

INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE

Tema 1

Ingeniería del Software
ETS Ingeniería Informática
DSIC – UPV

Curso 2024-2025

Objetivos

- Presentar la Ingeniería del Software y explicar su importancia en el desarrollo de software de calidad
- Responder las principales cuestiones relacionadas con la Ingeniería del Software
- Introducir el Proceso del Software

Contenidos

1. Introducción

2. El software

- Características
- La Crisis del Software
- Software de calidad
- Problemática de la Industria

3. Ingeniería del Software

- Definiciones
- El Proceso del Software
- Gestión de Proyectos Software

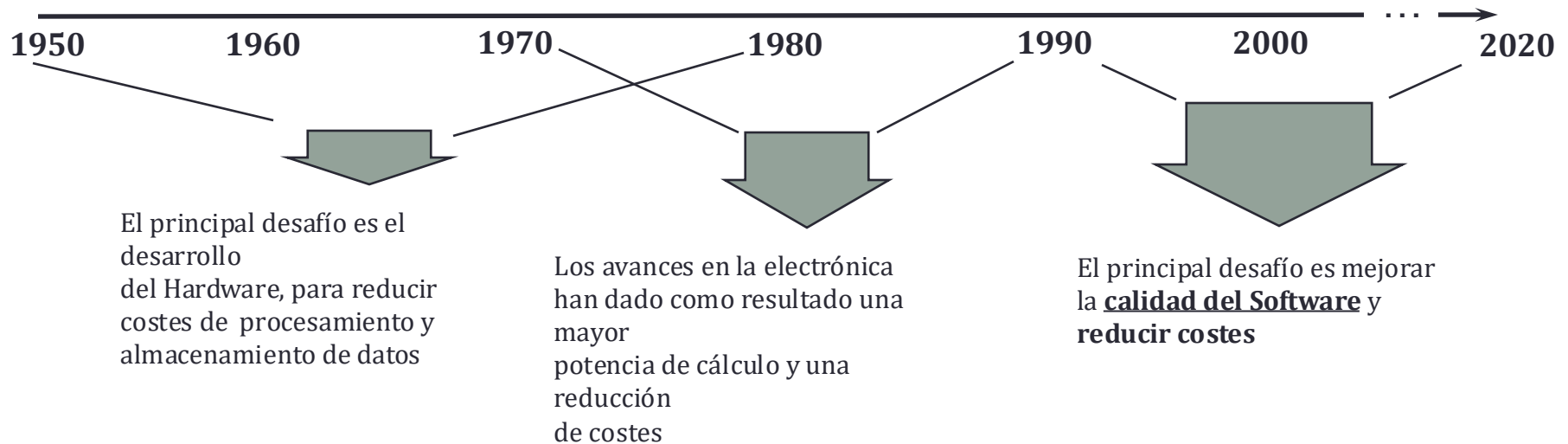
Bibliografía básica

- Sommerville, I. Software Engineering. (10th ed.) Pearson, 2016.
- Sommerville, I., Ingeniería del Software (9^a ed.), Addison-Wesley, 2011.
- Pressman, R., Ingeniería del Software. Un enfoque práctico (9^a ed.), McGraw-Hill, 2021.
- Weitzenfeld, A., Ingeniería del Software Orientada a Objetos con UML, Java e Internet. Thomson, 2005

INTRODUCCIÓN

El software marca la diferencia

En las últimas décadas el software ha superado al hardware como *factor decisivo de éxito*

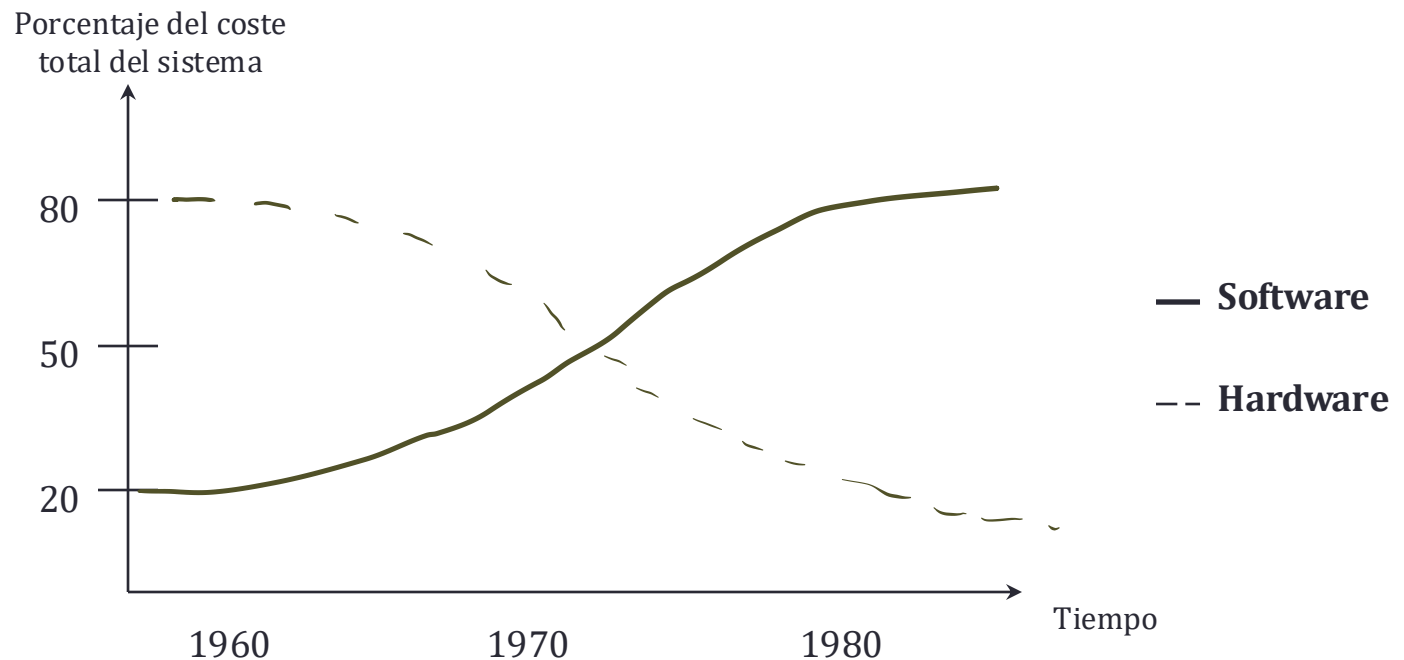


El software marca la diferencia

- En las últimas décadas, varios hitos han revolucionado la industria del software
 - la **Web** como plataforma
 - la **computación móvil**
 - la **computación en la nube**
 - nuevos dispositivos
 - nuevos lenguajes
 - nuevos métodos de desarrollo
 - ...

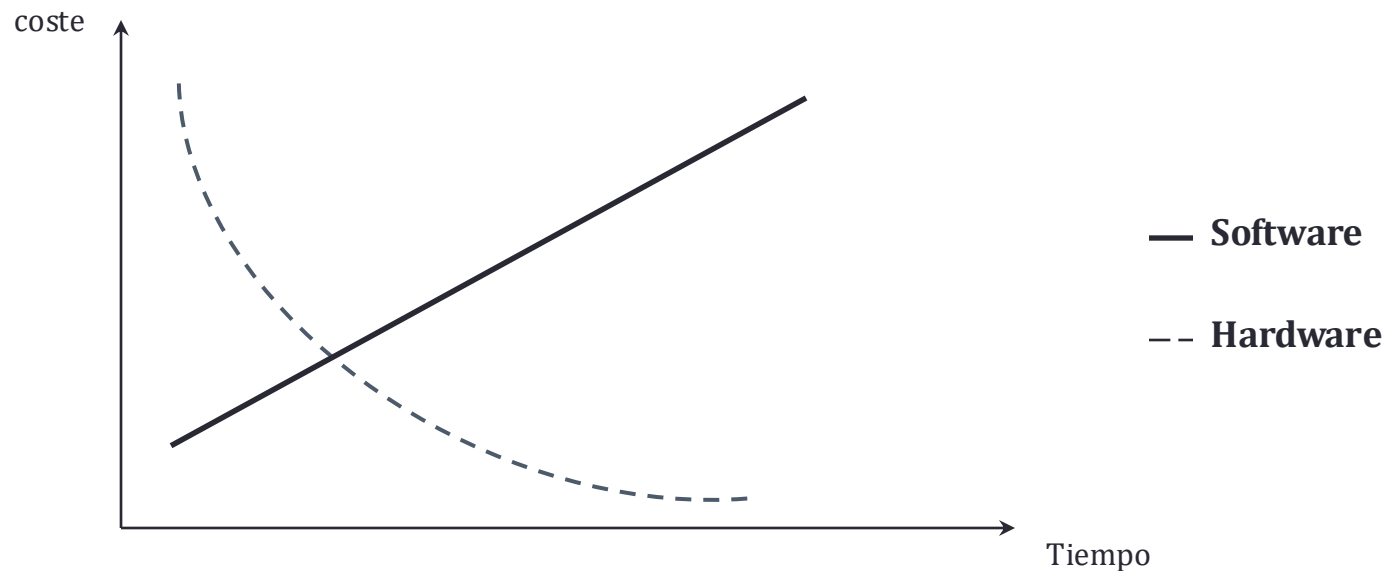
El software cuesta más...

