



# **Tema 1: Introducción a la Ingeniería de Requisitos**

Ingeniería de Requisitos

Raquel Martínez España

Grado en Ingeniería Informática

<i>Objetivos</i>	3
<i>Concepto de Ingeniería del Software</i>	3
<i>Concepto de Ingeniería de Requisitos</i>	4
<i>Motivos para realizar Ingeniería de Requisitos</i>	5
<i>Factores de calidad del software</i>	7
<i>La IR dentro de la Ingeniería del Software</i>	8
<i>Puntos clave</i>	9

## Objetivos

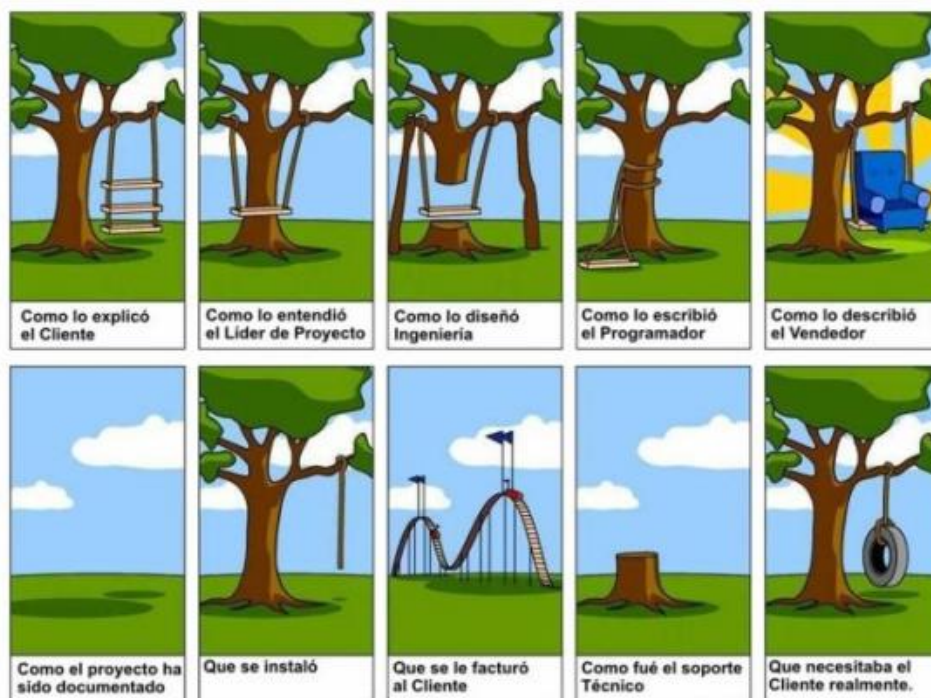
Este tema tiene como objetivos principales:

- Recordar el concepto de Ingeniería del Software.
- Entender el concepto de Ingeniería de Requisitos y su necesidad para el desarrollo de sistemas software de calidad.
- Comprender el papel de la Ingeniería de Requisitos dentro de las fases de la Ingeniería del Software.

## Concepto de Ingeniería del Software

A pesar de que no hay consenso, parece que el origen del término “Ingeniería del Software” viene derivado de dos conferencias organizadas por la OTAN en 1967 y 68. En dichas conferencias se trataba de solventar el problema ya existente en aquella época llamado “crisis del software”.

Un ejemplo de esta llamada crisis del software, es el clásico ejemplo del columpio:



Existen muchas definiciones para el término *Ingeniería del Software*. Entre las más utilizadas podemos nombrar la incluida en el glosario de términos estándar de Ingeniería del Software del IEEE (IEEE 90) (actualización y ampliación del estándar de 1.983):

“La utilización de una aproximación sistemática, disciplinada y cuantificable, al desarrollo, operación y mantenimiento de software; es decir la aplicación de la ingeniería al software”

Otra definición podría ser la establecida por Cota A. en su libro Ingeniería de Software. Soluciones Avanzadas de 1994, según el cual la Ingeniería del Software:

“Permite elaborar consistentemente productos correctos, utilizables y costo-efectivos”

## Concepto de Ingeniería de Requisitos

La construcción de un sistema software, como la mayoría de los productos de ingenierías, comienza con la determinación de lo que se pretende como resultado final del proceso. Esta tarea es más compleja pues la idea completa del artefacto deseado descansa en un conjunto de personas de diferentes formaciones y conocimientos que, en muchos casos, ni siquiera saben claramente lo que necesitan.

