

Métodos ágiles en el desarrollo de software

José Hilario Canós Cerdá

Departament de Sistemes Informàtics i
Computació

Universitat Politècnica de València

Contenido de la charla

Introducción

La Crisis del Software

La Ingeniería del Software

El proceso del software

Métodos Ágiles en el desarrollo del Software

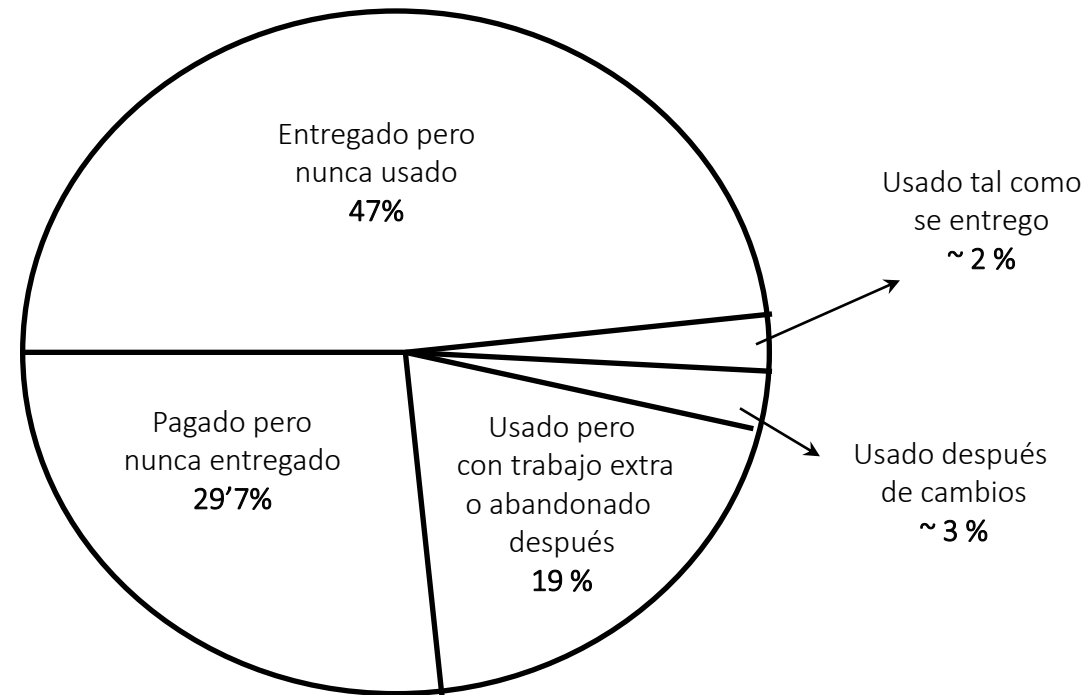
Extreme Programming

Introducción

La crisis del software

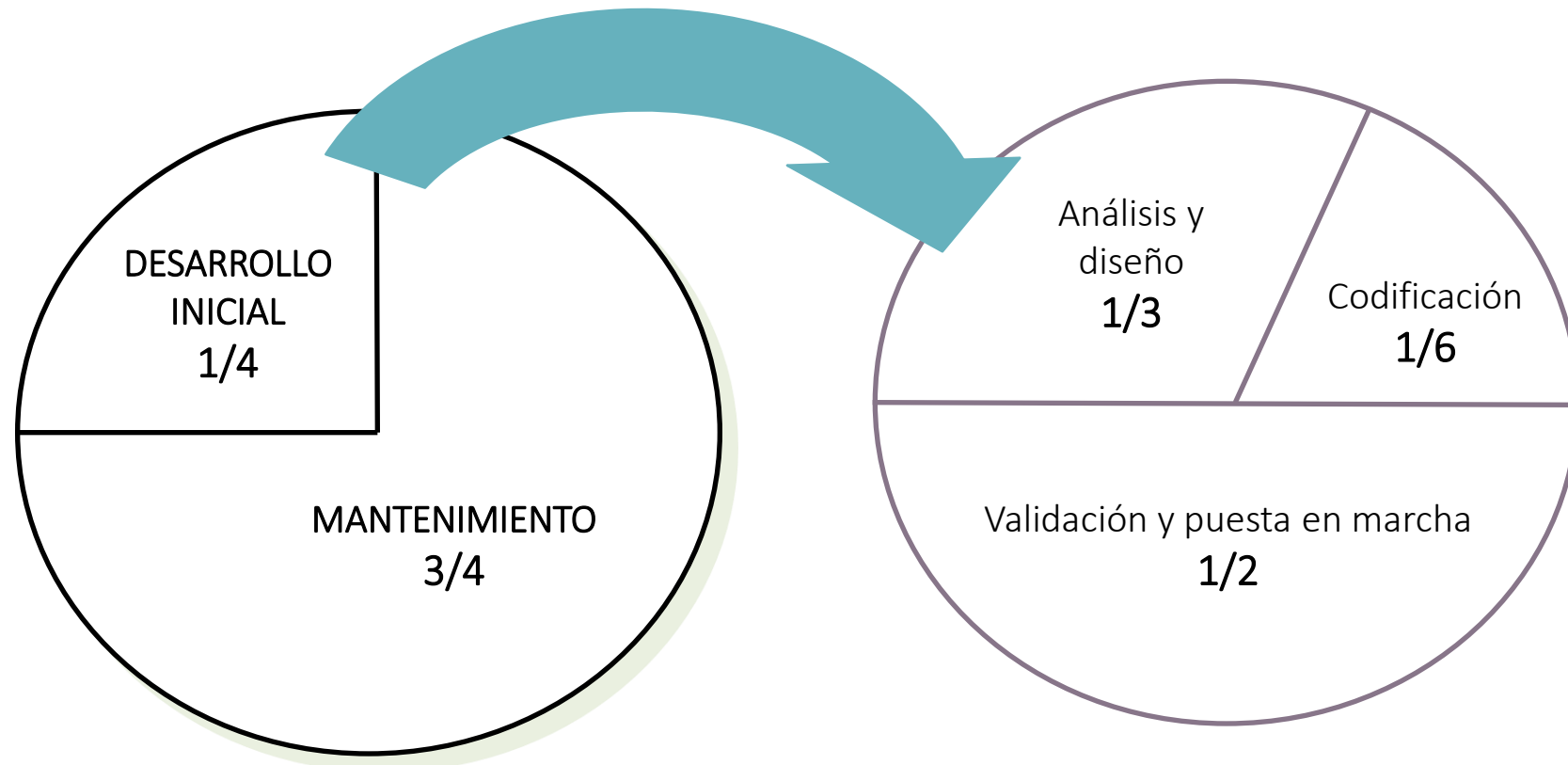