- Evolución del coste total del sistema en función del porcentaje invertido en software y en hardware



El software cuesta más...

- Evolución del coste total del sistema en función del porcentaje invertido en software y en hardware



...¡y no solo dinero!

- <https://www5.in.tum.de/~huckle/bugse.html>
- <http://www.pcmag.com/article2/0,1759,1636333,00.asp>
- <http://www.microsiervos.com/archivo/ordenadores/10-peores-bugs.html>
- <http://catless.ncl.ac.uk/Risks>

EL SOFTWARE

- ✓ Características
- ✓ La Crisis del Software
- ✓ Software de calidad
- ✓ Problemática de la Industria

¿Qué es el software?

- **Instrucciones** que cuando se ejecutan proporcionan la función y el comportamiento deseados
- **Estructuras de datos** que facilitan a los programas el manipular adecuadamente la información
- **Documentos** que describen la operación y uso de los programas

Características del software

El software es un elemento lógico:

- se *desarrolla*, no se fabrica en un sentido clásico
- no se estropea, se *deteriora* debido a los *cambios*
- la mayoría se construye a medida, en vez de *ensamblar componentes* existentes
 - Tendencia cambiante parcialmente en las últimas 2 décadas

Evolución del software (1/2)

- El **contexto** en el que se ha desarrollado el software está fuertemente ligado al desarrollo de los sistemas informáticos

Crisis del Software



La crisis del software

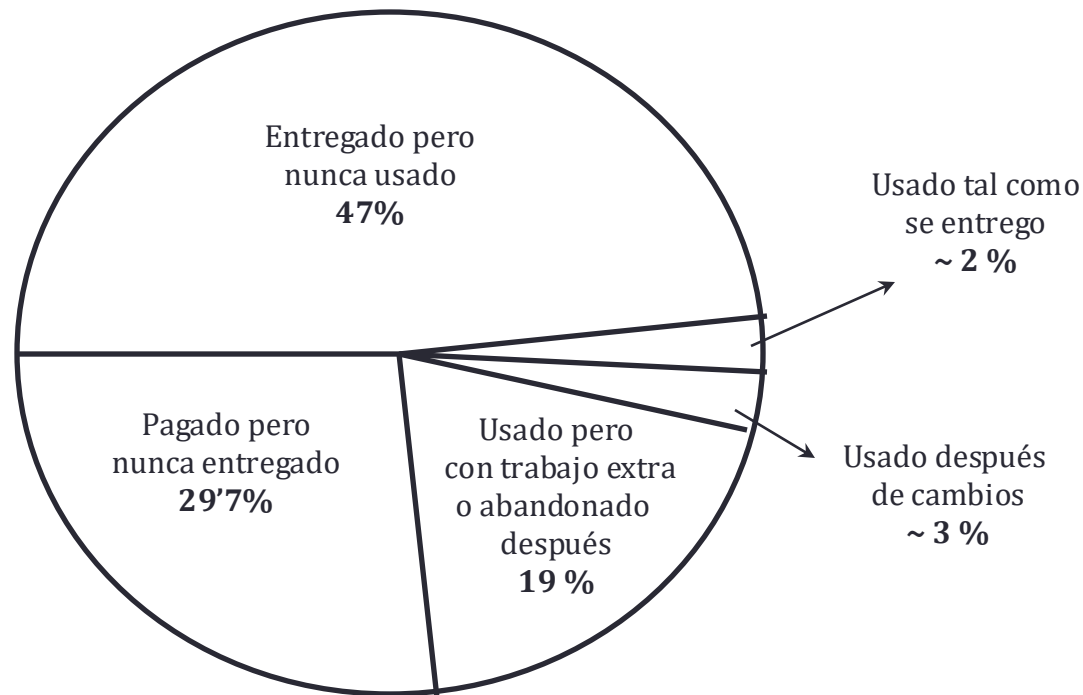
- Los productos exceden la estimación de costes
- Notables retrasos en la fecha de entrega
- Prestaciones inadecuadas
- Mantenimiento casi imposible
- Modificaciones a precios prohibitivos
- Falta de fiabilidad del producto software

¡Software de baja calidad!

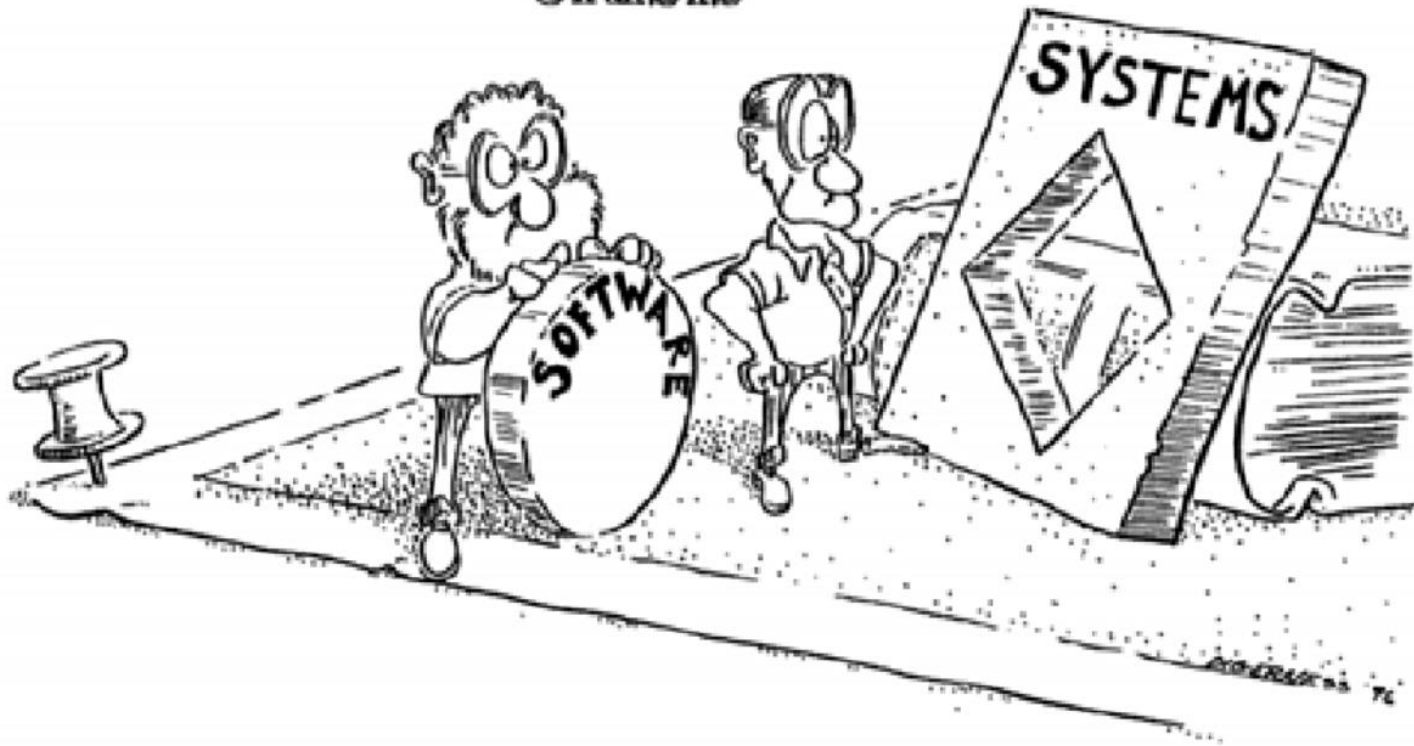
La crisis del software

- **Inversión** en desarrollo de Sistemas Software.

