

# Introducción al Proceso del Software

## Unidad 1

Patricio Letelier

letelier@dsic.upv.es

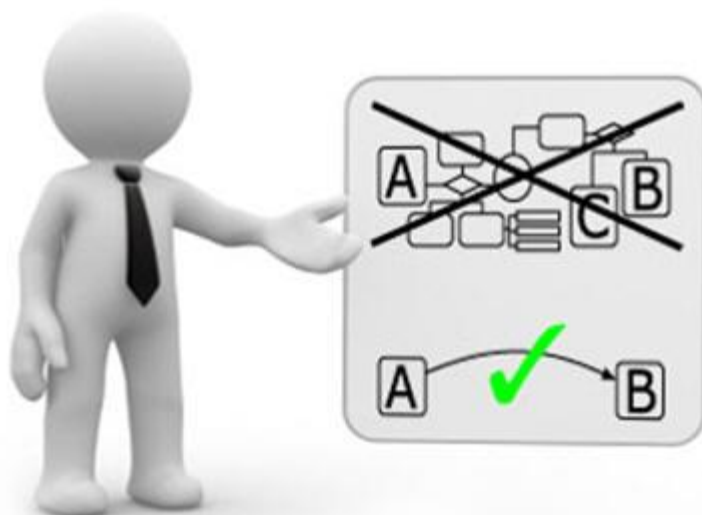


UNIVERSIDAD  
POLITECNICA  
DE VALENCIA

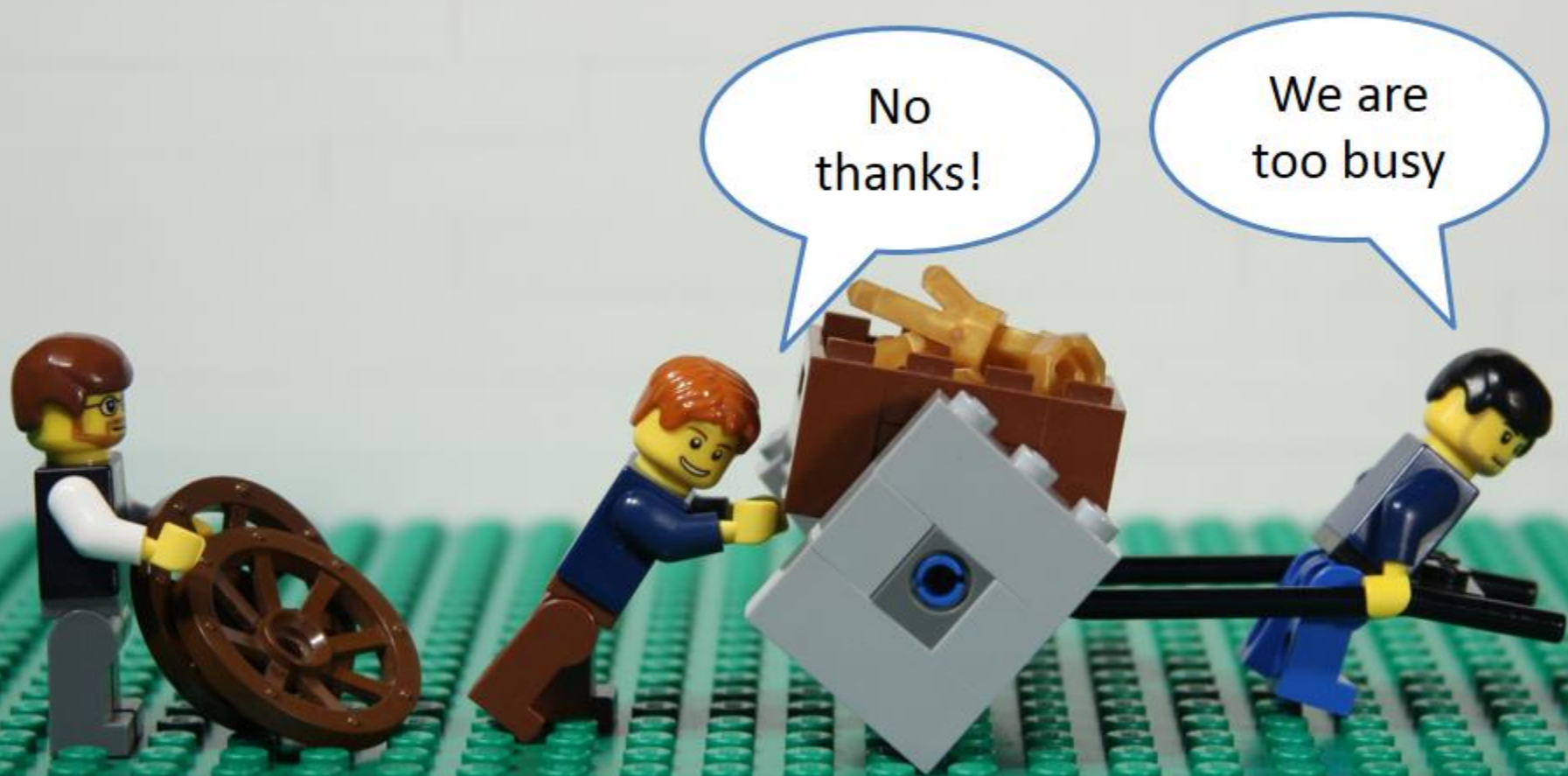


# Contexto:

Mejora del procesos de producción,  
mantenimiento y soporte.



# Are you too busy to improve?



¿De qué estamos hablando?