Inversión en desarrollo de Sistemas Software.

Año 1979 (Total: \$ 6.8 millones)



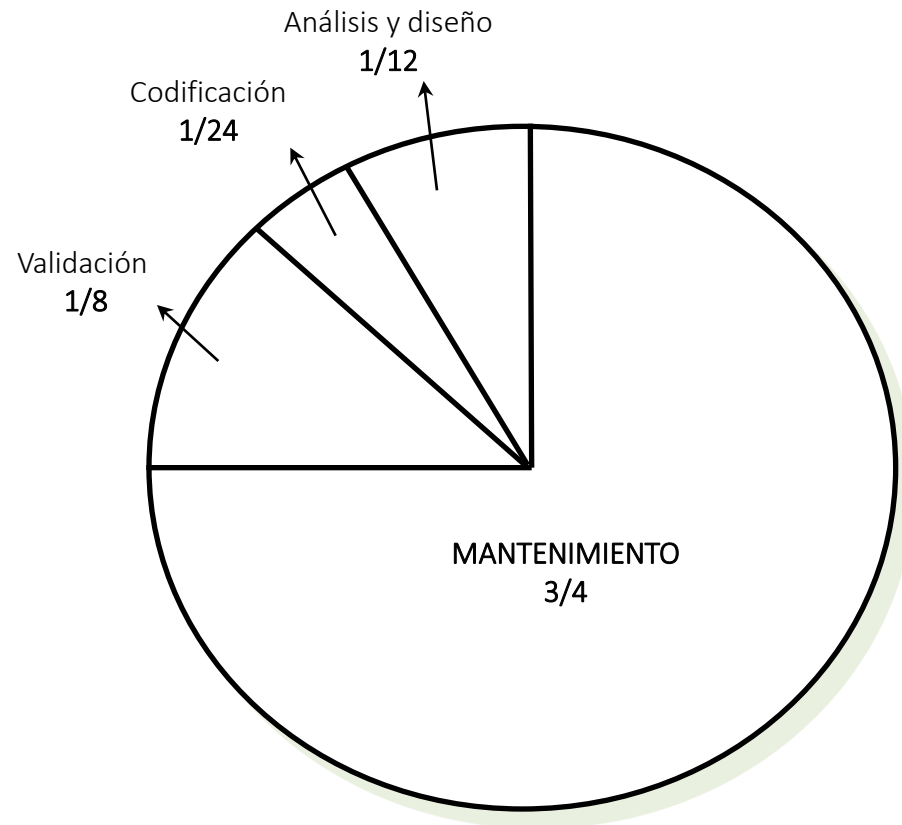
La crisis del software

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 %

Ingeniería del Software

“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”

(B. Boehm)

Ingeniería del Software

“La INS es una disciplina que integra métodos, herramientas y procedimientos para el desarrollo de SW de computador”.