- Estudio año 1979 (Total: \$ 6.8 millones)



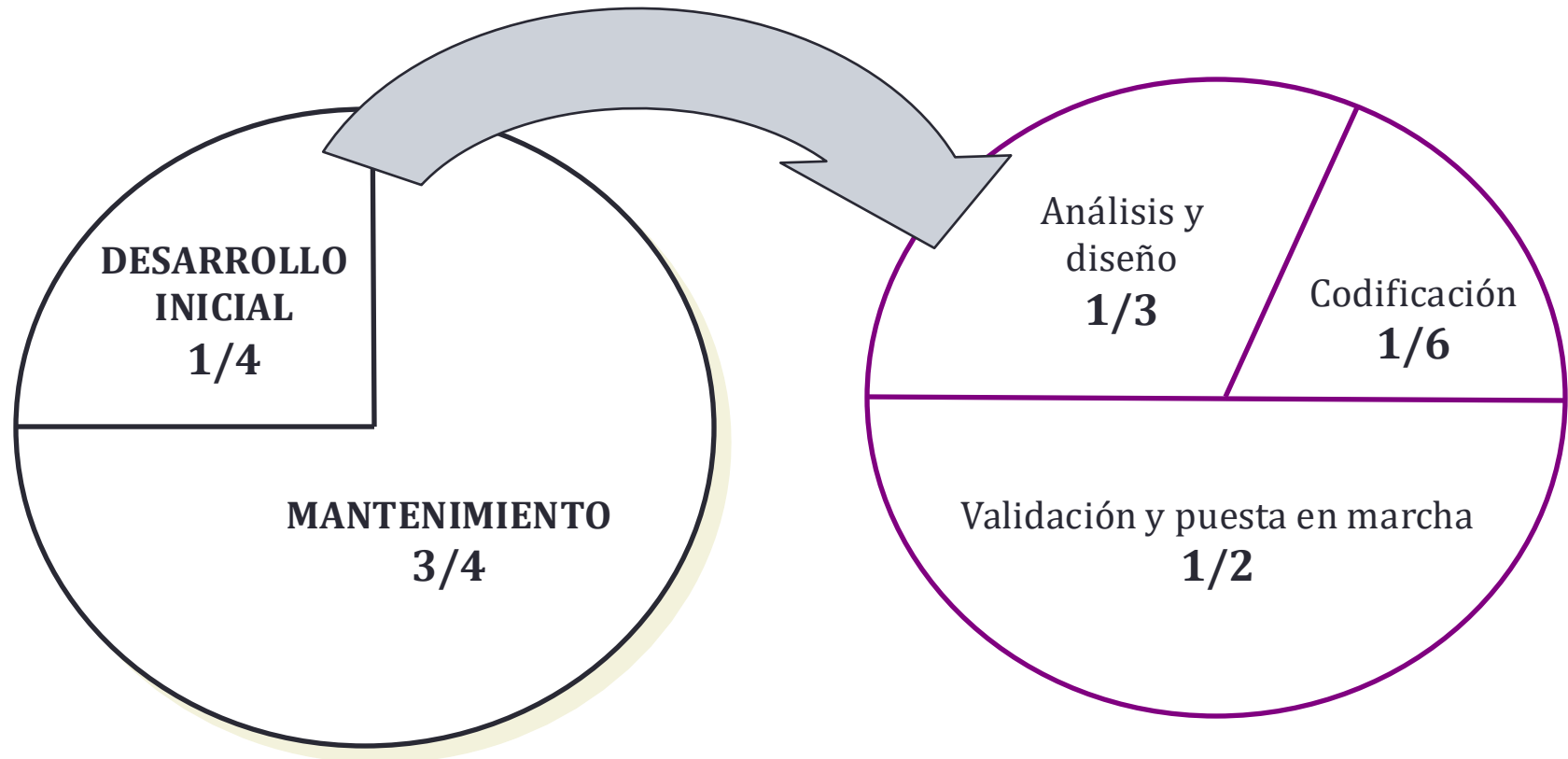
THE SOFTWARE CRISIS



<http://histinf.blogs.upv.es/2010/12/28/ingenieria-del-software/>

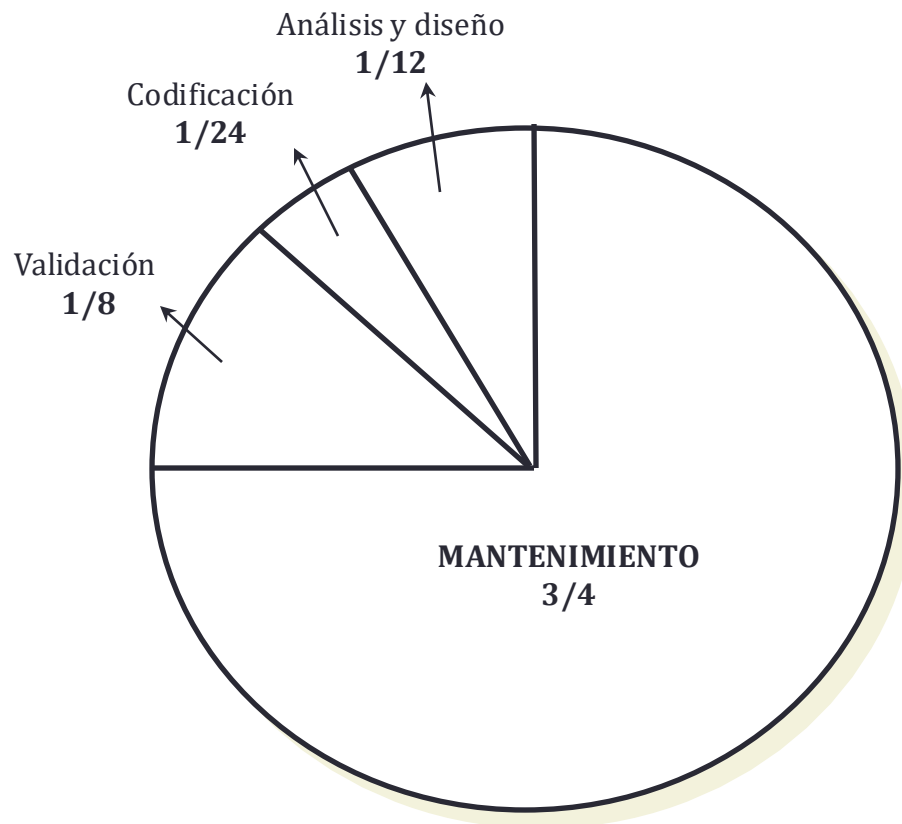
La crisis del software

- **Inversión** en desarrollo de Sistemas Software, desglosada en las distintas fases:



La crisis del software

- Resumen de la **inversión** en el desarrollo de Sistemas Software:

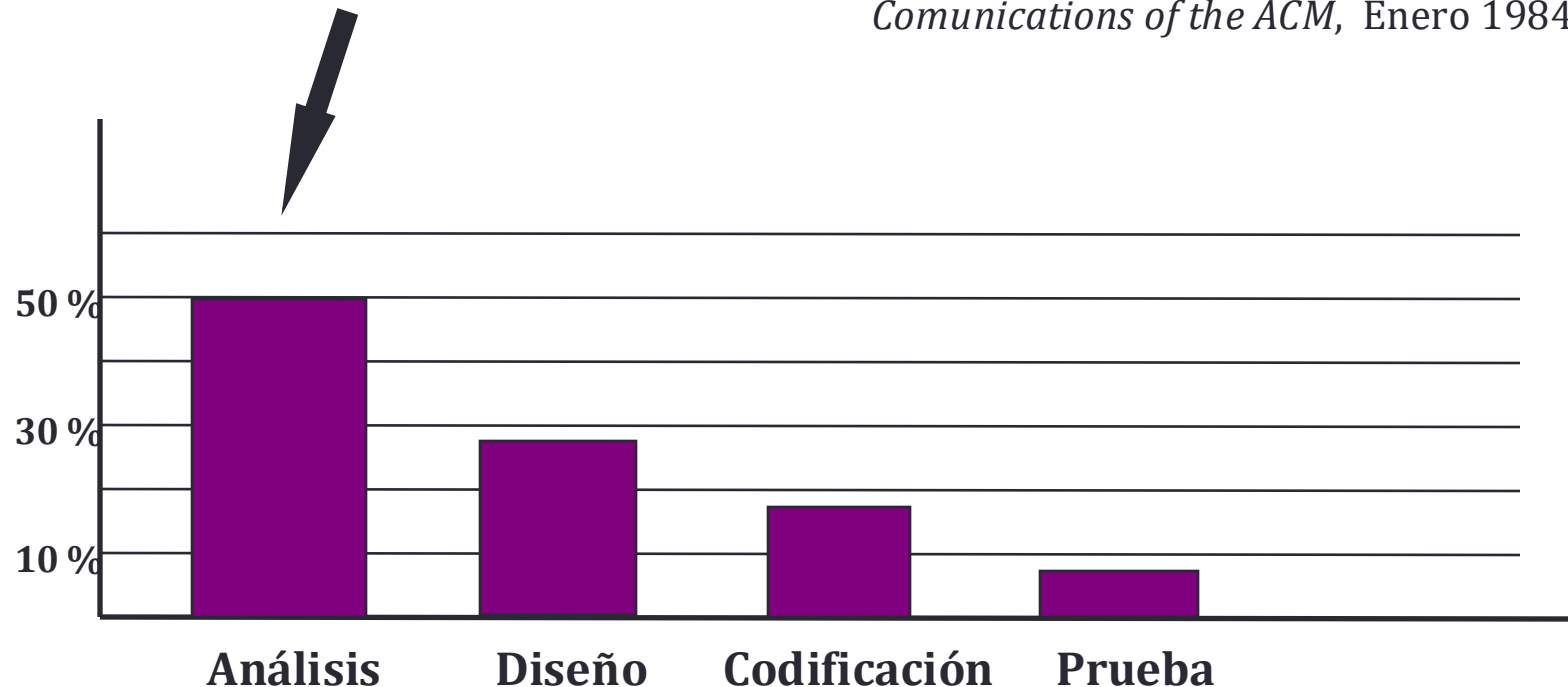


- Análisis y Diseño \Rightarrow 8 %
- Codificación \Rightarrow 4 %
- Validación y Mantenimiento \Rightarrow 88 %

