



INGENIERÍA DE SOFTWARE (IS)

INTRODUCCIÓN

Gabriel Pedraza Ferreira

¿Qué es ingeniería de software?

¿Por qué es necesaria la ingeniería de software?

<https://www.youtube.com/watch?v=gBd-ct58DCI>

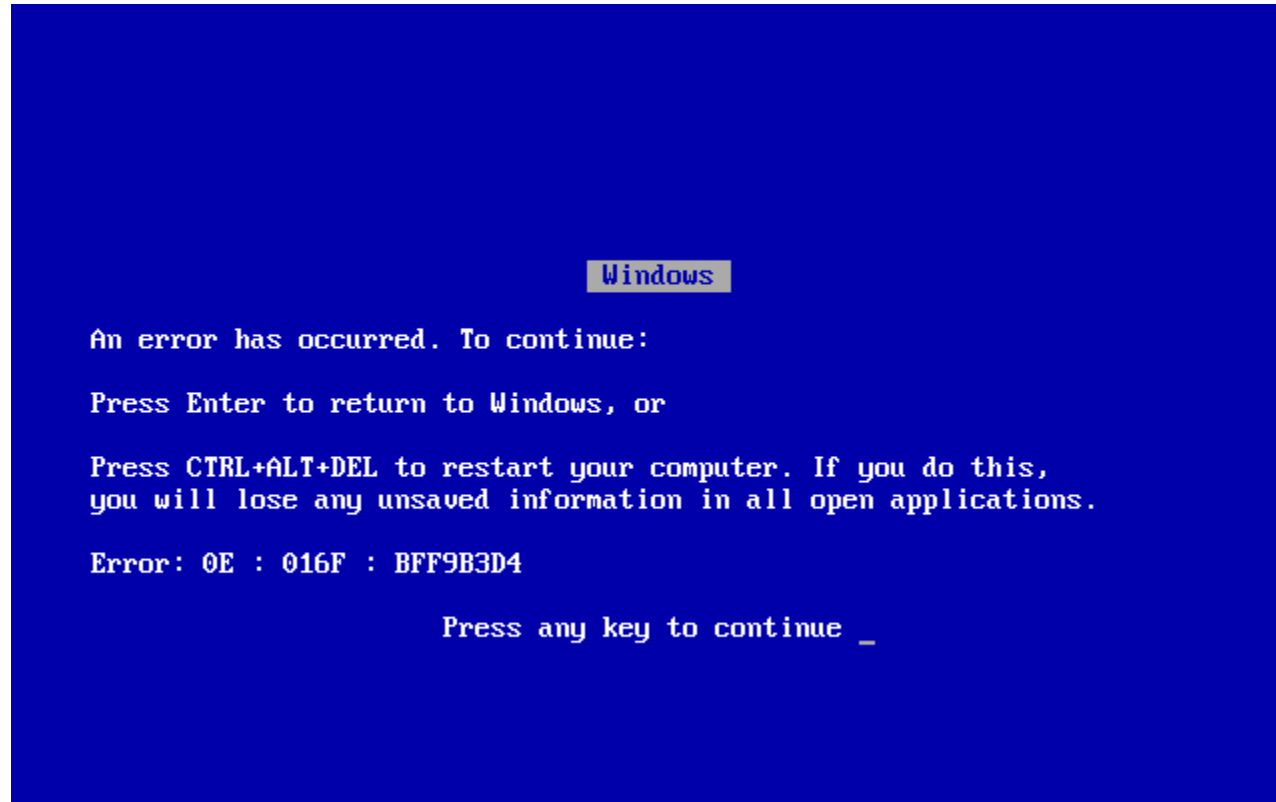




- 40 segundos en el aire
- 10 años de investigación y desarrollo
- 7.000 millones de dólares
- 500 millones de dólares a bordo



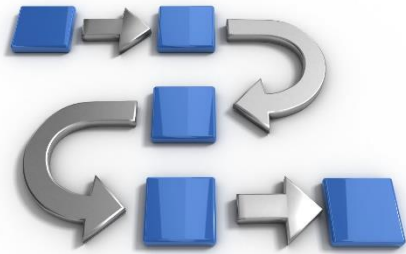
CRASH



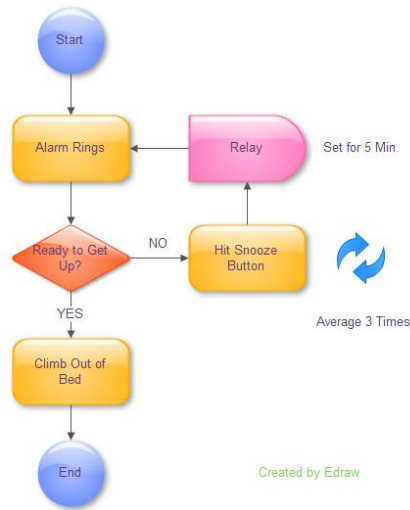
¿Por qué es tan difícil construir (buen) software?



