fiuba

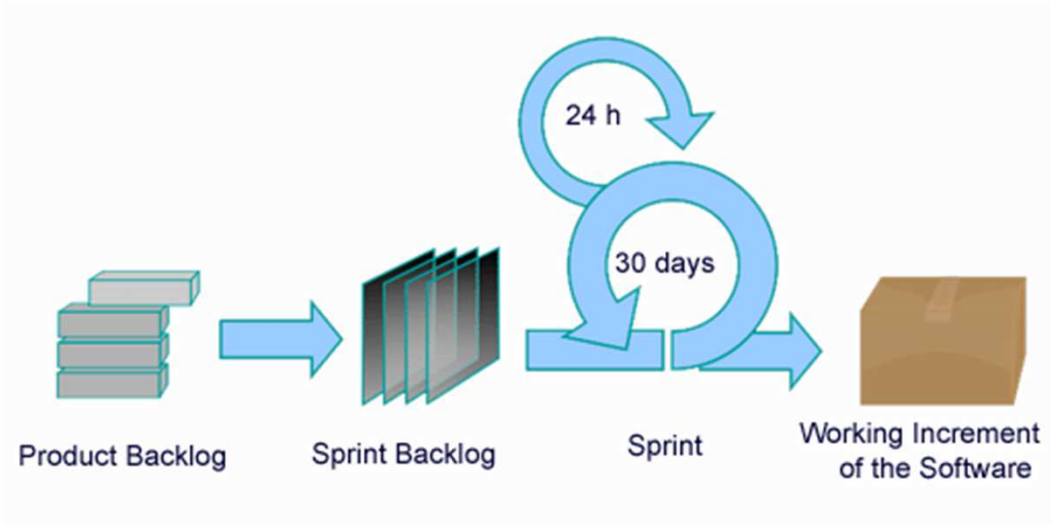
algo3

Metodología Desarrollo de software

Carlos Fontela

cfontela@fi.uba.ar

Metodología y Desarrollo



Temario

Metodología y tipos de métodos

Disciplinas del desarrollo

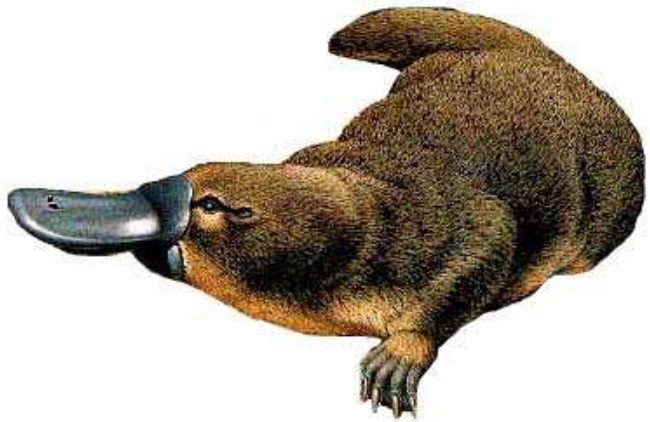
Problemas de los proyectos de desarrollo de software

fiuba algo3

¿Mitos o hechos?