# Más x Menos



# ¿Cuál es nuestra noción de éxito (de un proyecto)?



“Triángulo de la gestión de proyectos” o “Triángulo de hierro”

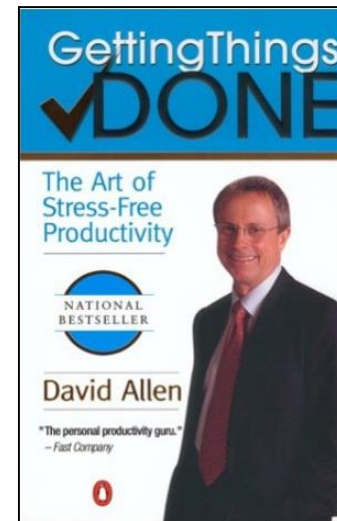
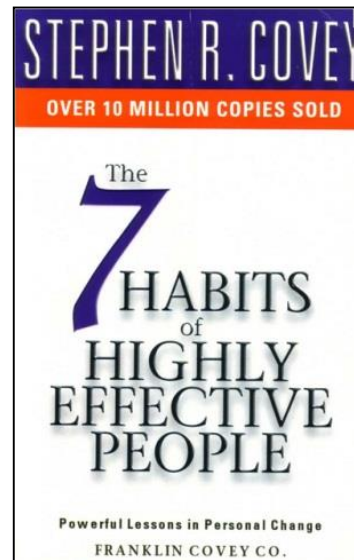
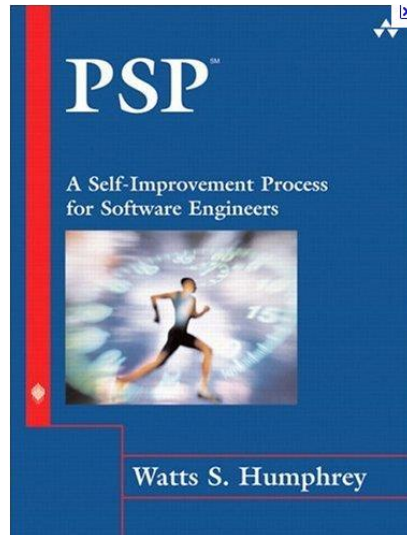
## ... Otra visión del éxito



# Un paréntesis: Productividad Personal

Indudablemente, la productividad del equipo también se puede mejorar centrándose en mejorar la productividad de cada uno de sus miembros