En el caso de la ingeniería del software, aquellas actividades relacionadas con la obtención y gestión de los requisitos resultan especialmente críticas. Este conjunto de actividades consiste en obtener y documentar los requisitos para desarrollar el sistema y posteriormente analizarlos con el objetivo de responder a la pregunta sobre qué debe hacerse.

De manera formal, Sommerville (Software Engineering (9th edition), Addison-Wesley, (2011)) define el concepto de **Ingeniería de Requisitos (IR)** como el “*proceso de establecer los servicios que el cliente requiere de un sistema y las restricciones bajo las cuales opera y es desarrollado*”.

En realidad, la obtención de requisitos consiste en capturar el propósito y funcionalidades del sistema desde la perspectiva del usuario. Es por ello, que la IR nos permite establecer un acuerdo o contrato entre desarrolladores, clientes y usuarios sobre las necesidades del sistema que establecerá las bases para el diseño software y el soporte tanto para la verificación y validación del sistema como para su evolución.

Téngase en cuenta, que dado que representa un acuerdo con el usuario para establecer qué quiere, los requisitos deben estar en un lenguaje que sea entendible por el usuario.

De acuerdo con Kotonya y Sommerville (2003), podemos definir el proceso de IR como el conjunto de actividades que son seguidas con el objetivo de descubrir, modelar, validar y mantener un documento de requerimientos. Así podríamos distinguir entre dos partes bien diferenciadas dentro de las actividades de la IR: el proceso de IR y la gestión de esos requisitos.

Mientras que el **proceso de IR** implicaría la *obtención, análisis, especificación y validación del documento de requisitos del sistema*, el proceso de **gestión de requisitos** es el *proceso que comprende y controla los cambios en los requisitos del sistema*, los cuales como veremos más adelante, son inherentes al proceso.

## Motivos para realizar Ingeniería de Requisitos

La importancia de la IR ha sido mencionada y probada en numerosas ocasiones. Diversos autores coinciden en que el éxito de un producto software y su calidad dependen de cómo se hayan realizados las actividades de IR.

Un popular ejemplo de fracaso derivado de un mal desarrollo en el proceso de IR es el supercohete europeo Ariane 5 (1996). En este sistema un sensor mal programado por el equipo de Francia destruyó el supercohete europeo Ariane 5.

El problema que ocurrió fue que se utilizó un diseño realizado para la versión previa del cohete, el Ariane 4, y no se modificó convenientemente. Este error estaba relacionado con la conversión de datos de coma flotante, 64 bits, a valor entero con signo, de 16 bits, el cual era un antiguo requisito del Ariane 4.



En este caso una inadecuada gestión de los requisitos provocó la destrucción de un cohete a los 3 minutos de ser lanzado.

Otro ejemplo similar es el de la nave Mars Polar Lander, en 1999. Diez minutos antes de su aterrizaje previsto en Marte, se perdió el contacto con la nave. Un error de software hizo creer a uno de los brazos de la sonda que ya había tocado suelo, cuando aún se encontraba a 40m de altura.



Más recientemente, en el 2000, el helicóptero MV-22 Osprey del ejército de EE. UU se estrelló provocando desgraciadamente la muerte de 4 soldados. En este caso, el culpable fue un error informático en el sistema de alarmas de fallos hidráulicos.



El error provocó que no se activara la alarma que debía advertir de un incidente en este sistema, y el piloto no pudo obtener información fiable para volar.