La crisis del software

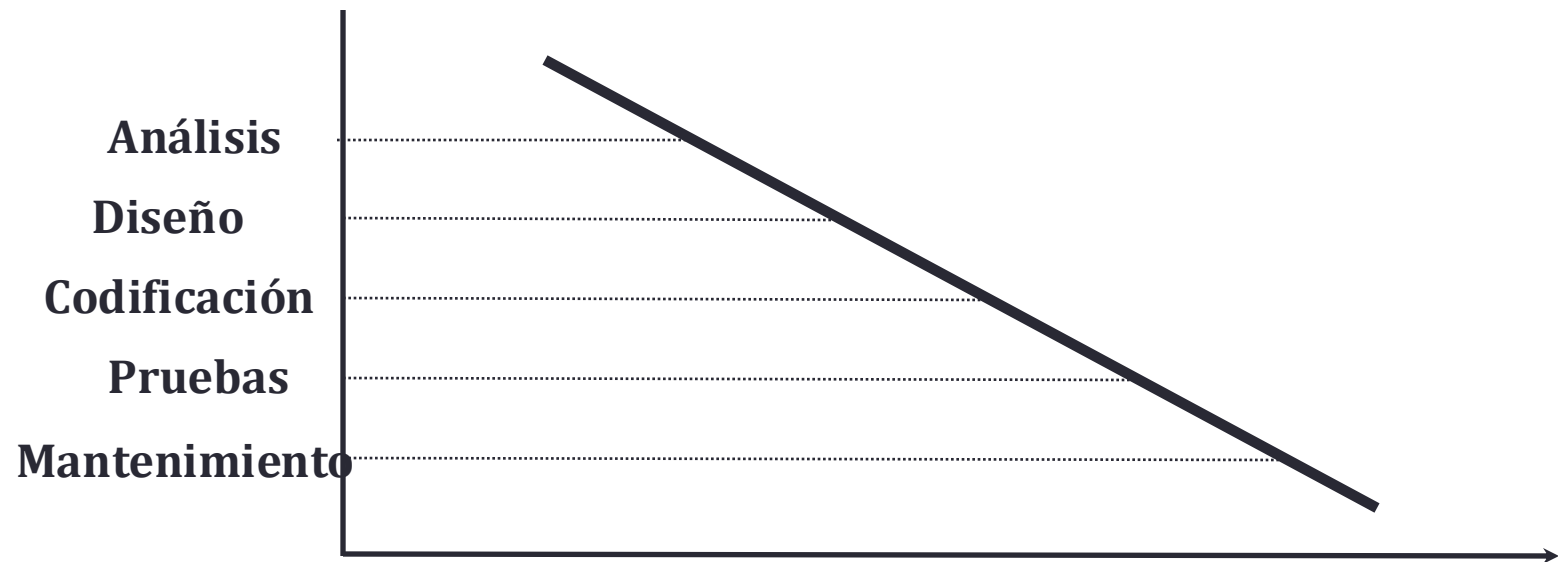
- **Errores** cometidos en el desarrollo de Sistemas Software, por fases:

Communications of the ACM, Enero 1984



La crisis del software

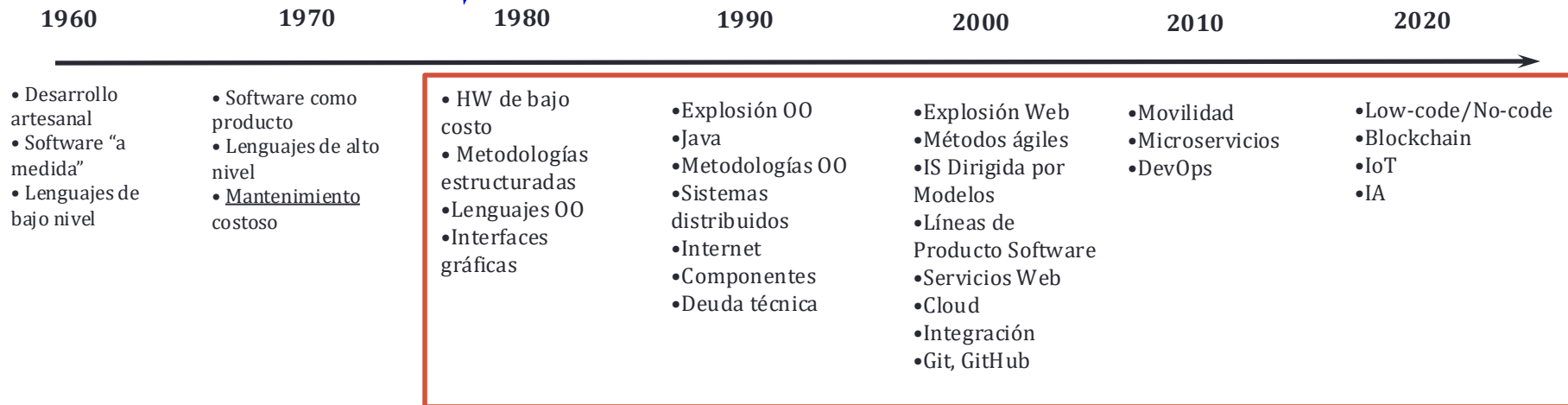
- **Coste** de eliminación de los errores cometidos:



Evolución del software (2/2)

- La evolución de la tecnología ha supuesto numerosos retos para la I+D en Ingeniería del software

Nacimiento de la Ingeniería del Software

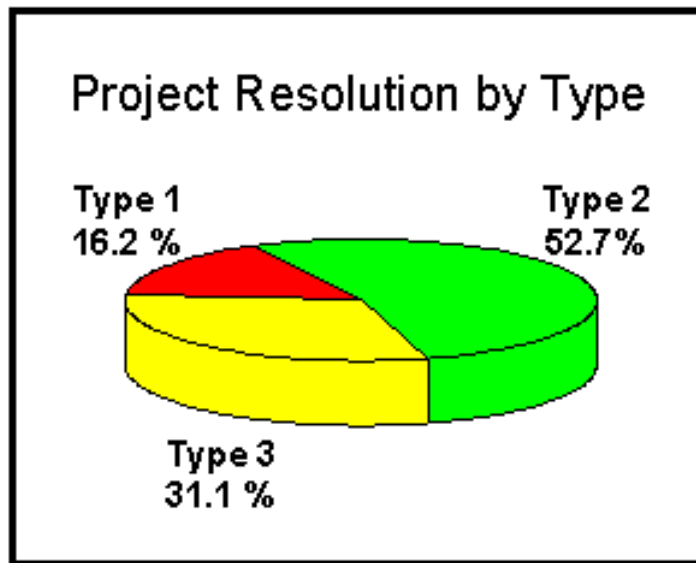


CHAOS report (*Standish Group*)

<http://www.standishgroup.com/service/index>

Inversión en desarrollo de Sistemas Software.

- Año **1994** (Total: \$ 250 Kmillones/año - - 175.000 proyectos)



• **Tipo 1 (Éxito).** El proyecto es finalizado en el tiempo acordado y dentro del presupuesto, con todas las características y funcionalidades inicialmente especificadas.

• **Tipo 2 (Cambios).** El proyecto es finalizado pero por en un plazo más largo y con un presupuesto mayor; además ofrece menos características y funcionalidades que inicialmente se especificaron.

• **Tipo 3 (Cancelado).** El proyecto es cancelado en algún punto durante el desarrollo del mismo.

https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf

CHAOS report (*Standish Group*)

Informe CHAOS ...

RESOLUTION					
	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

Project resolution results from CHAOS research for years 2004 to 2012.