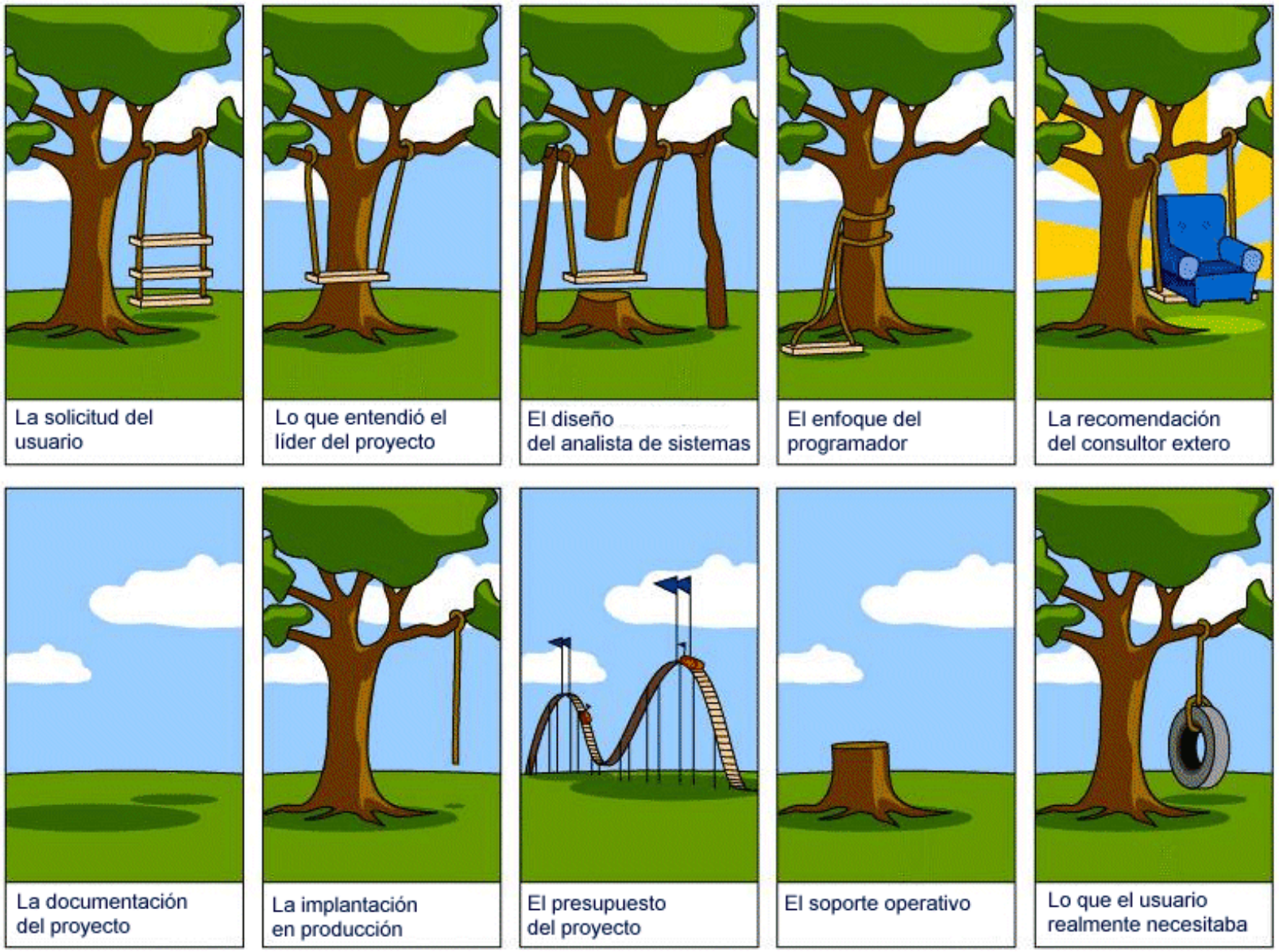
Desarrollo de software es lo mismo que programación

Entre las ingenierías, la de software es la de peor reputación en cuanto a la satisfacción de los clientes, al cumplimiento de plazos y presupuestos



fiuba algo3

Metodología



fiuba algo3

Método o proceso

Sirve para estructurar, planificar, desarrollar y controlar el desarrollo de software

Determina

- Fases
- Roles
- Actividades
- Artefactos
- Etc.