INGENIERÍA DE SOFTWARE



Metodologías



Técnicas



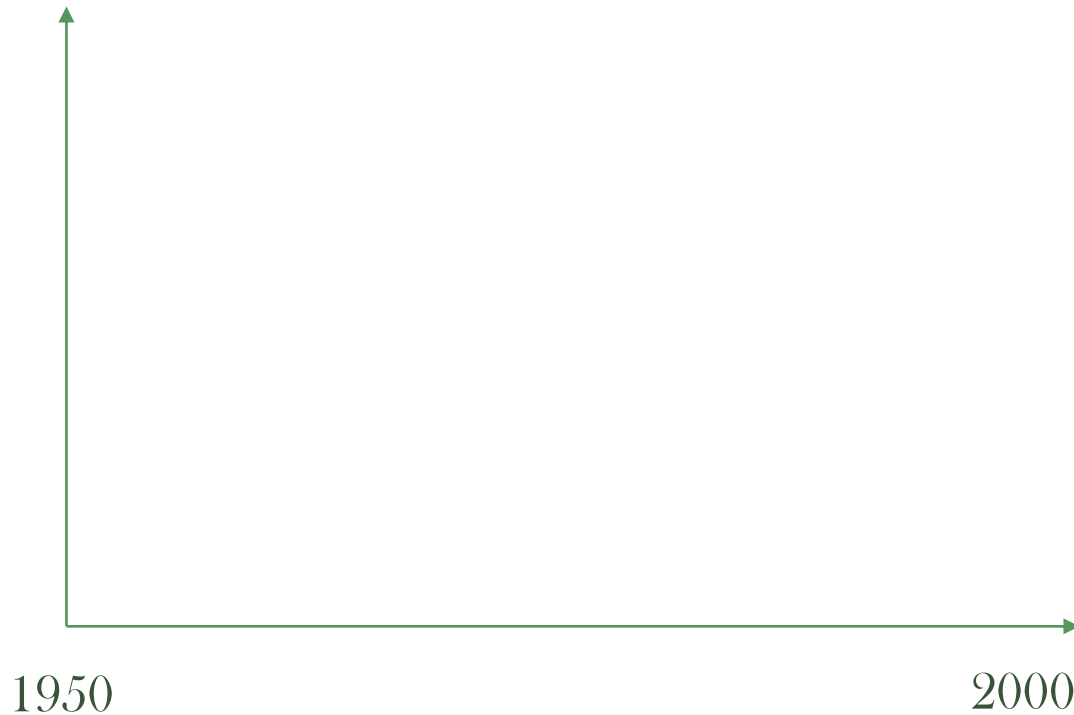
Herramientas

Software de alta calidad que funciona y cumple con el presupuesto!



CRISIS DEL SOFTWARE – ORIGEN 60'S

Hardware => Software



Demanda de software aumenta exponencialmente

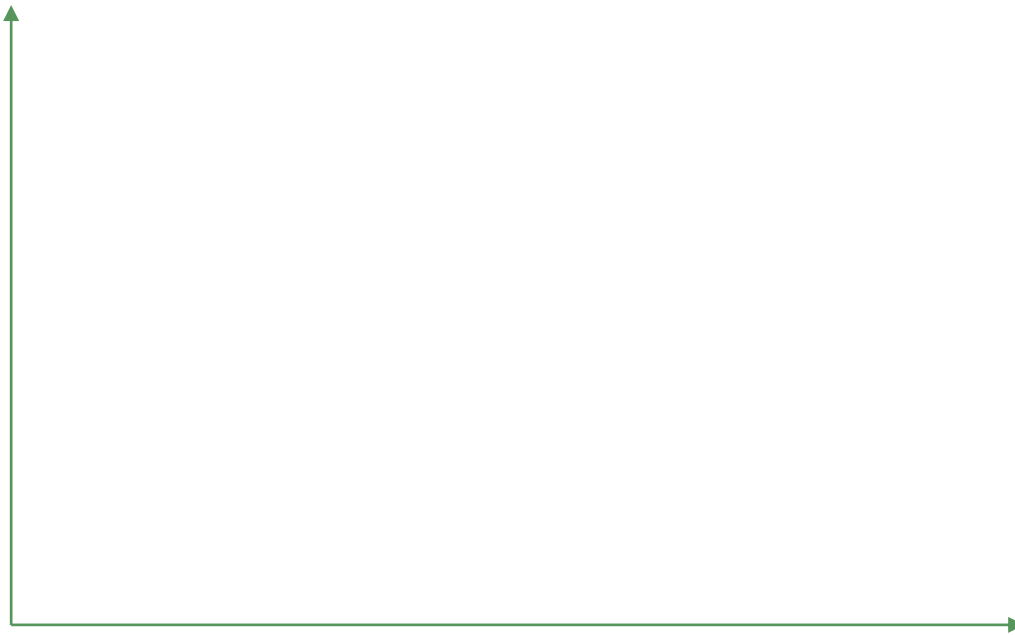
CRISIS DEL SOFTWARE

Tamaño	Proyecto
10^2 LOC	Ejercicio de clase
10^3 LOC	Proyecto Pequeño
10^4 LOC	Proyecto Semestre
10^5 LOC	Procesador de texto
10^6 LOC	Sistema Operativo
10^7 LOC	Sistema Distribuido

Esfuerzo de desarrollo



CRISIS DEL SOFTWARE



Crecimiento de la productividad del desarrollador



TAMAÑO DEL SOFTWARE

2012

A DVD Player contiene aproximadamente 50 millones de líneas de código.

