

Crisis del software: Etapa a principios de los 70 (donde no existía la Ingeniería de Software) donde se notaban las dificultades del desarrollo frente al crecimiento de su demanda, de la complejidad de los problemas a ser resueltos, y de la inexistencia de técnicas para el desarrollo de sistemas validables.

Ingeniería de Software: Establecer o usar principios de ingeniería para obtener software económicamente, que sea fiable y funcione eficientemente.

Actor: Agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema a construir. Son externos al sistema. Se diferencia de un usuario en que un actor es una clase de rol, mientras que el usuario es una persona que cuando usa el sistema asume un rol.

Universo de Información (UdI): Contexto de desarrollo y operación del software. Incluye a los actores.

Stakeholder: Interesado en el cambio a considerar. Son quienes ganan y pierden, y los que necesitan el requerimiento.

Dominios de aplicación del Software:

- Software de sistemas: Programas escritos para dar servicios a otros programas. Pueden procesar estructuras de información o datos indeterminados.
- Software de aplicación: Programas aislados que resuelven una necesidad de negocios específica.
- Software de ingeniería o ciencias: Desde algoritmos numéricos hasta CAD, simulación de sistemas, etc
- Software incrustado: Implementa y controla características y funciones para el usuario final y el sistema en sí
- Software de línea: Destinado a un público amplio.
- Aplicaciones web.
- Inteligencia artificial.

Requerimiento (1): Función o característica de un sistema que es necesaria, conductas cuantificables y verificables que un sistema debe poseer y lo restringen a que deba trabajar para satisfacer los objetivos de una organización y resolver un conjunto de problemas.

Requerimiento (2):

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que debe reunir o debe poseer un sistema o componente del sistema para satisfacer un contrato, norma, especificación u otros documentos
3. Representación documentada de una condición o capacidad como en 1. o 2.

Ingeniería de requerimientos: Proceso sistemático de desarrollar los requerimientos a través de un proceso iterativo de analizar un problema, documentar las observaciones resultantes y verificar la exactitud de la comprensión ganada. Según J. Leite: "Proceso en el que lo que se hará, se elicit y modela, a través de diferentes puntos de vista, usando una combinación de métodos, herramientas y actores. Este proceso tiene un fin que es el Documento de Requerimientos.

Objetivos de la ingeniería de requerimientos:

1. Que los requerimientos se integren sin restar funcionalidad al diseño del sistemas
2. Solucionar dificultades de análisis y validación.
3. Considerar los requerimientos cambiantes con el tiempo.

Etapas de la ingeniería de requerimientos:

1. Elicitación: Se obtienen los requerimientos reales relevando datos con el usuario/cliente a través de equipos multidisciplinarios.
2. Análisis: Permiten al analista representar el UdI usando un lenguaje más técnico y reduciendo ambigüedades. Brinda funciones que facilitan el diseño.
3. Especificación: Permite fijar o establecer una base de requerimientos para que el analista se base en estos en el proceso que sigue.

Clasificación de los requerimientos (1):

- Funcionales: Declaraciones de los servicios a proveer indicando comportamientos (permitidos y no permitidos). Deben tender a estar completos y consistentes. Esto quiere decir que estén todos los servicios definidos y que no deben tener definiciones contradictorias. Debido al permanente cambio de los requerimientos (relacionados a la inconsistencia de los mismos) es imposible de cumplir al 100%.
- No funcionales: Propiedades emergentes del sistema (restricciones de tiempo, proceso de desarrollo, estándares, etc). Se clasifican en:
 - Requerimientos del producto: Referidos al comportamiento del software (rapidez de ejecución, memoria tasa de fallas, probabilidad, usabilidad).
 - Organizacionales: Dependientes de los estándares de procesos a utilizar, requerimientos de implementación (ejemplo, lenguaje) y de entrega (cómo y cuándo se produce).
 - Externos: De interoperabilidad con sistemas ajenos, requerimientos legales y éticos.
- Inversos: Indican cómo el sistema no se debe comportar
- De dominio: Proviene del dominio de aplicación del sistema y reflejan las características de éste mismo. Si no se satisfacen es imposible que el sistema funcione bien.

Técnicas para la recolección de datos:

- Lectura de documentos (para incorporar vocabulario)
- Observación (del ambiente de trabajo)
- Entrevistas (para recoger hechos)
- Cuestionarios
- Análisis de protocolos
- Participación de los actores del UdI en el proceso de recolección (si se incorporan a los interesados,
- hay que verificar que los mismos conozcan el lenguaje de modelado)
- Enfoque antropológico (conociendo lo más posible del sistema integrándose al UdI)
- Reuniones (extensión de las entrevistas)
- Reutilización de hechos elicitados previamente
- Recuperación del Software existente

Técnicas para la elicitación:

LEL: Descripción con notación de hipertexto, donde el analista registra las frases o palabras que parecen tener un significado especial. Se cumplen los principios de circularidad y vocabulario mínimo. Sus entradas están compuestas por:

- **Símbolo:** Nombre o frase que designa al término.
- **Noción:** Definición del símbolo.
- **Impacto:** Cómo influye el símbolo en el sistema. Acciones que ejecuta y/o situaciones derivadas.