THE CHAOS MANIFESTO

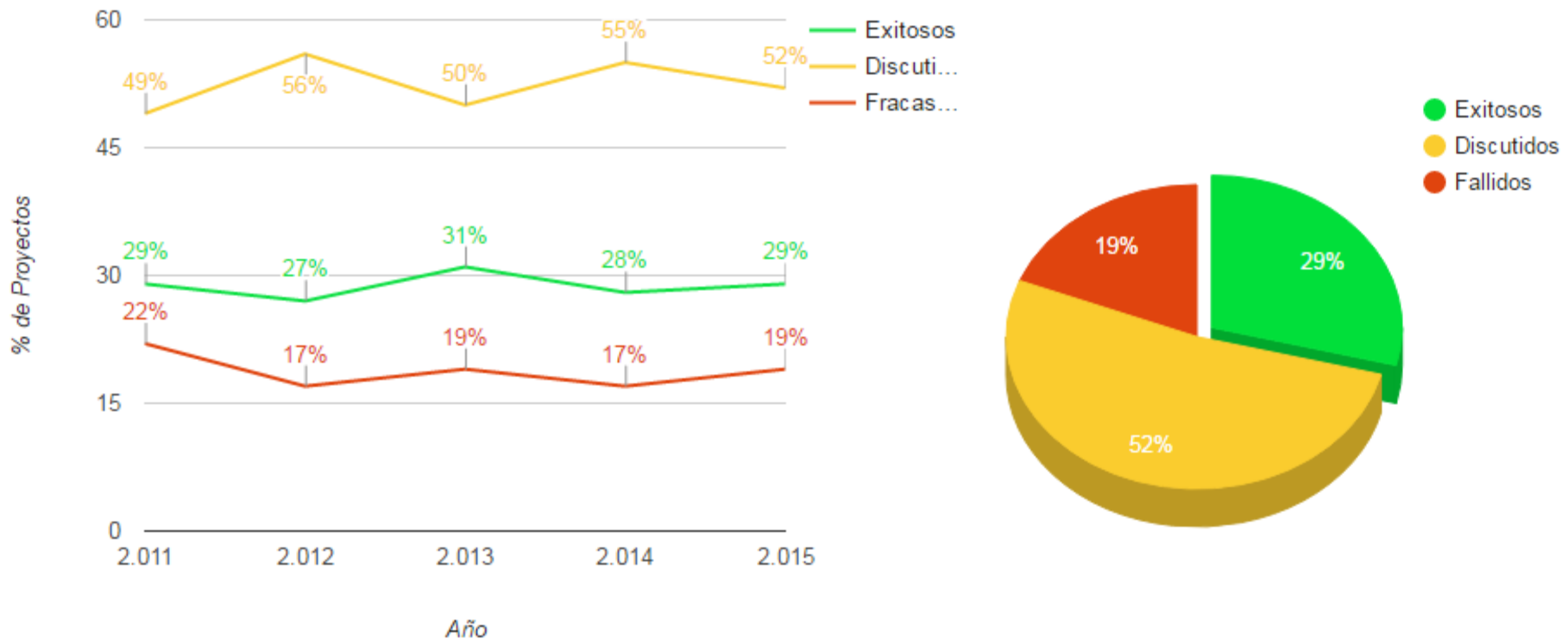
1

Copyright © 2013. The CHAOS Manifesto is protected by copyright and is the sole property of The Standish Group International, Incorporated. It may not under any circumstances be retransmitted in any form, repackaged in any way, or resold through any media. All rights reserved.

<http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>

CHAOS report (*Standish Group*)

Informe CHAOS 2015 ...

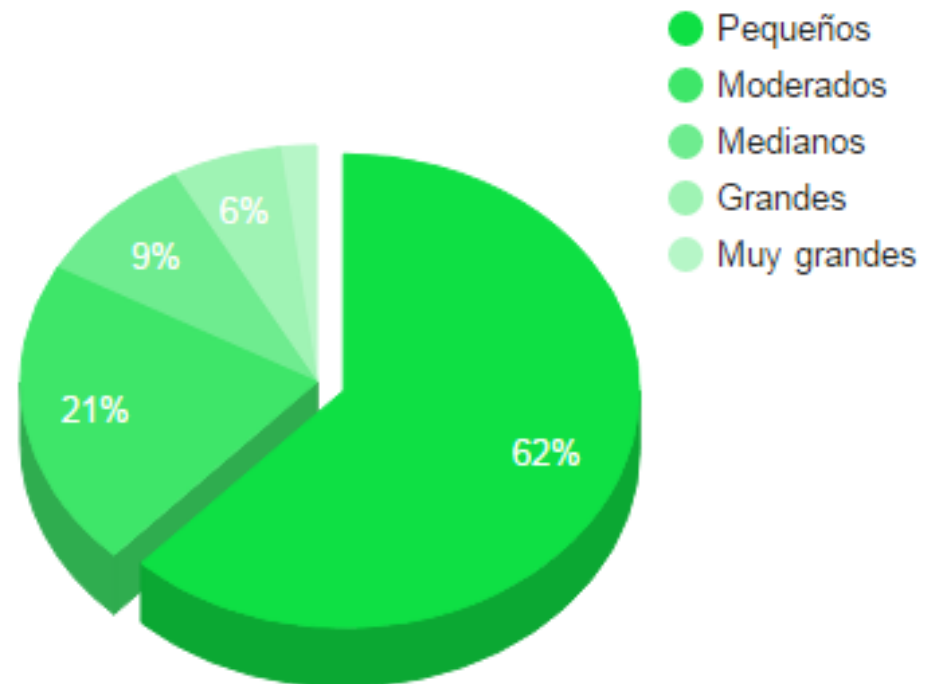


CHAOS report (*Standish Group*)

Informe CHAOS 2015 ...

¿Es importante el tamaño de los proyectos software?

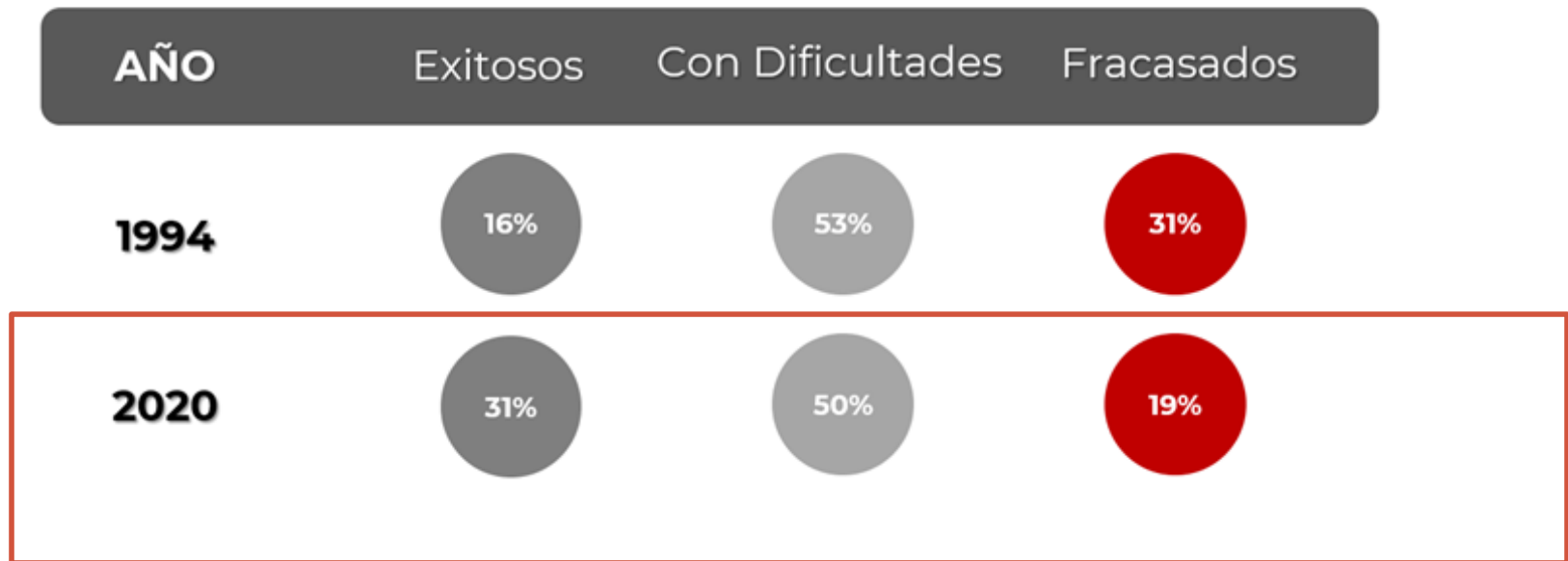
% sobre los proyectos exitosos
2011-2015



CHAOS report (*Standish Group*)

Informe CHAOS 2020

Comparativa **CHAOS Report** 1994 & 2020



¿Qué factores afectan al éxito de un proyecto software?

...

¿Qué factores afectan al éxito de un proyecto software?

... *Software de Calidad*

Software de calidad

- La finalidad es producir software de gran calidad

¿Qué es software de calidad?

Concordancia con:

- Los **requisitos funcionales y de rendimiento** establecidos explícitamente
- Los **estándares** de desarrollo explícitamente documentados
- Las **características implícitas** que se espera de todo software desarrollado profesionalmente

Factores de calidad

- La clasificación de los factores de calidad se centra en tres aspectos importantes de un producto software
 1. Sus características **operativas**
 2. Su capacidad de **soportar los cambios**
 3. Su **adaptabilidad** a nuevos entornos
- Se deben medir, directa o indirectamente, a lo largo de todo el proceso de desarrollo

Factores de calidad

Corrección ¿Hace lo que quiero?

Fiabilidad ¿Lo hace de forma fiable todo el tiempo?

Eficiencia ¿Se ejecutará en el Hw lo mejor que pueda?

Integridad ¿Es seguro?

Facilidad de uso ¿Está diseñado para ser usado?

**Características
operativas**

Facilidad de mantenimiento ¿Puedo corregirlo?

Flexibilidad ¿Puedo cambiarlo fácilmente?

Facilidad de prueba ¿Puedo probarlo?

**Capacidad de soportar
cambios**

Reusabilidad ¿Podré reusar alguna parte del Sw?

Portabilidad ¿Podré usarlo en otra máquina o SO?

Facilidad de interoperación ¿Puedo hacerlo interactuar con otro sistema?

**Adaptabilidad a
nuevos
entornos**

Problemática de la industria del software

- Los productos no son de calidad
- Altos costes de desarrollo y mantenimiento
- Grandes retrasos

Causas:

- ✓ *Poca inversión y esfuerzo en el análisis y la especificación*
- ✓ *Uso de lenguajes informales y modelos inadecuados*
- ✓ *Naturaleza no física de la programación*
- ✓ *Poca teoría y poca difusión*
- ✓ *Productos ya en el mercado dificultan la innovación*
- ✓ *Mucha artesanía*
- ✓ *Trabajo en grupo*
- ✓ *Comunicación con usuario*
- ✓ *Gestión de proyectos por no informáticos*

Soluciones

- Formalización:
 - métodos de razonamiento formal (lenguajes formales de especificación ejecutables: lógica + álgebra)
 - nuevos modelos de desarrollo y modificación del ciclo de vida
- Difusión de los avances tecnológicos
 - nuevos paradigmas de programación
 - Arquitecturas, protocolos, modelos de computación
- Inversión en herramientas
 - entornos de desarrollo modernos
 - generadores de documentación

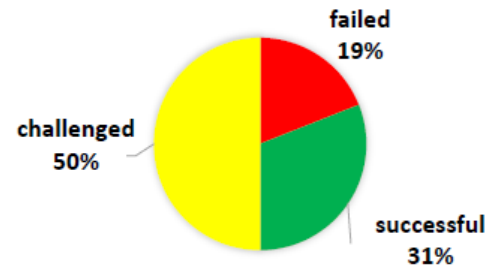
¿Qué factores afectan al éxito de un proyecto software?

... *Software de Calidad*

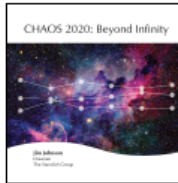
... *Otros*

Project Success Quick Reference Card

Based on CHAOS 2020: Beyond Infinity Overview, January 2021, QRC by Henny Portman



Modern measurement (software projects)



Good Sponsor, Good Team, and Good Place are the only things we need to improve and build on to improve project performance.



The Good Place is where the sponsor and team work to create the product. It's made up of the people who support both sponsor and team. These people can be helpful or destructive. It's imperative that the organization work to improve their skills if a project is to succeed. This area is the hardest to mitigate, since each project is touched by so many people. Principles for a Good Place are:



- The Decision Latency Principle
- The Emotional Maturity Principle
- The Communication Principle
- The User Involvement Principle
- The Five Deadly Sins Principle
- The Negotiation Principle
- The Competency Principle
- The Optimization Principle
- The Rapid Execution Principle
- The Enterprise Architecture Principle

Successful project Resolution by Good Place Maturity Level:

highly mature	50%
mature	34%
moderately mature	23%
not mature	23%

The Good Team is the project's workhorse. They do the heavy lifting. The sponsor breathes life into the project, but the team takes that breath and uses it to create a viable product that the organization can use and from which it derives value. Since we recommend small teams, this is the second easiest area to improve. Principles for a Good Team are:



- The Influential Principle
- The Mindfulness Principle
- The Five Deadly Sins Principle
- The Problem-Solver Principle
- The Communication Principle
- The Acceptance Principle
- The Respectfulness Principle
- The Confrontationist Principle
- The Civility Principle
- The Driven Principle

Successful project Resolution by Good Team Maturity Level:

highly mature	66%
mature	46%
moderately mature	21%
not mature	1%

The Good Sponsor is the soul of the project. The sponsor breathes life into a project, and without the sponsor there is no project. Improving the skills of the project sponsor is the number-one factor of success – and also the easiest to improve upon, since each project has only one.

Principles for a Good Sponsor are:

- The Decision Latency principle
- The Vision Principle
- The Work Smart Principle
- The Daydream Principle
- The Influence Principle
- The Passionate Principle
- The People Principle
- The Tension Principle
- The Torque Principle
- The Progress Principle



Successful project Resolution by Good Sponsor Maturity Level:

highly mature	67%
mature	33%
moderately mature	21%
not mature	18%

¿Qué factores afectan al éxito de un proyecto software?

... *Software de Calidad*

... *Otros*

...

LA INGENIERÍA DEL SOFTWARE

- ✓ Definiciones
- ✓ El Proceso del Software

Definiciones

- **B. Boehm:**
 - *“La INS supone la aplicación práctica y sistemática del conocimiento científico a la producción de programas que se desarrollan a tiempo y dentro de las estimaciones de presupuesto y la correspondiente documentación para desarrollarlos, instalarlos, usarlos y mantenerlos”*
- **R. Pressman:**
 - *“La INS es una disciplina que integra métodos, herramientas y procedimientos para el desarrollo de SW de computador”.*
- **A. Davis:**
 - *“La INS es la aplicación de principios científicos para: (1) la transformación ordenada de un problema en una solución SW y (2) el mantenimiento del mismo durante toda su vida útil”.*
- **I. Sommerville:**
 - *“La INS es una disciplina ingenieril que abarca todos los aspectos de la producción de software”*

En definitiva...

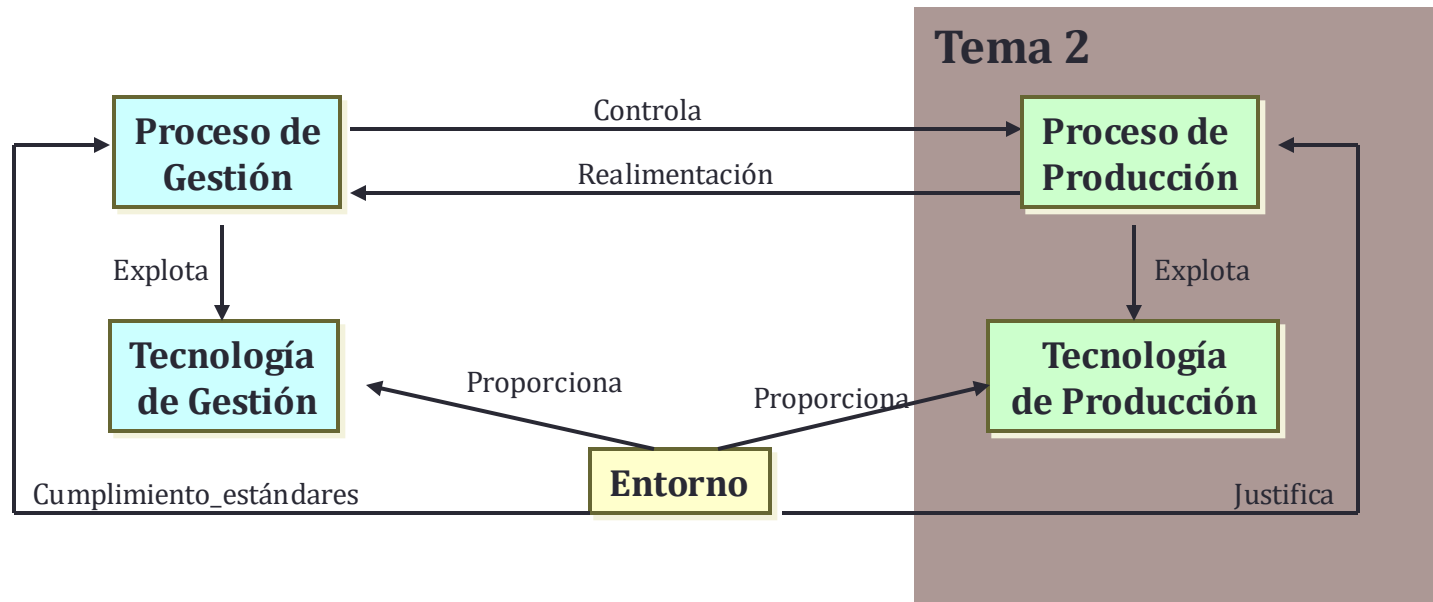
- La INS **es algo más que programar**
- El proceso de la INS comienza bastante antes de escribir líneas de código y continúa después de que la primera versión del producto haya sido completada
- Son cruciales la planificación, el seguimiento y el control rigurosos de los proyectos software