(R. Pressman)

Ingeniería del Software

“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”.*

(A. Davis)

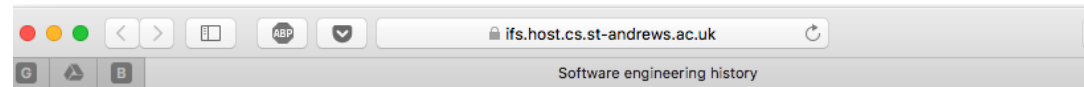
Ingeniería del Software

“La INS es una disciplina ingenieril que abarca todos los aspectos de la producción de software”

(I. Sommerville)

Un poco de historia

<http://bit.ly/2fAlkar>



Software engineering history

The term 'software engineering' was suggested at conferences organized by NATO in 1968 and 1969 to discuss the 'software crisis'. The software crisis was the name given to the difficulties encountered in developing large, complex systems in the 1960s. It was proposed that the adoption of an engineering approach to software development would reduce the costs of software development and lead to more reliable software.

The [Wikipedia article](#) on the history of SE was, at the time of writing (April 2010) quite good up until 1989. Then I think it lost its way.

Key dates in the history of software engineering are:

1968: Nato conference on software engineering. Publication of Dijkstra's note on the dangers of the goto statement in programs.

[NATO conference proceedings and conference info](#)

Early 1970s. Development of the notions of structured programming. Publication of Parnas's paper on information hiding. Development of Pascal programming language. Development of Smalltalk languages which introduced notions of object-oriented development.

Late 1970s. Early use of software design methods such as Yourdon and Constantine's structured design. Development of first programming environments.

Early 1980s. Development of the Ada programming language which included notions of structured programming and information hiding. Proposals for software engineering environments. CASE tools introduced to support design methods. Development of algorithmic approaches to software costing and estimation. Publication of the 1st edition of this book as the first student textbook on software engineering.

[The history of Ada](#)

Late 1980s. Increased use of object-oriented programming through languages such as C++ and Objective-C. Introduction of object-oriented design methods. Extensive use of CASE tools.

Early 1990s. Object-oriented development became a mainstream development technique.

En definitiva...

La INS es algo más que programar

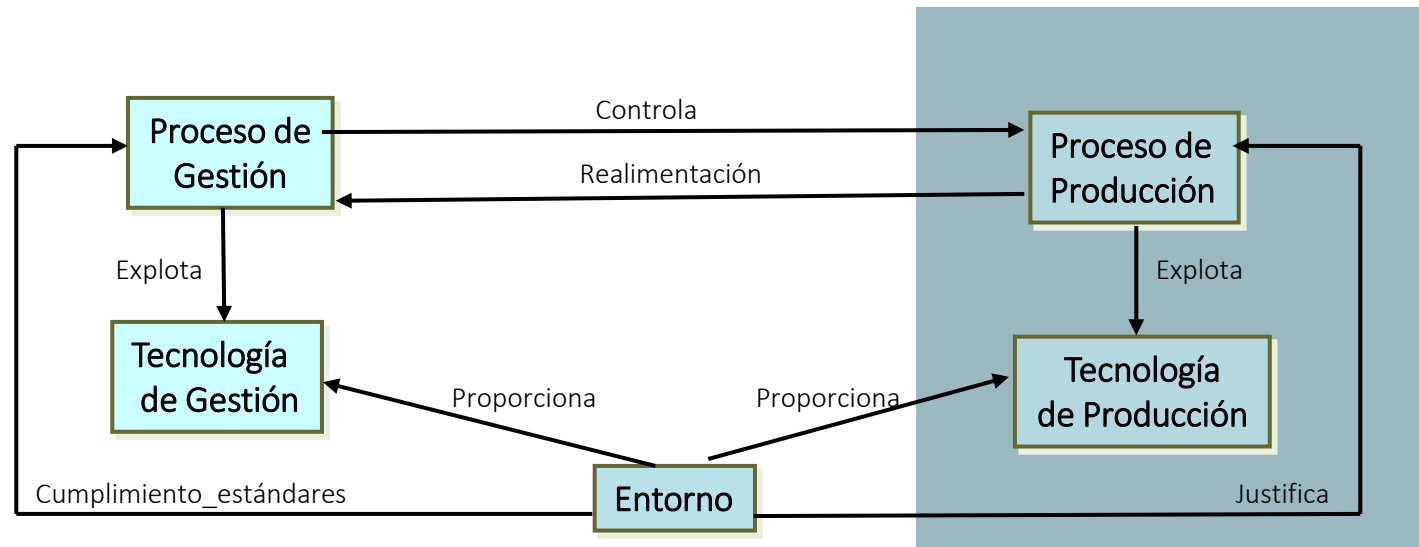
El proceso de la INS comienza bastante antes de escribir líneas de código y continua después de que la primera versión del producto haya sido completada