Un carro BMW tiene 50 dispositivos conectados en red.

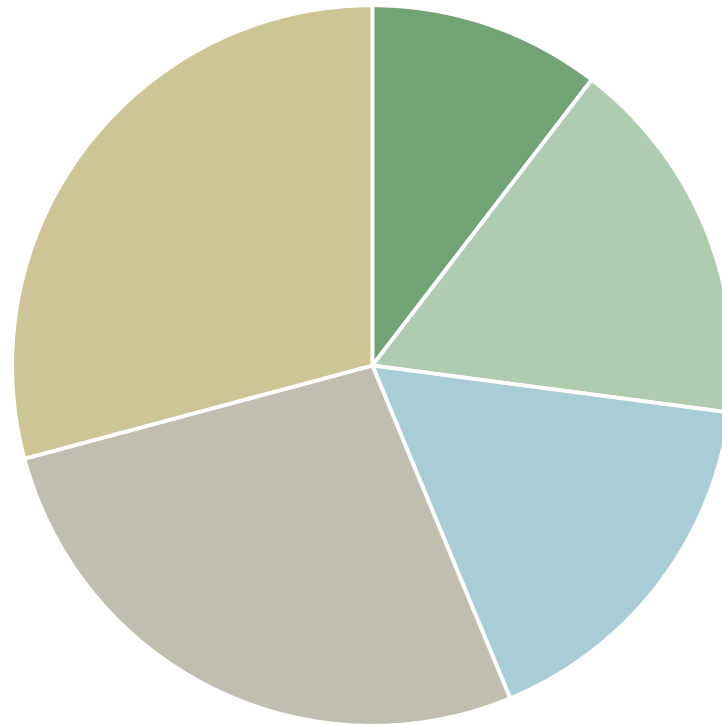
Eclipse IDE contiene 4 millones de líneas de código

El Kernel de Linux contiene 15 millones de líneas de código

Cada 7 años la cantidad de software se duplica
<http://users.jyu.fi/~koskinen/smcosts.htm>

ESTUDIO DE 9 CONTRATOS SOFTWARE (DAVIS, 1990)

Proyectos Software



■ Utilizable

■ Con cambios

■ Con cambios Complejos

■ Nunca utilizado

■ Nunca entregado



INGENIERÍA DE SOFTWARE

- “Software engineering is the study and an application of engineering to the design, development, and maintenance of software” (ACM)
- “The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software“ (IEEE)
- “An engineering discipline that is concerned with all aspects of software production” (Sommerville)
- “The establishment and use of sound engineering principles in order to economically obtain software that is reliable and works efficiently on real machines” (North-Holland)



Conclusion



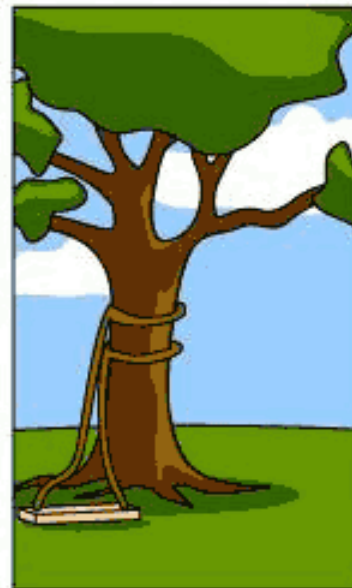
How the customer explained it



How the Project Leader understood it



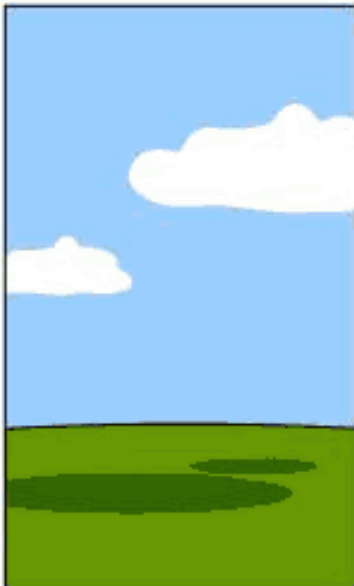
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



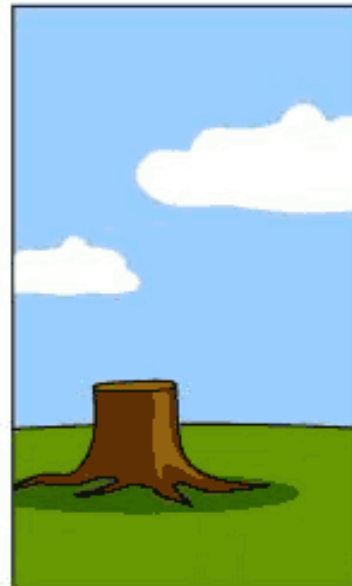
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed