibilidad, portabilidad, robustez, y similares. No existe una definición formal ni una lista completa de requisitos no funcionales.

Por supuesto, el término requerimientos no funcionales no significa que no son capaces de realizar una función, pero se interpreta fuera de contexto, puede crear confusión.

Esquemas de clasificación

- Requisitos de interfaz: Describir cómo debe interactuar el sistema con su entorno, usuarios y otros sistemas. Por ejemplo, interfaces de usuario y sus cualidades (facilidad de uso).
- Requisitos de desempeño: Describir las **restricciones de desempeño** que involucran límites de tiempo/espacio, como cargas de trabajo, tiempo de respuesta, rendimiento y espacio de almacenamiento disponible. **Confiabilidad** que involucra la disponibilidad de los componentes y la integridad de la información mantenida y suministrada por el sistema. **Capacidad de supervivencia**, como la resistencia del sistema.
- Requisitos operativos: Incluyen limitaciones físicas (tamaño y peso), disponibilidad de personal, consideraciones sobre el nivel de habilidad, accesibilidad del sistema para mantenimiento.
- Requisitos del ciclo de vida: Se pueden clasificar en dos subcategorías. Las cuales son:
 - o Calidad del diseño: Medida en términos tales como mantenibilidad, mejora y portabilidad.
 - o Límites al desarrollo: Como limitaciones de tiempo de desarrollo, disponibilidad de recursos, estándares metodológicos.
- Requisitos económicos: Costos inmediatos y/o a largo plazo.
- Requisitos políticos

Representaciones de requisitos no funcionales

Los requisitos relacionados con los NFR suelen estar separados de los requisitos de funcionalidad. Una forma habitual de representarlos es mediante frases de requisitos, que enumeren por separado en diferentes apartados de la sección de requisitos técnicos. El estándar IEEE 830 (práctica recomendada para especificaciones de requisitos de software) es un buen ejemplo.

Algunos autores proponen una estructura en torno a las oraciones de requisitos, que se compone de:

- Número de identificación
- Tipo de NFR
- Caso de uso relacionado con él
- Descripción
- Justificación
- Originador
- Criterio de ajuste
- Satisfacción del cliente
- Insatisfacción del cliente
- Prioridad
- Conflictos
- Material de apoyo e historia

Todos estos son información textual informal. Una definición de requisitos debe responder a tres tipos de preguntas:

- ¿Por qué se necesita un sistema?
- ¿Qué características del sistema servirán para satisfacer este contexto?
- ¿Cómo se implementa el sistema a construir?

En el marco NFR, cada objetivo blando o contribución está asociado con una etiqueta, que indica el grado en que se cumple o se niega. En el Marco NFR se ofrece un procedimiento de propagación de etiquetas para determinar el efecto de varias decisiones de diseño, independientemente de si son a nivel de sistema o de software. Cuando se utiliza el Marco NFR, estos se publican como objetivos blandos que deben abordarse o lograrse, y se aplica un proceso iterativo y entrelazado para satisfacerlos, a través de descomposiciones, operacionalizaciones y argumentaciones. A lo largo del proceso, se crea y mantiene una representación visual, SIG (gráfico de interdependencia de objetivos blandos), que realiza un seguimiento de los objetivos blandos y sus interdependencias, junto con el impacto de diversas decisiones a través de etiquetas. Cuando las NFR eventualmente se operacionalizan, en términos de operaciones, entidades o restricciones de software, se convierten en la justificación de por qué existen tales operacionalizaciones en el sistema de software, es decir, para servir los atributos de calidad especificados como objetivos blandos de NFR. Si los ingenieros de software son lo suficientemente cuidadosos como para mantener el historial de la construcción de software, entonces podrán explicar y justificar por qué existen tales operacionalizaciones. Dicho de otra manera, se toman diferentes tipos de decisiones de diseño a lo largo de un proceso de desarrollo de software, y los NFR actúan como criterio para dichas decisiones de diseño.

Conclusiones

En resumen, los requisitos no funcionales en el ámbito del desarrollo de software abarcan aspectos no directamente relacionados con la funcionalidad del sistema, capturando propiedades y restricciones que definen su operación. Estos requisitos incluyen atributos generales, como portabilidad, confiabilidad, eficiencia, entre otros. Se clasifican en categorías como interfaz, desempeño, operativos, del ciclo de vida, económicos y políticos.

La representación de requisitos no funcionales se realiza a menudo mediante frases desglosadas en secciones específicas, siguiendo estándares como el IEEE 830. La estructura típica incluye información detallada como identificación, tipo, caso de uso relacionado, descripción, justificación, origen, criterio de ajuste, satisfacción del cliente, insatisfacción del cliente, prioridad, conflictos, y material de apoyo e historia. Se destaca la importancia de abordar los requisitos no funcionales a través de un marco que incluya la propagación de etiquetas para evaluar decisiones de diseño y una representación visual, como el SIG (gráfico de interdependencia de objetivos blandos). Este enfoque proporciona un proceso iterativo para satisfacer los objetivos blandos de los requisitos no funcionales, asegurando que las decisiones de diseño estén alineadas con los atributos de calidad especificados. En última instancia, los requisitos no funcionales actúan como criterio para las decisiones de diseño a lo largo del desarrollo del software, contribuyendo a la justificación y explicación de las operacionalizaciones implementadas.

Referencias

Barbacci, M., Longstaff, TH, Klein, MH, Weinstock, CB: Quality Attributes, Informe técnico CMU/SEI-95-TR-021, ESC-TR-95-021 (diciembre de 1995)

Glinz, M.: Sobre requisitos no funcionales. En: 15ª Conferencia Internacional de Ingeniería de Requisitos del IEEE (RE 2007), págs. 21-26 (2007)

Roman, G.-C.: Una taxonomía de problemas actuales en ingeniería de requisitos. Computadora IEEE, 14-21 (abril de 1985)

Grady, R., Caswell, D.: Métricas de software: establecimiento de un programa para toda la empresa. Prentice-Hall, acantilados de Englewood (1987)