Como vemos, un mal desarrollo en las actividades de la IR puede ocasionar grandes pérdidas. De hecho, según estudios realizados por IBM se demuestra que el 80% de todos los defectos en el desarrollo software son insertados en la fase de captura de requisitos. Según algunos autores, ninguna otra parte del desarrollo afecta tanto al sistema resultante si se lleva a cabo de manera incorrecta. Ninguna, de hecho, es más difícil de modificar a posteriori si se hizo mal en un principio.

Otros estudios (Stockman and Norris 1991) sobre la gestión de los proyectos de desarrollo nos dicen que:

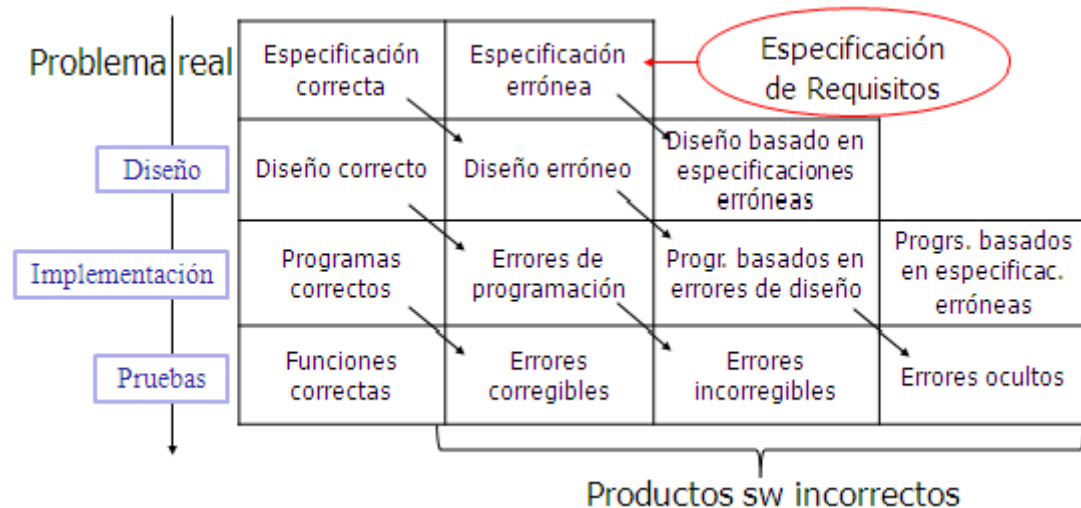
En promedio, los sistemas son entregados un año después de tiempo.

- El 1% de los proyectos termina dentro del tiempo y costes presupuestados.
- Un 25% de los proyectos nunca finalizan.
- El 55% de los fallos se producen en la etapa de análisis y especificación de requisitos.

- El 43% de los errores no son encontrados hasta después de la etapa de pruebas.

Boehm (Boehm 1981) reportó que un 6% del costo y entre un 9 y 12% del tiempo del proyecto son gastados en la fase de requisitos. Y que cuesta entre 5 y 10 veces más reparar los errores en la fase de codificación, y entre 100 y 200 en la fase de mantenimiento que si se repararan en la de requisitos. Similar conclusión obtuvo Davis [Davis 1993].

El siguiente esquema muestra este efecto acumulativo de errores en el desarrollo de software:



Si tenemos en cuenta además, que los procesos/sistemas complejos implican miles de requisitos, para llevar a cabo una correcta gestión y mantenimiento de los requisitos se hace imprescindible una gestión y soporte automatizado de los requisitos.

## Factores de calidad del software

Tal y como se describe en los apartados anteriores, la ingeniería del software busca la producción del software de calidad, y por tanto, establece unas técnicas para conseguir dicha calidad. Sin embargo, cabe preguntarnos primero a qué llamamos calidad del software.

Podemos decir que la calidad del software es una combinación de una serie de factores (objetivos). Estos factores pueden ser externos, cuando son detectados por los usuarios, o internos, aquellos que sólo son percibidos por los diseñadores o implementadores.