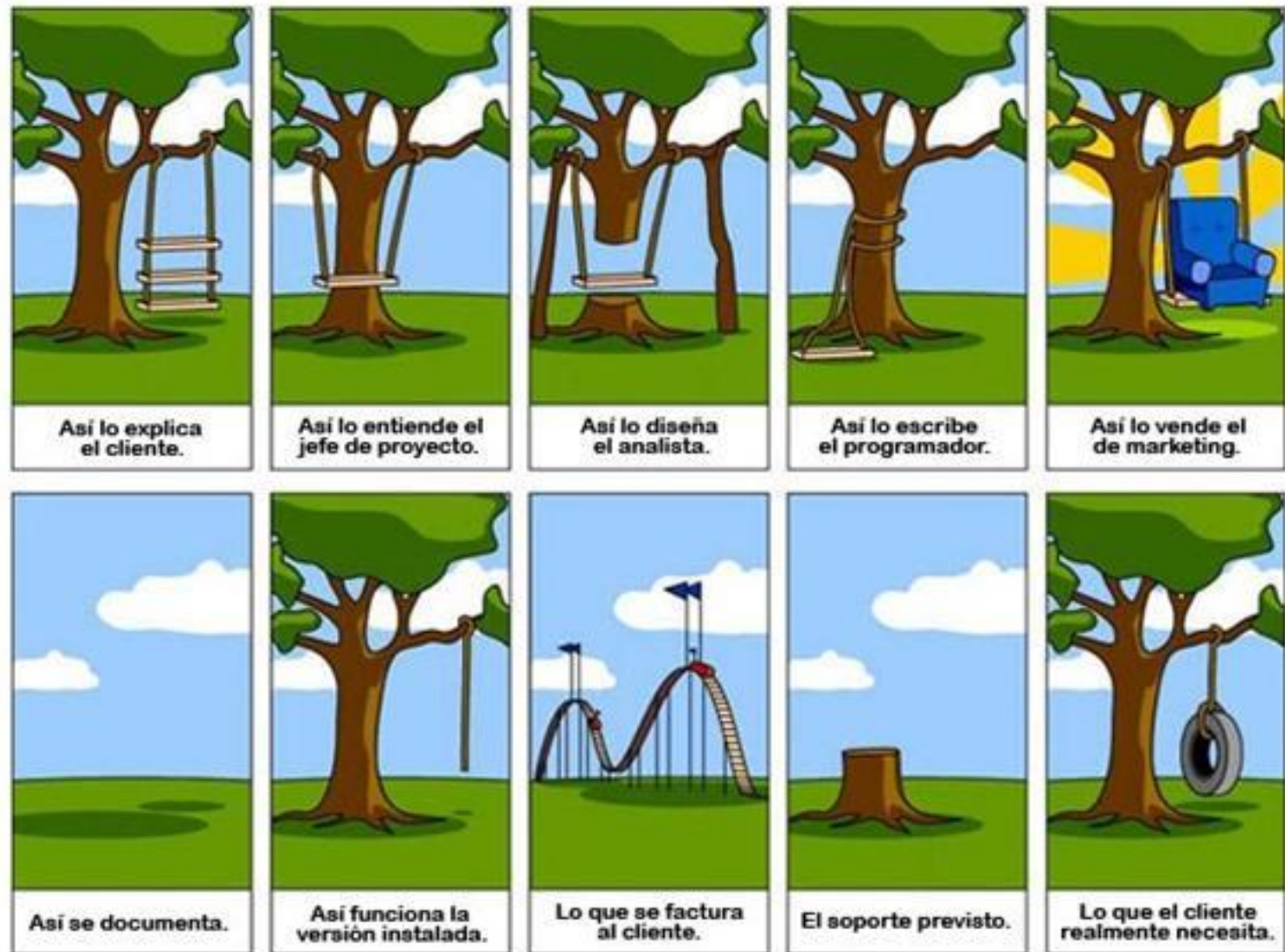
La importancia del proceso

- Los nuevos retos a los que se enfrenta el desarrollo de software requieren respuestas rápidas y eficaces a cambios en los requisitos.
- La especificación clara de un proceso de desarrollo, y el uso de herramientas para su ejecución y monitorización, se han convertido en un requisito en sí mismas

El Proceso del Software

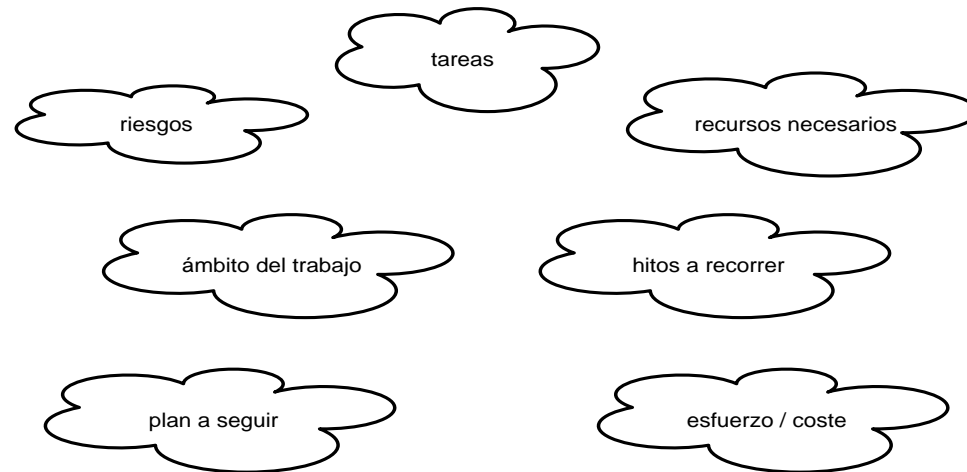
- Establece un marco para el desarrollo de software





* Gestión de proyectos software

- La administración de un proyecto software es el primer nivel del proceso de ingeniería del software y cubre todo el proceso de desarrollo



* Gestión de proyectos software

Ingeniería del Software frente a Ingeniería

• Semejanzas

- Las actividades a realizar no son específicas de la administración de proyectos de software
- Muchas técnicas de gestión de proyectos de ingeniería son igualmente aplicables a los proyectos software
- Muchos de los problemas que aparecen en los sistemas de ingeniería complejos, aparecen también en los sistemas software (tiempo, recursos, cambio en especificaciones...)

• Diferencias

- El producto (software) es intangible y flexible
- El proceso de desarrollo de software no es estándar. Existen distintas propuestas y cada organización adopta una distinta
- Muchas veces los proyectos software son “únicos”

* Gestión de proyectos software

- Actividades que incluye la gestión de proyectos:
 - Redacción de la propuesta
 - Planificación del proyecto
 - Estimación del coste del proyecto
 - Selección y evaluación del personal
 - Seguimiento y control del proyecto
 - Redacción y presentación de informes