Son cruciales la planificación, el seguimiento y el control rigurosos de los proyectos software

El proceso del software

El Proceso del Software

Establece un marco para el desarrollo de software



El Proceso de Desarrollo

Conjunto de actividades cuya meta es el desarrollo o evolución de software

Conocido también como **Ciclo de Vida**

Actividades genéricas que aparecen siempre:

Especificación

Desarrollo

Validación

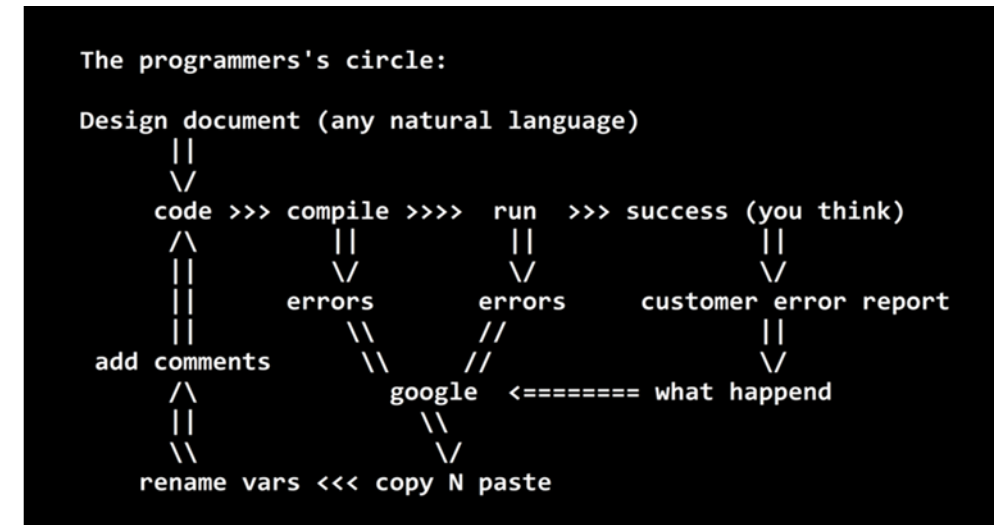
Evolución

Modelos de Ciclo de Vida

Codificar y corregir (*code-and-fix*)

Clásico o en cascada

Clásico con prototipado



Programación Automática



Modelos evolutivos

Incremental

En espiral

Modelos de Ciclo de Vida

Codificar y corregir (*code-and-fix*)

Clásico o en cascada

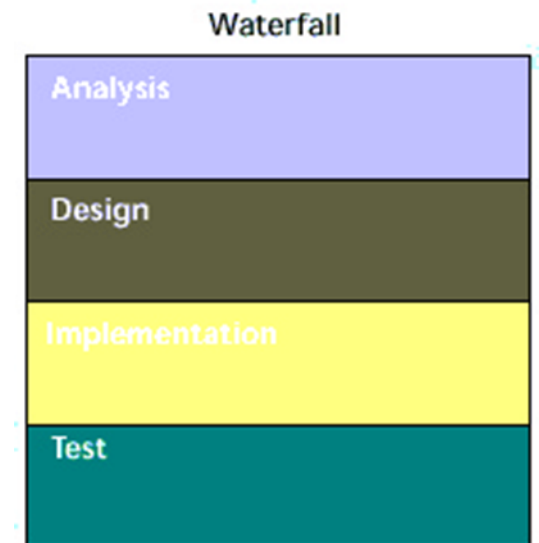
Clásico con prototipado

Programación Automática

Modelos evolutivos

Incremental

En espiral



Modelos de Ciclo de Vida

Codificar y corregir (*code-and-fix*)

Clásico o en cascada

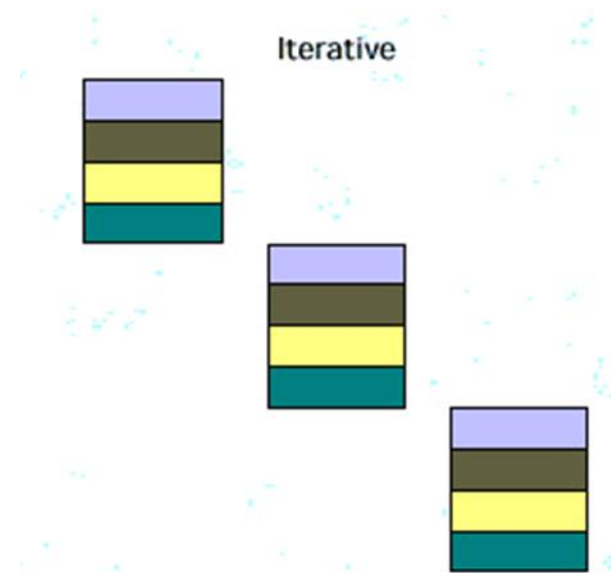
Clásico con prototipado

Programación Automática

Modelos evolutivos

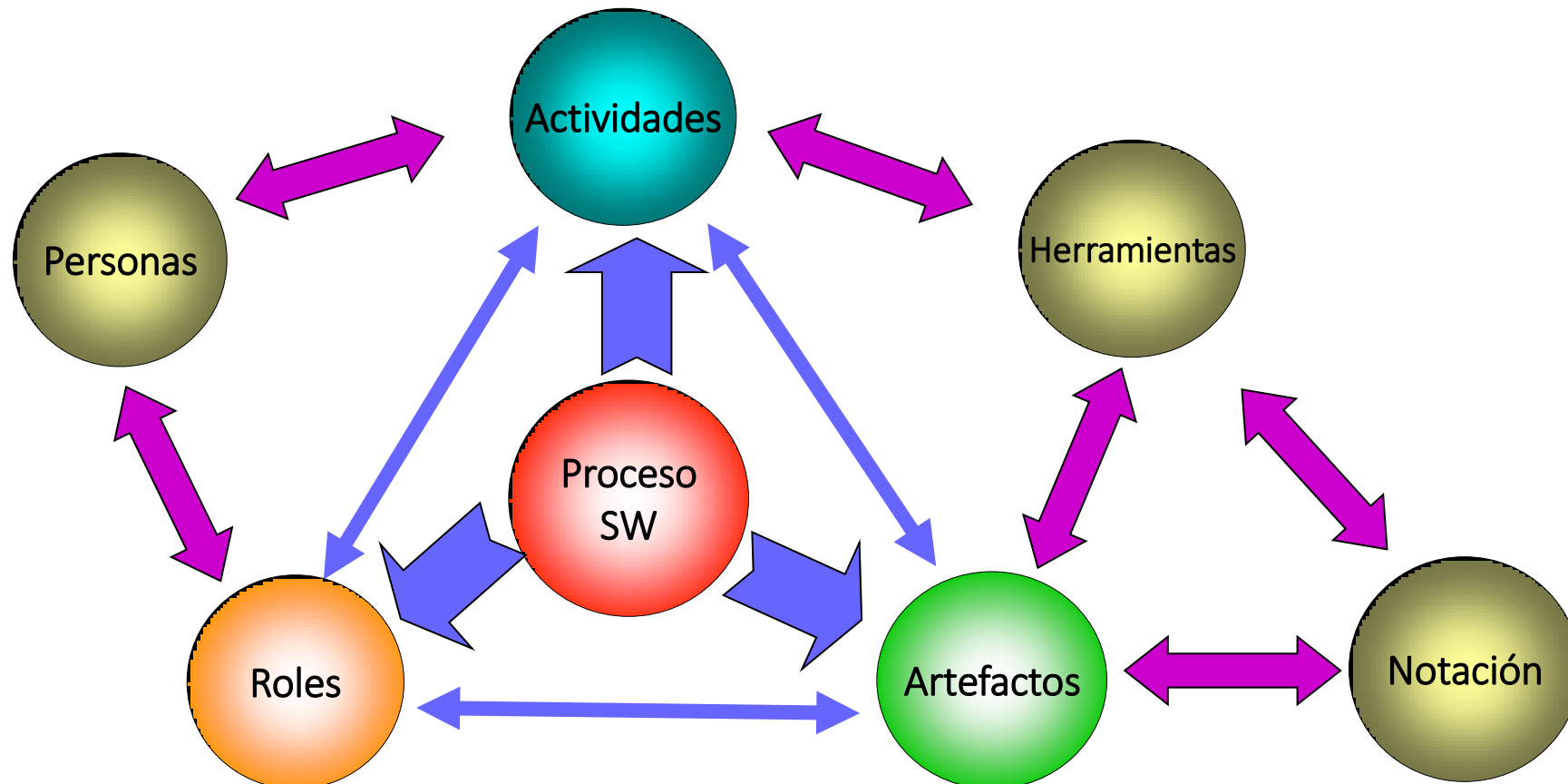
Incremental

En espiral



Metodología

En un proyecto de desarrollo de software, la metodología define: Quién / Qué / Cómo / Cuándo



Metodología

No existe una metodología universal de desarrollo de software

Metodologías estructuradas

Metodologías orientadas a objetos

Métodologías ágiles

Las Metodologías Ágiles valoran...