Categorías de métodos (1)

- Cascada
  - Distribución en el tiempo basada en actividades
    - Análisis, diseño, programación, pruebas, …

- Incrementales
  - Distribución en el tiempo basada en el desarrollo y entrega de funcionalidades
    - Entrega I, entrega II, etc.

fiuba algo3

Categorías de métodos (2)

- Procesos predictivos (¿basados en planes?)
  - Planificación más detallada y rígida
  - Se utilizan en grandes proyectos: suelen ser inaceptablemente pesadas para sistemas pequeños o medianos
  - Destacan el Proceso Unificado (UP), TSP, Cleanroom
- Métodos ágiles o adaptables
  - Más abiertos a los cambios
  - Permiten organizar desarrollos medianos sin caer en burocracias paralizantes
  - Nacieron como alternativa a carecer de metodología
  - Destacan Extreme Programming (XP) y Scrum

fiuba algo3

Ágiles vs. predictivos

Equipos pequeños y requisitos cambiantes => Ágiles

Equipos grandes y requisitos estables => Predictivos

fiuba algo3

Métodos ágiles

Evidencia

- El desarrollo de software es inherentemente cambiante
- => Aceptar el cambio, no gerenciarlo

Variables a controlar

- calidad, costo, tiempo de desarrollo, alcance
- => ajustar cualquier variable (¿alcance?) menos la calidad

Objetivos

- Bajar el riesgo
- Permitir cambios de especificaciones durante el desarrollo
- Favorecer la comunicación con el cliente

Hay un “Manifiesto ágil” =>

fiuba algo3

Manifiesto ágil
(http://www.agilemanifesto.org/iso/es/)

Estamos descubriendo formas mejores de desarrollar software tanto por nuestra propia experiencia como ayudando a terceros

A través de este trabajo hemos aprendido a valorar:

Individuos e interacciones sobre procesos y herramientas

Software funcionando sobre documentación extensiva

Colaboración con el cliente sobre negociación contractual

Respuesta ante el cambio sobre seguir un plan

Esto es, aunque valoramos los elementos de la derecha, valoramos más los de la izquierda.

fiuba algo3

Los métodos ágiles

- Extreme programming (XP), de Kent Beck y la comunidad Smalltalk
  - Lleva al extremo las buenas prácticas => es un conjunto de buenas prácticas
    - Lo analizamos acá, en un curso de Programación
- Scrum, de Ken Schwaber y Mike Beedle
  - Provee roles y artefactos centrados en seguimiento y control del proyecto
    - Lo van a analizar en materias de Administración de Proyectos
- Otros: ASD, Crystal Clear, FDD, DSDM, MSF for Agile

fiuba algo3

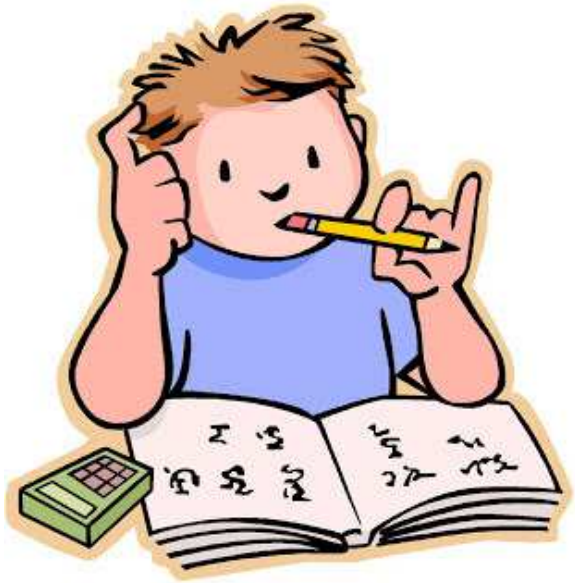
Recapitulación

fiuba algo3

Recapitulación: preguntas

¿Qué tipo de método de desarrollo (ágil/predictivo) aplicaría a un desarrollo con requerimientos estables y un equipo chico?

¿Qué tipo de método de desarrollo (ágil/predictivo) aplicaría a los proyectos que ha encarado?