→ Disciplina y hábitos individuales de trabajo



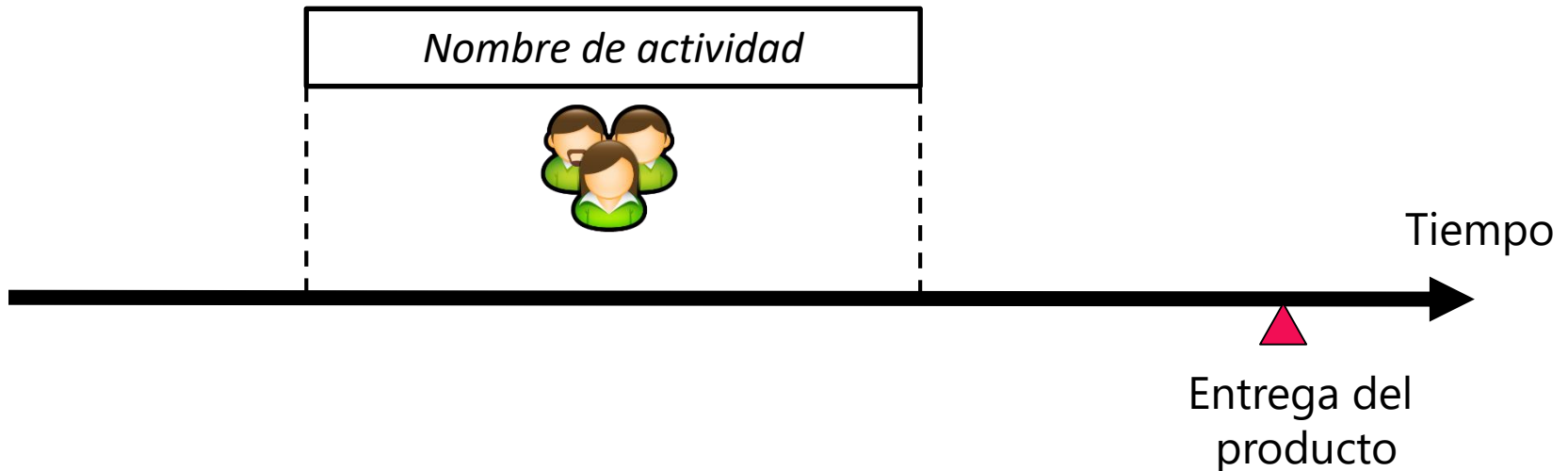


# ¿Cómo desarrollarías software?

**¿Cuándo deberían comenzar y terminar?**

**¿Qué actividades deberían realizarse?**

**¿Quién(es) debería(n) realizarlas?**





# Modelos de Proceso y Metodologías

- Aportan el carácter de disciplina a la **Ingeniería de Software**
- Un modelo de proceso de software es una estrategia global respecto de cómo abordar un proyecto de desarrollo de software
- Modelos de proceso de software:
  - Codificar y corregir (code-and-fix)
  - Desarrollo en cascada
  - Desarrollo evolutivo
    - Desarrollo incremental
    - Desarrollo en espiral
  - Desarrollo formal de sistemas
  - Desarrollo basado en reutilización

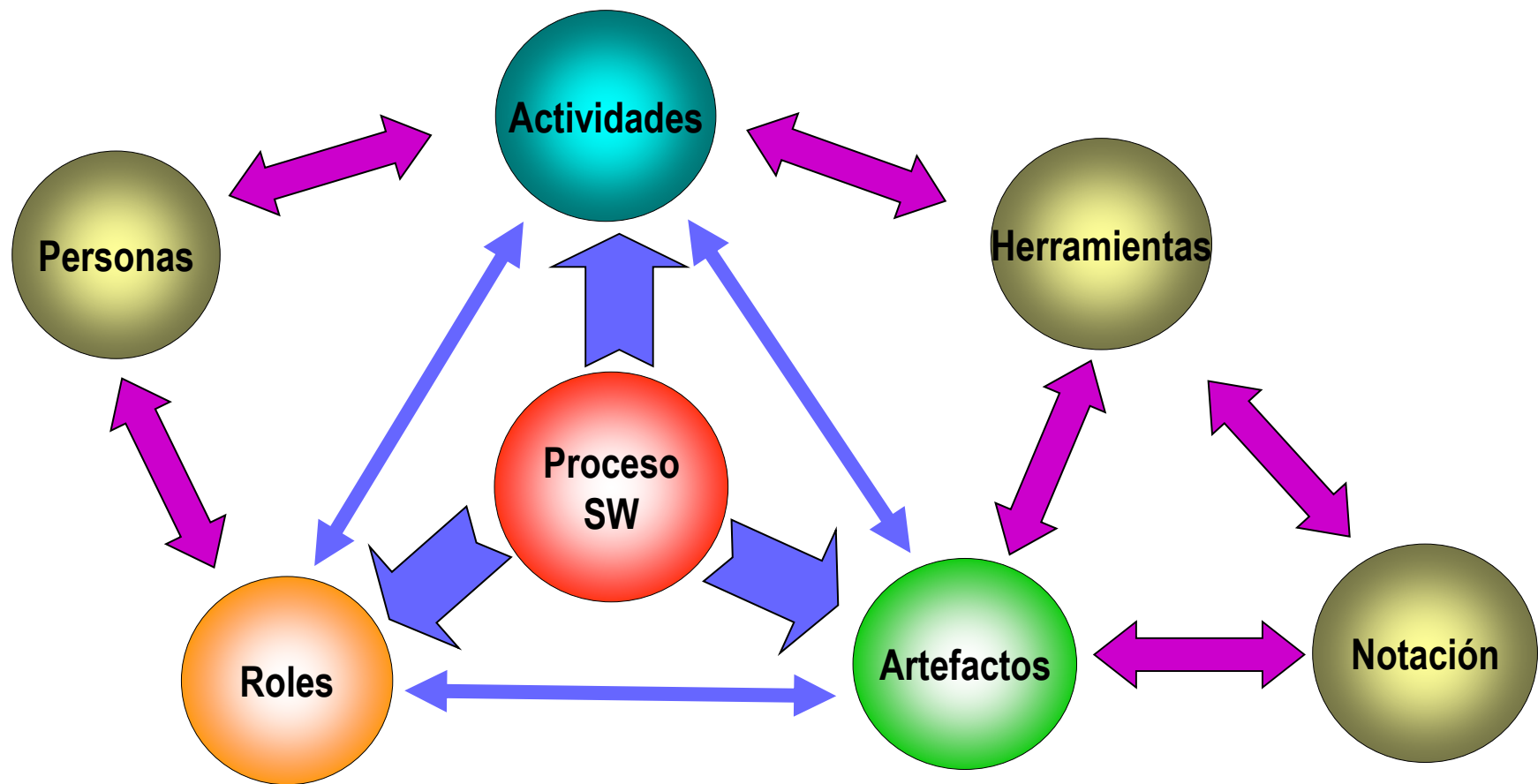
<http://agilismoatwork.blogspot.com.es/2014/10/modelos-de-proceso-para-desarrollo-agil.html>

# ¿Qué es una metodología?

- Una metodología define **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo



# Elementos de una Metodología



# Clasificación de Metodologías

- Las metodologías se basan en alguna combinación de modelos de proceso.
- Desde la perspectiva de las técnicas utilizadas para las actividades de análisis, diseño e implementación las metodologías pueden ser clasificadas como: Metodologías Estructuradas o Metodologías Orientadas a Objetos.
- Desde la perspectiva de las prácticas utilizadas las metodologías pueden ser clasificadas como: Metodologías Tradicionales o Metodologías Ágiles.

# Metodologías Estructuradas

- Los métodos estructurados comenzaron a desarrollarse a fines de los 70's con la Programación Estructurada, luego a mediados de los 70's aparecieron técnicas para el Diseño Estructurado primero y luego para el Análisis Estructurado.
- Ejemplos de metodologías estructuradas gubernamentales: [MERISE](#) (Francia), [MÉTRICA\\_3](#) (España), [SSADM](#) (Reino Unido).
- Ejemplos de métodos estructurados en el ámbito académico: [Gane & Sarson](#), [Ward & Mellor](#), [Yourdon & DeMarco](#) e [Information Engineering](#).