... al individuo y las interacciones en el equipo de desarrollo más que a las actividades y las herramientas

... desarrollar software que funciona, más que conseguir una buena documentación \Rightarrow Minimalismo respecto del modelado y la documentación del sistema

... la colaboración con el cliente más que la negociación de un contrato

... responder a los cambios más que seguir estrictamente una planificación

<http://www.agilealliance.com>

Principios de las Metodologías Ágiles (1/4)

La prioridad principal es satisfacer al cliente mediante tempranas y continuas entregas de software utilizable.

Dar la bienvenida a los cambios. Los procesos ágiles aplican los cambios para que el cliente sea competitivo.

Entregar el software desarrollado frecuentemente con el menor intervalo de tiempo posible entre una entrega y la siguiente.

Principios de las Metodologías Ágiles (2/4)

La gente de negocios y los desarrolladores trabajan juntos a través de un proyecto

Construir proyecto empujados por motivaciones personales. Dar el entorno que necesitan las personas y confiar en ellos.

Principios de las Metodologías Ágiles

(3/4)

El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.

Desarrollar software es la primera medida de progreso.

Los procesos ágiles promueven un desarrollo llevadero. Los patrocinadores, desarrolladores y usuarios son capaces de mantener una paz constante.

Principios de las Metodologías Ágiles

(4/4)

La atención continua a la calidad técnica y al buen diseño incrementa la agilidad

La simplicidad es esencial

Las mejores arquitecturas, requisitos y diseños surgen de la propia organización del equipo

En intervalos regulares, el equipo reflexiona en como llegar a ser más efectivo, sincronizar y ajustar su comportamiento.

Metodologías Ágiles vs. “pesadas” (1/2)

Metodología Ágil	Metodología No Ágil
El cliente es parte del equipo de desarrollo (además <i>in-situ</i>)	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (< 10 integrantes) y trabajando en el mismo sitio	Grupos grandes
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura	La arquitectura es esencial

Metodologías Ágiles vs. “pesadas” (2/2)

Metodología Ágil	Metodología No Ágil
Tolerante a los cambios	Resistente a los cambios
Impuestas internamente (equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe un contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado

Principales Metodologías Ágiles

Extreme Programming (XP) <http://www.extremeprogramming.org>

SCRUM <http://www.controlchaos.com>

Crystal Methods <http://alistair.cockburn.us/Crystal+methodologies>

Adaptive Development Software (ADS) <http://www.adaptivesd.com>

Dynamic Systems Development Method (DSDM) <http://www.dsdm.org>

Feature-Driven Development (FDD) <http://www.featuredrivendevelopment.com>

Lean Development (LD) <http://www.poppendieck.com>



Extreme Programming (XP)

Kent Beck, Ward Cunningham y Ron Jeffries

www.extremeprogramming.org

www.xprogramming.com

Diseñado para entornos dinámicos

Pensado para equipos pequeños (≤ 10 programadores)