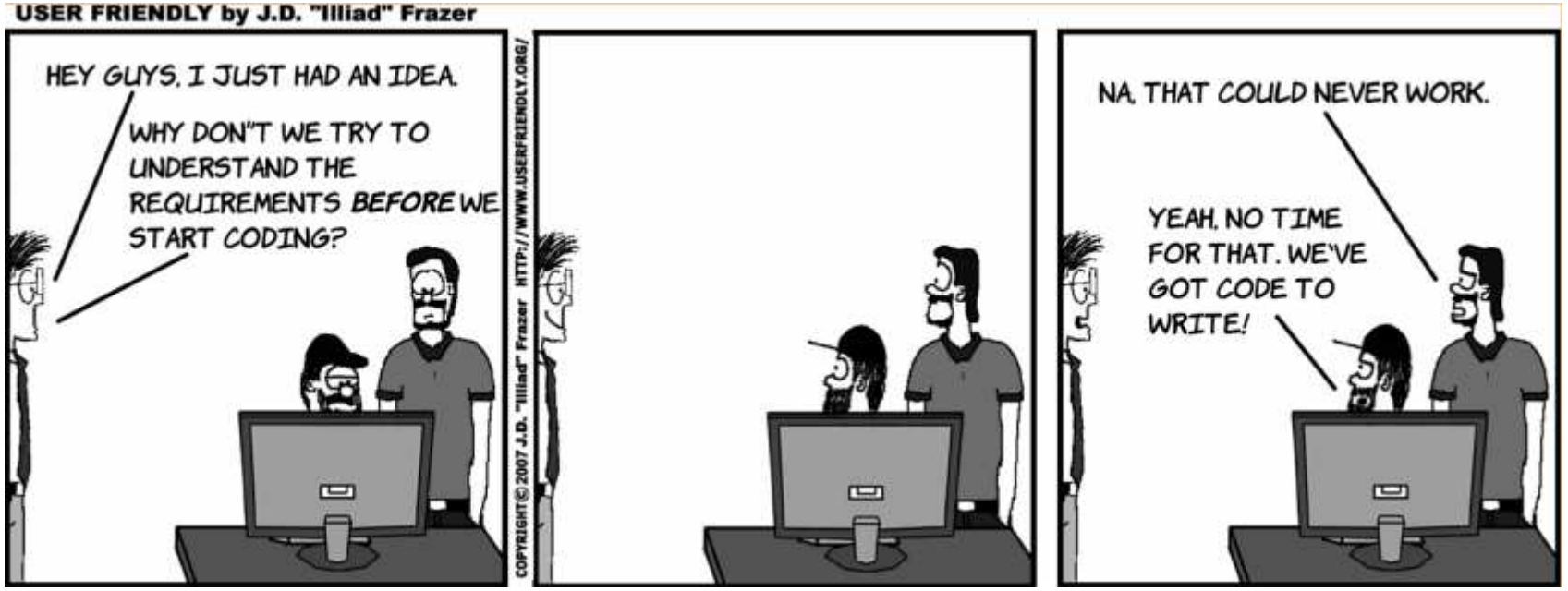
fiuba algo3

Desarrollo de software

Principal ocupación de los egresados de carreras informáticas de FIUBA

Desarrollo != Programación

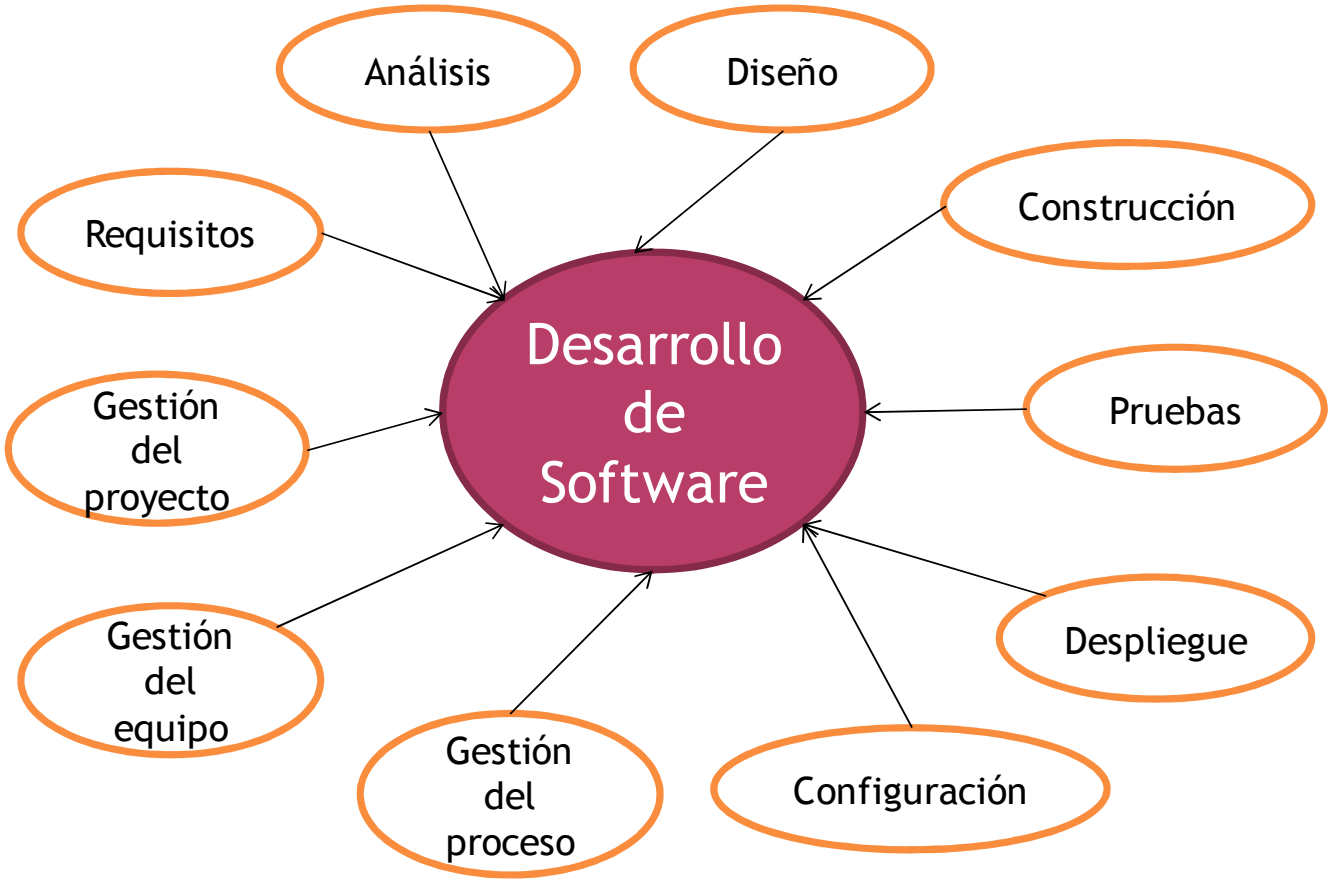
- Desarrollo de software incluye a la Programación
- Pero también a otras disciplinas



fiuba algo3

Disciplinas del desarrollo de software

No todo es programación en el desarrollo de software



fiuba algo3

Disciplinas del desarrollo (operativas)

Captura de requisitos: qué necesita el cliente

Análisis: qué vamos a construir

Diseño: cómo

Construcción o implementación

Pruebas: verificación y validación

Despliegue (en hardware)

fiuba algo3

Disciplinas del desarrollo (soporte)

Administración del proyecto, incluyendo seguimiento y control de tiempos y costos