Símbolo	Noción	Impacto
<u>Sujeto</u>	<ul style="list-style-type: none">• Quién es el sujeto	<ul style="list-style-type: none">• Cuáles son las acciones que ejecuta
<u>Verbo</u>	<ul style="list-style-type: none">• Quién ejecuta la acción• Cuándo y cuáles son los procedimientos incluidos	<ul style="list-style-type: none">• Identificar las situaciones que impiden la acción• Efectos de la acción en el ambiente
<u>Objeto</u>	<ul style="list-style-type: none">• Definición• Objetos con los que se relaciona	<ul style="list-style-type: none">• Acciones que pueden ser aplicadas sobre el objeto
<u>Situación</u>	<ul style="list-style-type: none">• Significado• Acciones que llevan a esa situación	<ul style="list-style-type: none">• Acciones o situaciones derivadas de esta

Escenarios: Describen la situación externa al sistema de software, ayudan a definir los requerimientos funcionales y no funcionales mediante sus respectivos análisis e implicancias, para luego escribir los requerimientos. Su estructura consiste en:

- **Título:** Frase+Verbo+Predicado (Nombre o subescenario)
- **Objetivo:** Actor+Verbo+Predicado (Finalidad)
- **Contexto:** UG+UT+Precondición (Describe ubicación y condición inicial)
- **Recursos:** Nombre+Restricciones (Elementos necesarios)
- **Actores:** Nombre (Personas que cumplen un rol en el escenario)
- **Episodios:** Lenguaje propio (Acciones que detallan al escenario y describen su comportamiento)
- **Excepciones:** Causa(Sujeto+Verbo+Predicado)+Solución

Objetivos de los escenarios:

- Capturar requerimientos:
 - Evitan abstracciones, las definiciones se orientan a una solución
 - Brindan una visión más amplia del comportamiento del macrosistema
- Proveer un medio de comunicación:
 - Derivan del uso del lenguaje de la aplicación
- Aseguran la comunicación usuario-ingeniero en requerimientos
- Garantizan la comunicación entre el equipo de desarrollo
- Facilitan la validación con el usuario
- Comprometen al usuario en el desarrollo:
 - Validan el LEL
 - Contar con un instrumento de trazabilidad
- Documentar los requerimientos
- Capacitar a los miembros del equipo para comprender la aplicación.
- Promover la evolución de los escenarios a medida que avanza el proceso.
- Generar casos de prueba

Proceso de elaboración de los escenarios:

1. Identificar actores: Se identifican actores (sujetos) principales y secundarios
2. Listar escenarios candidatos: A partir de los actores principales se ven sus impactos y se tienen escenarios potenciales.
3. Describir escenarios candidatos: Armar estructura
4. Ampliar lista de escenarios candidatos: Ahora con los actores secundarios repetir el paso 2
5. Describir escenarios candidatos: Con los actores secundarios repetir el paso 3.
6. Revisión: Detectar duplicaciones, unificar, definir restricciones, detectar excepciones. Si dos escenarios tienen episodios comunes, mismo objetivo y mismo contexto, se unifican.
7. Validación: Con los clientes, se valida y corrige. Se relee el contexto en función de las correcciones hechas.

Casos de uso: Desarrollan el comportamiento del sistema. Secuencia de interacciones entre el sistema y alguien o algo que usa sus servicios. Los mismos son independientes del método de diseño utilizado y del lenguaje de programación (esto le otorga flexibilidad). Consiste en:

- **Actores**
- **Casos de uso:** Posee las siguientes características:
 - Están expresados desde el punto de vista del actor.
 - Son documentados con texto informal.
 - Describen lo que hacen tanto el actor como el sistema.
 - Son iniciados por un único actor.
 - Están acotados al uso de una funcionalidad del sistema.

Los mismos se vinculan mediante relaciones de comunicación, asociación, inclusión, extensión y generalización.

Clasificación de los casos de uso:

- **De trazo grueso:** Descripción de todos los casos de uso a través de una descripción superficial, se identifican los casos de uso del sistema. No se detalla la interacción actor-sistema. No se entra en detalle.
- **De trazo fino:** Una vez implementados, se especifican, detallando todo lo que quedó pendiente del trazo grueso (detalles de la forma de la interfaz y sus especificaciones del comportamiento interno).
- **Temporales:** Cuando el caso de uso es provocado por el paso del tiempo

- Primarios: Ejecutados por un actor.
- Secundarios: Ejecutados por otro caso de uso mediante relaciones.

Casos de uso vs. Eventos: Los eventos describen el comportamiento del sistema cuando el mismo ocurre, en cambio los casos de uso describen el dialogo usuario-sistema. Los eventos son “atómicos” (entrada-proceso-salida), en cambio, un caso de uso se puede prolongar más en el tiempo (un caso de uso puede agrupar varios eventos)

Proceso de elaboración de los casos de uso:

1. Identificar actores (¿Para quiénes hago el sistema?)
2. Identificar casos de uso principales para cada actor.
3. Identificar nuevos casos de uso a través de los existentes.