Orientado fuertemente hacia la codificación

Énfasis en la comunicación informal, verbal

Otros valores: simplicidad, realimentación y coraje

XP

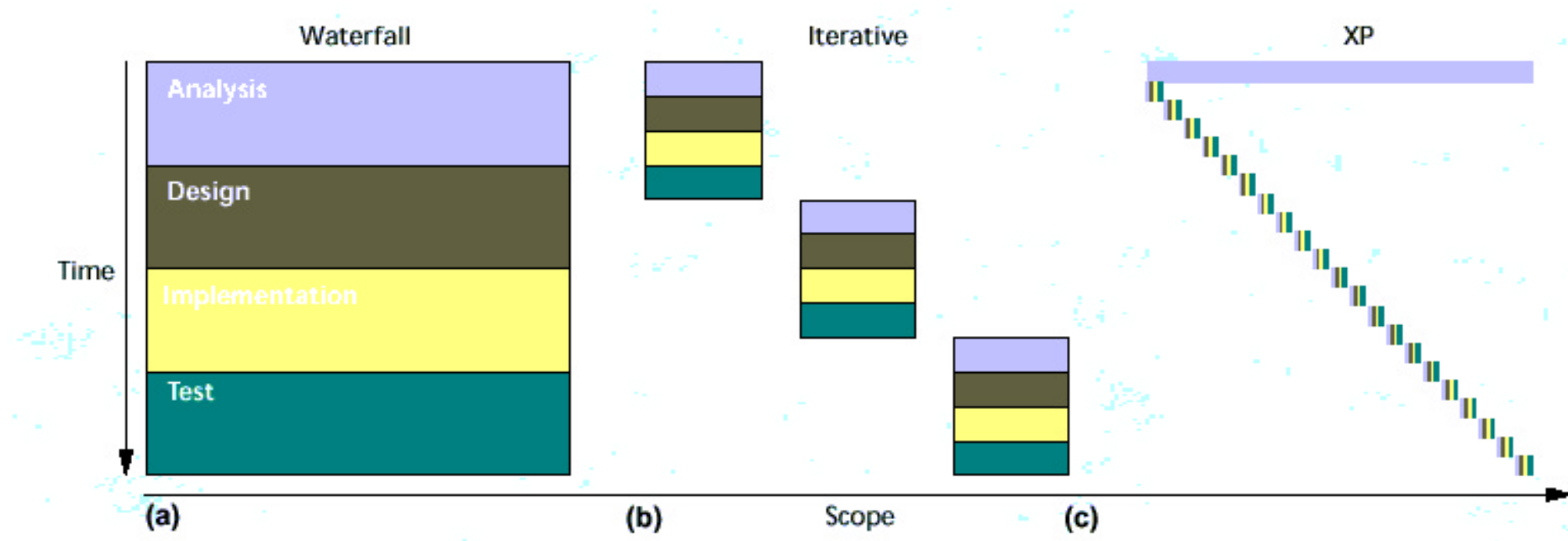
Historias, Iteraciones, Versiones, Tareas y Casos de Prueba

El cliente selecciona la **siguiente versión** a construir, eligiendo las **características funcionales** que considera más valiosas (llamadas **Historias**) de un conjunto posible de historias, siendo informado de los **costes** y del **tiempo** que costará su implementación.

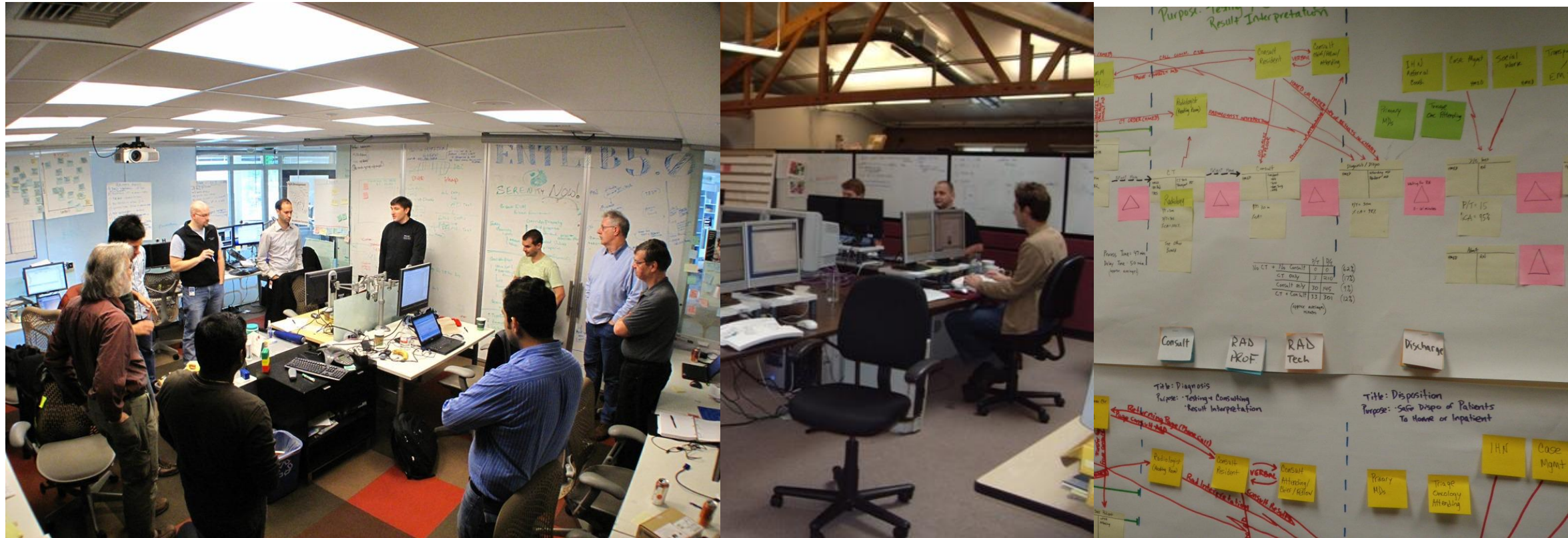
Los programadores **convierten las historias** en **tareas a realizar** y a continuación convierten las **tareas** en un **conjunto de casos de prueba** para demostrar que las tareas se han finalizado.