Los que realmente preocupan serán los externos, ya que son los que permiten evaluar la calidad de forma externa. Sin embargo, son los factores internos los que nos van a ofrecer el medio para conseguir dicha calidad externa. Por tanto, el objetivo será tener buenas propiedades internas, que nos permitan satisfacer los factores externos.

Entre los factores externos podemos destacar:

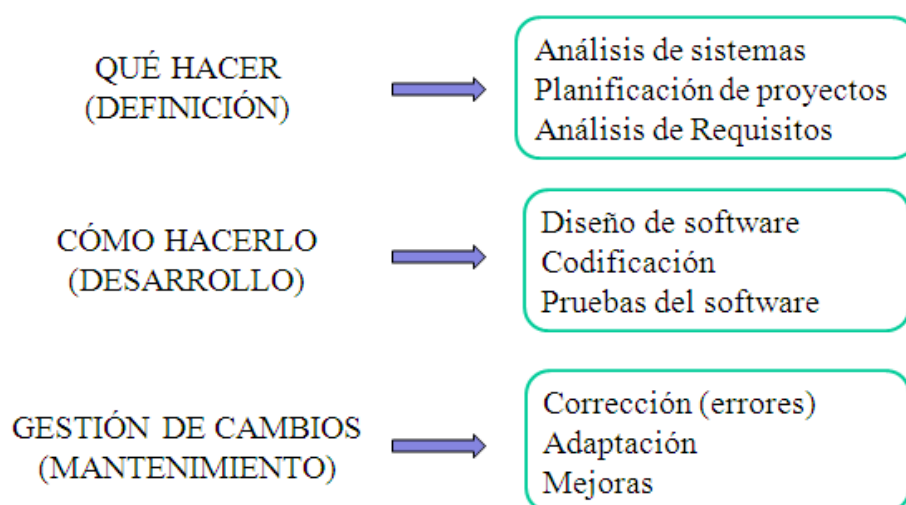
- **Mantenibilidad:** habilidad del software para evolucionar para cumplir las necesidades de cambio de los clientes. Este es un atributo crítico debido a que el cambio en el sw es una consecuencia inevitable en un entorno de negocios.
- **Confiabilidad:** habilidad del software para mantenerse operativo (funcionando). Tiene un gran número de características: fiabilidad, protección, seguridad, etc. Un sw confiable no debe causar daños físicos o económicos en el caso de una falla del sistema.
- **Eficiencia:** el sw no debe malgastar los recursos del sistema, como la memoria y los ciclos de procesamiento. Por tanto, incluye tiempos de respuesta y procesamiento, utilización de la memoria, etc.
- **Usabilidad:** el sw debe ser fácil de utilizar, sin esfuerzo adicional por el usuario para quién está diseñado. Eso significa que debe tener una interfaz de usuario apropiada y una documentación adecuada.

Otros factores destacables son la portabilidad, reutilización, corrección, robustez, compatibilidad, funcionalidad, oportunidad, extensibilidad, etc.

Por último, entre las características inherentes al software que sólo son percibidas por profesionales de la informática podemos destacar la modularidad o legibilidad.

## La IR dentro de la Ingeniería del Software

Para finalizar con este capítulo, podemos ver el siguiente esquema que muestra un resumen de las diferentes actividades en el proceso de ingeniería del software. Se puede apreciar cómo se integra de manera inherente el proceso de ingeniería de requisitos dentro del proceso global.





## Puntos clave

En este apartado se muestran aquellos puntos más importantes que el alumno debería tener claros al finalizar este capítulo:

- ¿Qué es la Ingeniería del Software?
- ¿Qué es la Ingeniería de Requisitos?
- Proceso y gestión de la IR: Obtención, análisis, especificación, validación y gestión del documento de requisitos del sistema
- Necesidad de la IR
- Ingeniería del Software e Ingeniería de Requisitos → Construir sw de calidad
- Factores de calidad del Software
- El papel de la IR en la Ingeniería del Software.