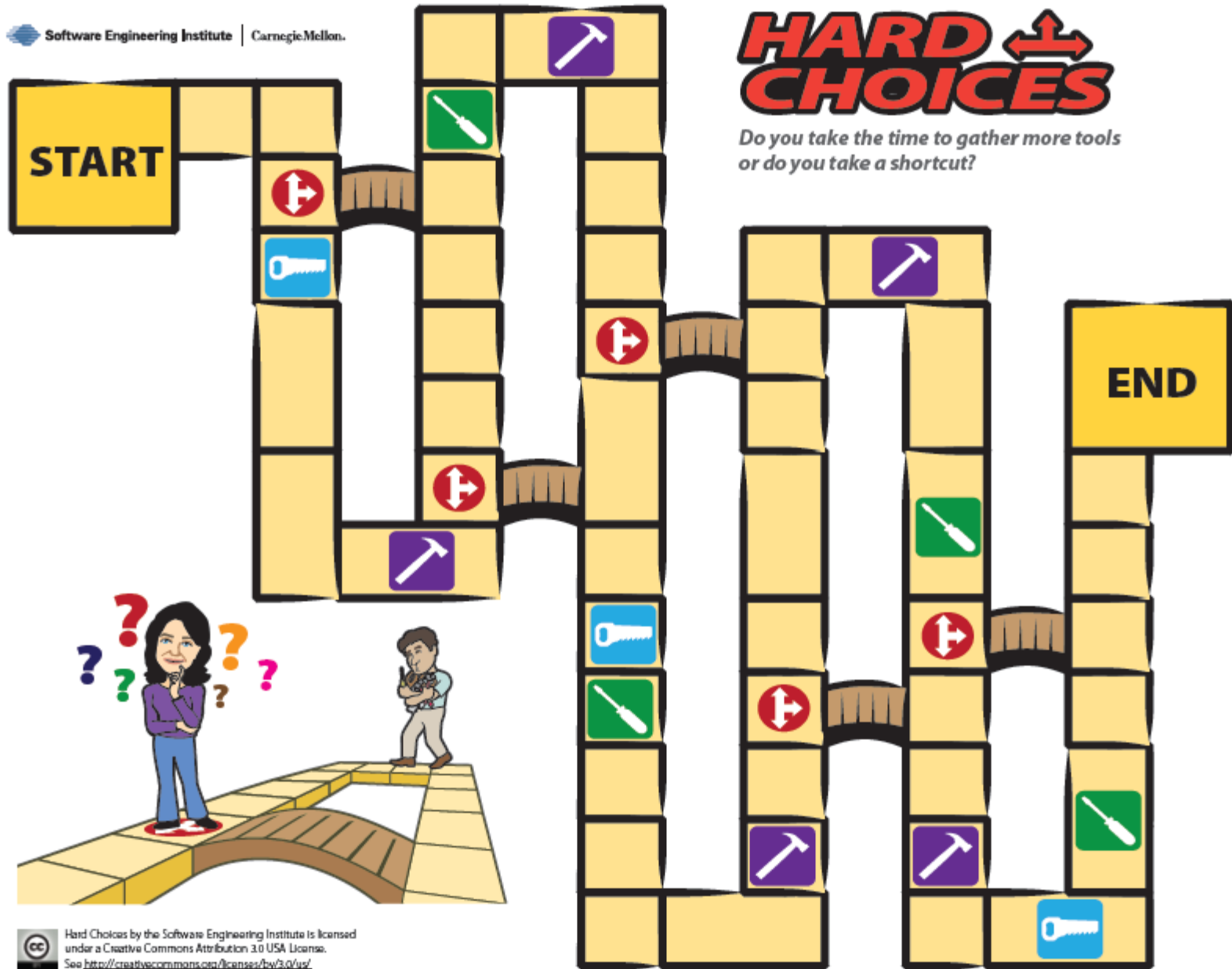
# Metodologías Orientadas a Objetos (OO)

- Su historia va unida a la evolución de los lenguajes de programación orientada a objeto. En los 60's [SIMULA](#), en los 70's [Smalltalk-80](#), la primera versión de [C++](#) por Bjarne Stroustrup en 1981 y actualmente [Java](#), [C#](#) y [otros](#). A fines de los 80's comenzaron a consolidarse algunos métodos Orientadas a Objeto.
- En 1995 aparece el Método Unificado, que posteriormente se reorienta para dar lugar al Unified Modeling Language (UML), la notación OO más popular en la actualidad.
- Métodos OO con notaciones previas a UML: [OOAD \(Booch\)](#), [OOSE \(Jacobson\)](#), [Coad & Yourdon](#), [Shaler & Mellor](#) y [OMT \(Rumbaugh\)](#). Metodologías orientadas a objetos basadas en UML: [Rational Unified Process \(RUP\)](#), [OPEN](#), [MÉTRICA 3](#).

# "Cambia el chip"







Hard Choices by the Software Engineering Institute is licensed under a Creative Commons Attribution 3.0 USA License. See <http://creativecommons.org/licenses/by/3.0/us/>

# The Hard Choices Game

- Technical debt (deuda técnica)
- Condiciones cambiantes e impedimentos
- Estrategia individual y colectiva

# Ágil v/s Tradicional