Trabajando con un compañero el programador **ejecuta los casos de prueba_y evoluciona el diseño** intentando mantener su simplicidad.

XP vs. metodologías clásicas



Un día en la vida de un proyecto ágil



Herramientas

The screenshot displays the Microsoft Teams interface for the 'ProjectSoftware Team'. The top navigation bar includes 'Team Services / ProjectSoftware / ProjectSoftware Team' and the user 'Antonio Garrido'. The main menu on the left shows 'Backlogs' and 'Queries'. The 'Backlogs' section is expanded, showing 'Current' and 'Future' iterations. The 'Current' iteration is 'Iteration 1. Análisis'. The 'Future' iterations are 'Iteration 2. Primera versión' and 'Iteration 3. Segunda versión...'. The 'Iteration 1. Análisis' backlog is selected, showing a list of tasks. The tasks are: 'Visualizar los vehículos disponibles' (Antonio Garrido), 'Ver las características de cada vehículo' (margomez@dsic.u...), and 'Insertar el alquiler concreto' (emarzal@dsic.upv....). The tasks are in the 'New' state. The backlog is displayed in a 'Board' view, showing columns for 'New', 'Active', 'Resolved', and 'Closed'. The 'New' column contains three tasks: 'Realizar la conexión a la BD' (Antonio Garrido), 'Acceder a tabla para recuperar los vehículos' (M. mpenades), and 'Ver las características de cada vehículo' (margomez@dsic.u...). The 'Active' column is empty. The 'Resolved' and 'Closed' columns are also empty. A green callout box with the text 'Ejemplo de stories y tasks en la 1ª iteración' is overlaid on the 'Active' column.

Team Services / ProjectSoftware / ProjectSoftware Team

Antonio Garrido | Search work items

HOME CODE WORK BUILD TEST RELEASE

Backlogs Queries

Features

Stories

Current

Iteration 1. Análisis

Future

Iteration 2. Primera versión

Iteration 3. Segunda versión...

ProjectSoftware Team Iteration 1. Análisis

No iteration dates Set dates

Backlog Board Capacity

Group by Stories Person All

New Active Resolved Closed

Visualizar los vehículos disponibles

Antonio Garrido

State New

Realizar la conexión a la BD

Antonio Garrido

Acceder a tabla para recuperar los vehículos

M. mpenades

Ver las características de cada vehículo

margomez@dsic.u...

State New

Insertar el alquiler concreto

emarzal@dsic.upv....

State New

Ejemplo de stories y tasks en la 1ª iteración

XP: prácticas

El juego de la Planificación

Entregas Pequeñas

Semanas de 40 horas

Diseño Sencillo

Programación en Parejas

Pruebas
Propiedad Colectiva

Metáfora

Refactorización

El Cliente siempre con el Desarrollador

Integración Continua

Estándares de Codificación

¿Funcionan los métodos ágiles?

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL				
SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011-2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

La ¿perenne? crisis del software

Informe CHAOS ...

RESOLUTION						Project resolution results from CHAOS research for years 2004 to 2012.
	2004	2006	2008	2010	2012	
Successful	29%	35%	32%	37%	39%	
Failed	18%	19%	24%	21%	18%	
Challenged	53%	46%	44%	42%	43%	

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>

Agilismo at Work



<http://agilismoatwork.blogspot.com.es/>

Bibliografía básica

- 📖 Sommerville, I. Ingeniería del Software. (8ª ed.). Addison-Wesley, 2008
- 📖 Presman, R.S., Ingeniería del Software: un enfoque práctico (6ª ed.), McGraw-Hill, 2005
- 📖 Royce, W.W., Managing the Development of Large Software Systems: Concepts and Techniques. Proc WESCON, 1970
- 📖 Agresti, W.W. Tutorial: New Paradigms for Software Development. IEEE Computer Society Press, 1986
- 📖 Balzer, R., Cheatman, T.E. and Green, C., Software Technology in the 1990's: Using a New Paradigm, IEEE Computer, Nov. 1983, pp. 39-45
- 📖 McDermid, J. And Rook, P., Software Development Process Models. Software Engineer's Reference Book, CRC Press, 1993
- 📖 Boehm, B.W.. A Spiral Model of Software Development and Enhancement, IEEE Computer, pages 61-72, May 1988.

Bibliografía metodologías

- 📖 Krutchen, P., *The Rational Unified Process- An Introduction*. Addison –Wesley, 1998
- 📖 Jacobson, G. Booch and J. Rumbaugh., *The Unified Software Development Process*, Addison-Wesley, 1999
- 📖 Beck, K., *Extreme Programming Explained: Embrace Change*. The XP Series. Addison-Wesley, 2000