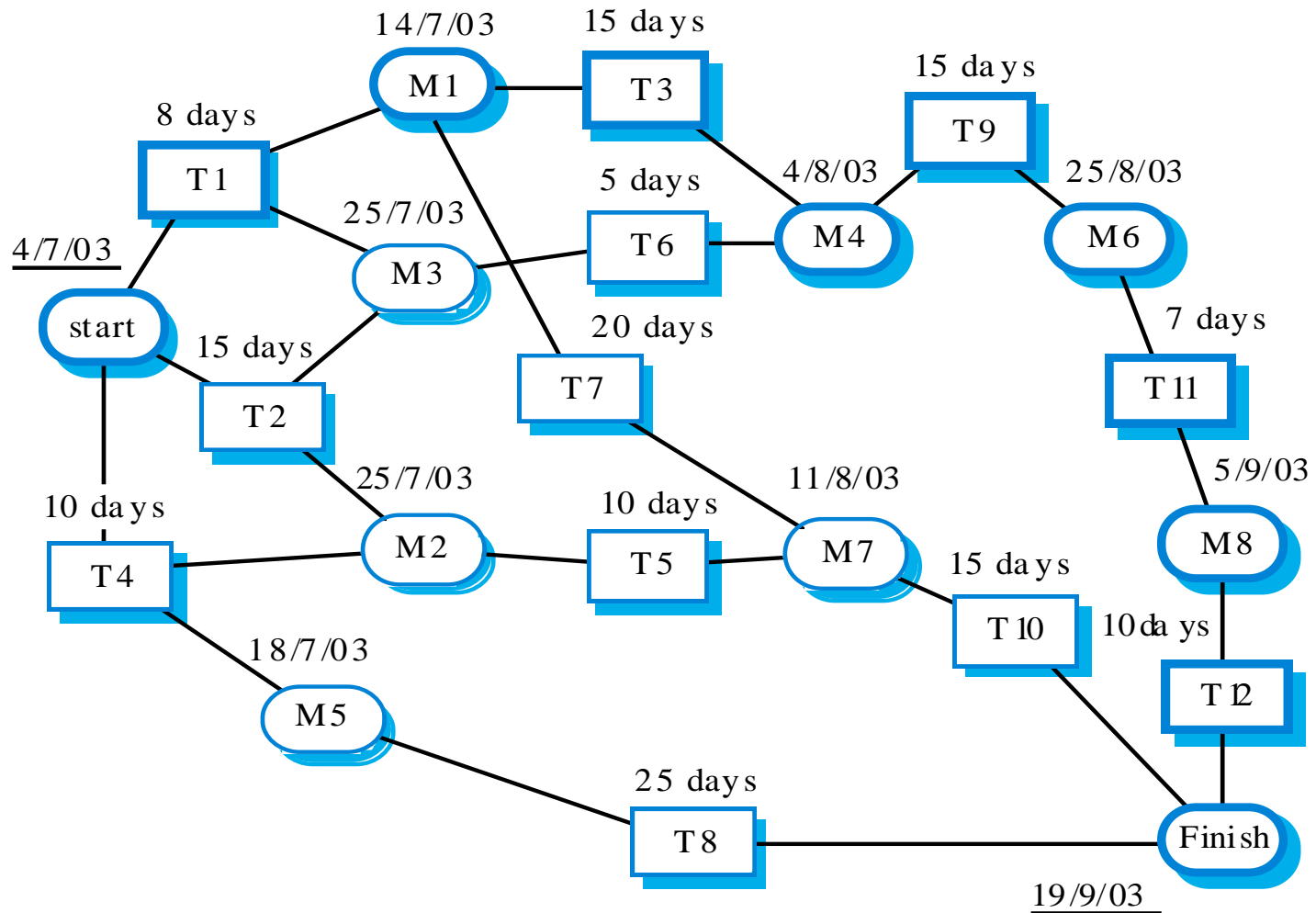
* Gestión de proyectos software

El Plan del Proyecto - Estructura

1. **Introduction** Objetivos y restricciones (presupuesto, tiempo...)
2. **Organización del proyecto** Organización del equipo (personal-roles)
3. **Análisis de riesgo** Riesgos, probabilidades y estrategias
4. **Requisitos de recursos software y hardware**
Compras, precios, fechas entrega ...
5. **División del trabajo** Actividades, Hitos y Productos a entregar
6. **Planificación** Dependencias entre actividades,
Tiempo estimado y asignación de personal
7. **Mecanismos de Supervisión e Informes**

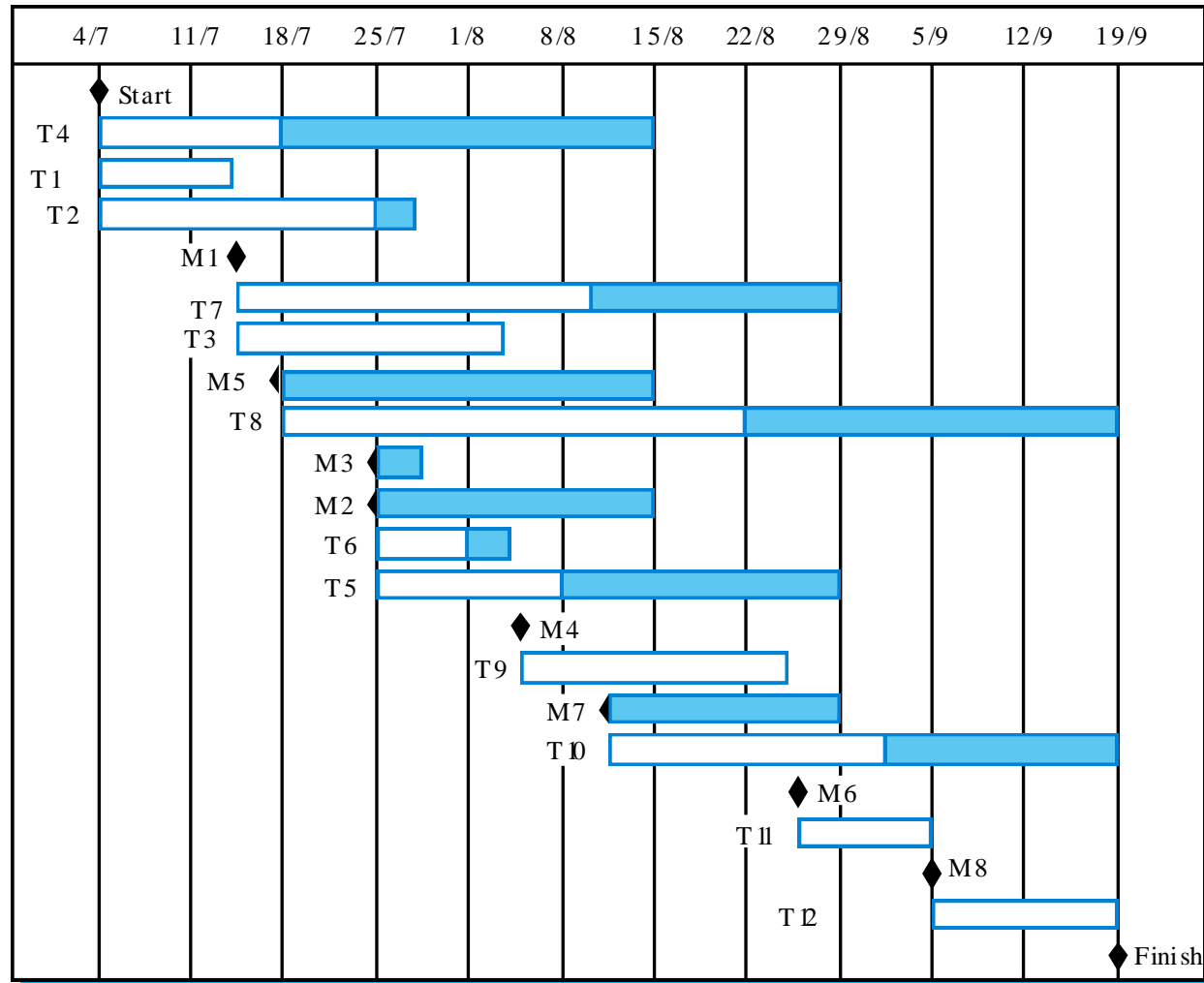
* Gestión de proyectos software

Red de Tareas (Diagrama PERT)



* Gestión de proyectos software

Diagrama de Gantt



En este curso de ISW ...

- Vamos a estudiar una pequeña parte de todo el universo de la Ingeniería del software



1. **"Software Engineering" de Ian Sommerville**

- Edición: 10ª edición (2015)

- Páginas: 816 páginas

2. **"Software Engineering: A Practitioner's Approach" de Roger S. Pressman**

- Edición: 8ª edición (2014)

- Páginas: 976 páginas

3. **"Fundamentals of Software Engineering" de Carlo Ghezzi, Mehdi Jazayeri y Dino Mandrioli**

- Edición: 2ª edición (2002)

- Páginas: 672 páginas

4. **"Software Engineering" de Hans van Vliet**

- Edición: 3ª edición (2008)

- Páginas: 800 páginas

5. **"Object-Oriented Software Engineering: Using UML, Patterns, and Java" de Bernd Bruegge y Allen H. Dutoit**

- Edición: 3ª edición (2009)

- Páginas: 720 páginas

Tras este curso...

- Rama de Ingeniería del Software
 - <https://bit.ly/45N2U1s>
- Máster en Ingeniería y Tecnología de Sistemas Software
 - <https://bit.ly/4f0m7kP>