



# Requerimientos

Autor: Luciano Straccia

# AGENDA



- ▶ Concepto y Tipo de Requerimientos
- ▶ Ingeniería de Requerimientos
- ▶ Atributos de calidad



# ▶ Requerimientos

# Requerimientos

- ▶ Es una característica que un sistema debe tener para cubrir alguna de las necesidades que lo motivan
- ▶ Condición o capacidad que debe satisfacer o poseer un sistema o un componente de un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto. (IEEE)

# Requerimiento / Requisito

- ▶ En inglés: acuerdos en el uso de «Requirements» (no se usa Requisite)
- ▶ En español: uso de «requerimientos» como término genérico.
- ▶ Algunos diferencian «requerimiento» como necesidad del cliente y «requisito» como algo que debe cumplir el sistema. Sin embargo las definiciones de IEEE no contemplan esta diferencia.

# Tipos de requerimientos

- ▶ **Requerimientos Funcionales**
- ▶ **Requerimientos No Funcionales (Relación con Atributos de Calidad, Normas ISO 25000)**
- ▶ **Restricciones**

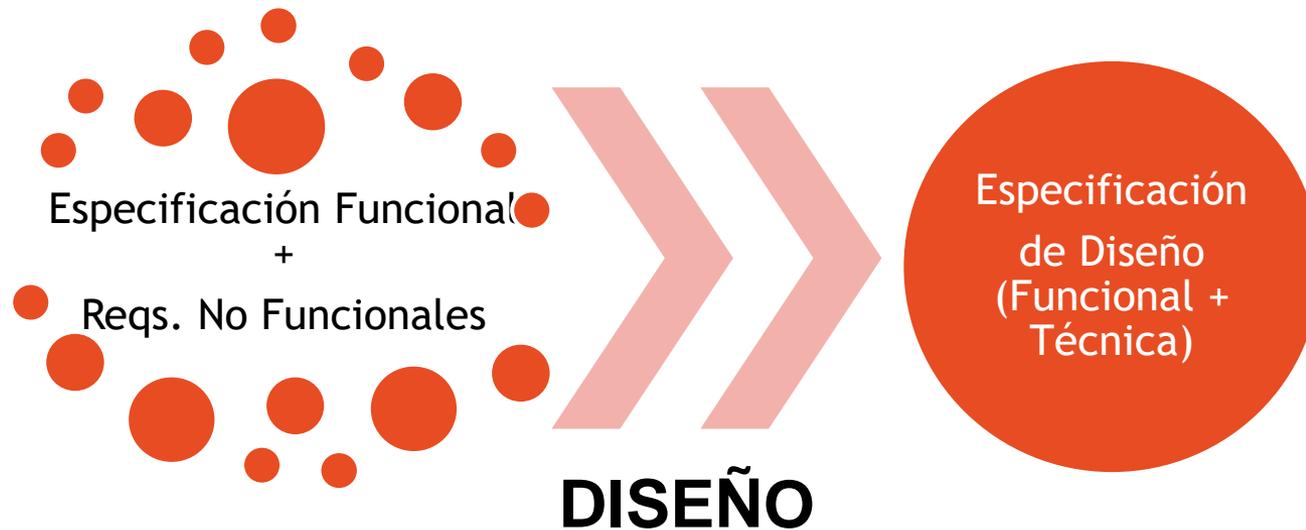
# Cómo documentar según tipos de requerimientos

- ▶ **Requerimientos Funcionales:**
  - ▶ Modelos diversos: Casos de Uso, Estado, Actividades, etc.
- ▶ **Requerimientos No Funcionales:**
  - ▶ Listado de requerimientos organizados preferentemente según los atributos de calidad

# Análisis de sistemas



# Diseño de sistemas



# Atributos de una buena especificación

- ▶ Atributos de una “buena especificación”:
  - ▶ Documentada
  - ▶ Correcta
  - ▶ Consistente
  - ▶ Completa
  - ▶ No ambigua
  - ▶ Verificable y Validable
  - ▶ “Traceable” (rastreadable)



# Ingeniería de

- ▶ **Requerimientos**

# Ingeniería de Requerimientos

- Conjunto de actividades que intentan entender las necesidades de los usuarios y traducirlas en afirmaciones precisas y no ambiguas, que se usarán en el desarrollo del sistema.
- Es el proceso de descubrir, analizar, documentar, y verificar los servicios y las restricciones que conforman los requerimientos del sistema.
- Una forma disciplinada y sistemática de llegar desde las necesidades de los usuarios a la especificación.