Documento de requerimientos: Declaración oficial de que es lo que requieren los desarrolladores del sistema. Sus destinatarios son:

- Clientes: Especifican, verifican y cambian los requerimientos
- Administradores: Planean el proceso de desarrollo.
- Ingenieros de sistemas: Comprenden para qué se desarrolla.
- Probadores: Desarrollan pruebas de validación
- Mantenedores: Ayudan a comprender el sistema y sus relaciones

Características del documento de requerimientos:

- Contiene sentencias no ambiguas
- Completo
- Verificable
- Posee consistencia
- Es modificable

Clasificación de los requerimientos (2): (en base en el documento de requerimientos)

- De usuario: Declaraciones en lenguaje natural y diagramas de los servicios que se pretende proporcionar y de las restricciones bajo las cuales debe hacerlo. Consideraciones:
 - Al proporcional un lenguaje natural, el documento se puede tornar confuso
 - No se pueden distinguir a veces, los requerimientos funcionales y los no funcionales.
 - Diversos requerimientos distintos se podrían expresar como uno solo
- De sistema: Establecen en detalle funciones, servicios y restricciones operativas del sistema. Su documento de requerimientos debe ser preciso, se debe definir lo que se va a implementar. Consideraciones:
 - Deben describir el comportamiento externo del sistema y sus restricciones operativas. No hacen referencia al diseño o a la implementación. Hay que reducir ambigüedades para la comprensión.

Formas de redactar los requerimientos de sistema:

- Lenguaje natural estructurado: Se guía en formularios o plantillas estándares.
- Lenguaje de descripción de diseño: Similar al de programación pero más abstracto. Útil para especificaciones de interfaces.
- Notaciones gráficas: Lenguaje gráfico+Anotaciones de texto
- Especificaciones matemáticas: Basándose en máquinas de estado finito o conjuntos.

Especificación de la interfaz: Especificar la interfaz favorece a la integración con otros sistemas externos o internos. Tipos:

- De procedimientos: Los sistemas existentes brindan servicios invocables a través de procedimientos de la interfaz (API)
- Estructuras de datos: Migran de un sistema a otro. Los lenguajes facilitan herramientas gráficas de modelado.
- Representación de datos: Establecidas para un subsistema existente. Algunos lenguajes soportan este nivel de especificación.

Proceso de la ingeniería en requerimientos (elaboración del D.R.):

- Estudios de viabilidad: Evalúan si el sistema es útil para el negocio. Comprende la evaluación y recopilación de la información y la redacción de informes. Se realiza un informe del estudio. Se pueden proponer cambios en el alcance, presupuesto, agendas y requerimientos adicionales.
- Obtención y Análisis: Implican el descubrimiento de los requerimientos de los stakeholders. Esto es complicado, ya que:
 1. Los stakeholders no conocen lo que quieren o pretenden demasiado
 2. Los vocabularios entre los grupos de trabajo son diferentes
 3. Varios stakeholders pueden pretender requerimientos diferentes
 4. Factores políticos
 5. Entorno económico

Proceso de obtención: En este proceso se arman los escenarios y casos de uso.

1. Descubrimiento de los requerimientos: Interactuar con el software y el hardware y su documentación
2. Clasificación y organización de los requerimientos.
3. Ordenación en prioridades y negociación
4. Documentación

Luego el proceso de debe repetir cuantas veces sea necesario.

- Validación de requerimientos: Ver si los requerimientos definen lo que el cliente quiere. Se utilizan técnicas como:
 - Revisiones de requerimientos: Se analizan los requerimientos a través de un equipo de revisores
 - Construcción de prototipos: Mediante un modelo ejecutable para ver si cumple con las necesidades
 - Generación de casos de prueba: Se prueban los requerimientos. Estas pruebas determinan la dificultad para implementar los mismos.
- Gestión de requerimientos: Pueden surgir problemas con los usuarios finales:

- Los requerimientos del sistema no son los mismos que los que pretende el usuario.
- Quienes pagan por el sistema generalmente no lo usan, por lo tanto puede haber un conflicto entre requerimientos reales y proyectados.
- Puede cambiar el entorno de negocios y/o técnico del sistema (nuevo hardware, prioridades de negocio, cambios de legislación, etc)

La gestión de requerimientos se encarga de ir actualizándolos a lo largo del tiempo.

Clasificación de los requerimientos (3): Según el tiempo

- Requerimientos duraderos: Son estables, y están relacionados al dominio principalmente.
- Requerimientos volátiles: Cambian durante el proceso de desarrollo o después de la puesta en marcha. Se clasifican en:
 - Cambiantes: Cambian debido al entorno
 - Emergentes: Surgen al incrementarse la comprensión del cliente hacia el sistema.
 - Consecuentes: Son resultado de la introducción del sistema informático ya que cambia la forma de trabajo
 - Compatibilidad: Dependen de sistemas particulares o procesos de negocios en la organización.