# Ingeniería de Requerimientos

- Las actividades básicas son:
  - Extracción (elicitación) de requerimientos
  - Análisis & Especificación de requerimientos
  - Verificación & Validación

# Extracción de Requerimientos

- ▶ Obtener conocimiento relacionado al problema que hay que resolver, con el propósito de producir posteriormente una especificación de requerimientos lo mas rigurosa posible.
- ▶ El conocimiento en el dominio del problema es muy importante para la actividad.
- ▶ Técnicas: brainstorming, entrevistas, cuestionarios, encuestas, análisis de documentación, diagrama causa-efecto

# Análisis y Especificación

- ▶ Dividir el problema con el fin de comprenderlo
- ▶ Utilizar modelos para facilitar la comprensión
- ▶ Modelos: representaciones abstractas de la realidad
- ▶ Modelos: representaciones del problema (y la solución) en un lenguaje de modelado
- ▶ Distintos problemas requieren distintos tipos de modelos

# Verificación y Validación

¿Estamos haciendo lo que decimos que hacemos? (**Verificación**) (asociado al cumplimiento del proceso)

¿Es lo que estamos haciendo es lo adecuada? (**Validación**) (asociado a las necesidades)



# Atributos de

- ▶ Calidad

# Modelos de Calidad

- ▶ Familia de normas ISO 9126
- ▶ Familia de normas ISO 25000

# Atributos de calidad

- ▶ **Funcionalidad:** Capacidad del producto software de proporcionar funciones que ejecuten las necesidades explícitas e implícitas de los usuarios cuando el software es usado bajo condiciones específicas.
- ▶ **Idoneidad:** capacidad del producto software (PS) para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuario especificados.
- ▶ **Complejidad:** grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- ▶ **Corrección:** capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.

# Rendimiento

- ▶ **Rendimiento:** Representa la relación entre el grado de rendimiento del sitio y la cantidad de recursos (tiempo, espacio, entre otros) usados bajo ciertas condiciones.
- ▶ **Comportamiento en el tiempo:** los tiempos de respuesta y procesamiento y los ratios de throughput de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas establecido.
- ▶ **Utilización de recursos:** las cantidades y tipos de recursos utilizados cuando el software lleva a cabo su función bajo condiciones determinadas.

# Usabilidad

- ▶ **Usabilidad:** Capacidad del producto software de ser entendido, aprendido y usado por los usuarios bajo condiciones específicas.
- ▶ **Inteligibilidad:** capacidad del producto que permite al usuario entender si el software es adecuado para sus necesidades.
- ▶ **Aprendizaje:** capacidad del producto que permite al usuario aprender su aplicación.
- ▶ **Operabilidad:** capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.

# Usabilidad (continúa)

- ▶ Protección a errores de usuario: capacidad del sistema para proteger a los usuarios de hacer errores.
- ▶ Atractividad: capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- ▶ Accesibilidad: capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

# Fiabilidad

- ▶ **Fiabilidad:** Capacidad del producto software de mantener un nivel especificado de rendimiento cuando es usado bajo condiciones específicas.
- ▶ **Madurez:** capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- ▶ **Disponibilidad:** capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- ▶ **Tolerancia a fallos:** capacidad del sistema o componente para operar según lo previsto en presencia de fallos hardware o software.
- ▶ **Capacidad de recuperación:** capacidad del producto software para recuperar los datos directamente afectados y reestablecer el estado deseado del sistema en caso de interrupción o fallo.

# Seguridad

- ▶ **Seguridad:** Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.
- ▶ **Confidencialidad:** Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- ▶ **Integridad:** Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- ▶ **No repudio:** Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados posteriormente.
- ▶ **Autenticidad:** Capacidad de demostrar la identidad de un sujeto o un recurso.
- ▶ **Responsabilidad:** Capacidad de rastrear de forma inequívoca las acciones de una entidad.

# Mantenibilidad

- ▶ **Mantenibilidad:** Capacidad del producto software de ser modificado y probado.
- ▶ **Modularidad:** Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- ▶ **Reusabilidad:** Capacidad de un activo que permite que sea utilizado en más de un sistema software o en la construcción de otros activos.

# Mantenibilidad (cont)

- ▶ Analizabilidad: Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del software, diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar.
- ▶ Cambiabilidad: Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- ▶ Capacidad de ser probado: Facilidad con la que se pueden establecer criterios de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

# Portabilidad

- ▶ **Portabilidad:** Capacidad del producto software de ser transferido de un ambiente a otro.
- ▶ **Adaptabilidad:** Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de hardware, software, operacionales o de uso.
- ▶ **Facilidad de instalación:** Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- ▶ **Intercambiabilidad:** Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno.

# Compatibilidad

- ▶ **Compatibilidad:** Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno hardware o software.
- ▶ **Coexistencia:** capacidad del producto para coexistir con otro software independiente, en un entorno común, compartiendo recursos comunes sin detrimento.
- ▶ **Interoperabilidad:** capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

# Atributos de calidad



- Fuente: Certificación de la Mantenibilidad del Producto Software: Un Caso Práctico - Scientific Figure on ResearchGate. Available from: [https://www.researchgate.net/figure/Modelo-de-calidad-del-producto-software-segun-la-ISO-IEC-25010\\_fig1\\_283699208](https://www.researchgate.net/figure/Modelo-de-calidad-del-producto-software-segun-la-ISO-IEC-25010_fig1_283699208) [accessed 7 May, 2021]

# Conflictividad

- ▶ Los atributos de calidad pueden entrar en conflicto
- ▶ Ejemplos de conflictos típicos:
  - ▶ Seguridad vs. Usabilidad
  - ▶ Performance vs. Modificabilidad
  - ▶ Portabilidad vs. Performance
- ▶ Estrategias de resolución:
  - ▶ Priorización / Jerarquización
  - ▶ Balanceo / Good Enough