Gestión de cambios

Administración de la configuración

Gestión de los recursos humanos y la comunicación

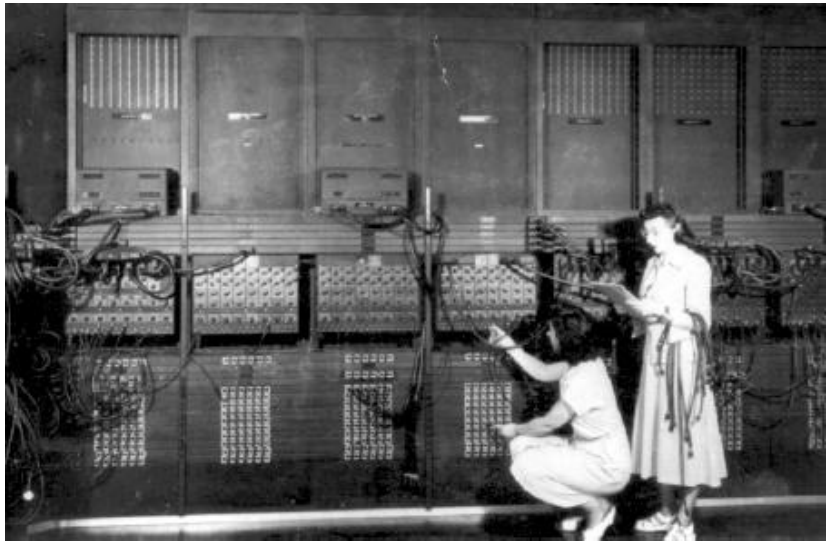
Gestión de riesgos

Gestión del proceso

fiuba algo3

Significado de programar

- 1945: ENIAC, programas cableados
- 1954: lenguajes de programación y grandes máquinas
- 1970s/80s: PC aisladas o en redes locales
- 1990s: redes amplias e Internet
- 2000s/10s: celulares, automóviles, tablets…
- Historia hacia la computación ubicua => impactos en programación



fiuba algo3

Significado de desarrollo

Hasta 1970: programar

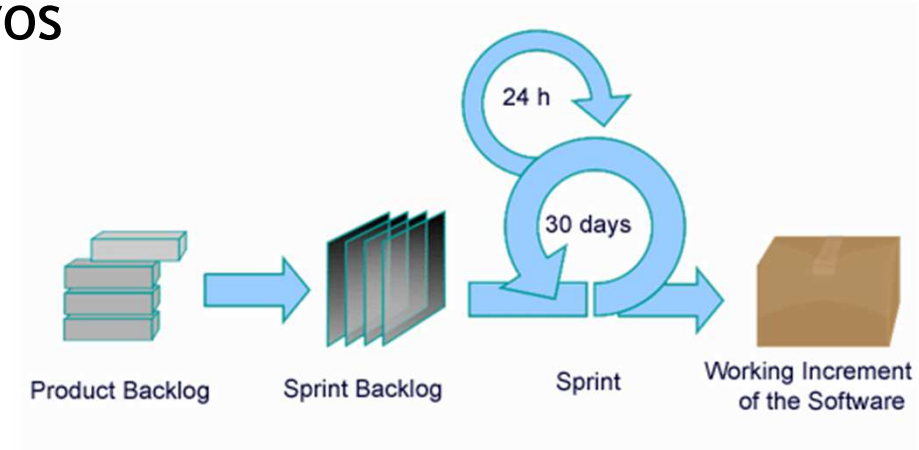
1970s: búsqueda de procesos

1980s: aumento de la complejidad => paradigmas de desarrollo

1990s: métodos iterativos e incrementales

2000s: métodos ágiles

2020s: desarrollo => “no proyectos”



fiuba algo3

Mini-historia de carreras argentinas

1961: llega Clementina a la FCEN

1960s: carrera de Computador Científico en FCEN
- Departamento de Matemática
- Muy centrada en programación

1970s: carrera de Analista Universitario de Sistemas en FIUBA
- Centrada en problemas ingenieriles y cuestiones organizacionales

1990s: carrera de Ingeniería Informática en la FIUBA
- Y cambio de nombres de las carreras antiguas: licenciaturas

1990s: primera Facultad de Informática de la Argentina: UNLP

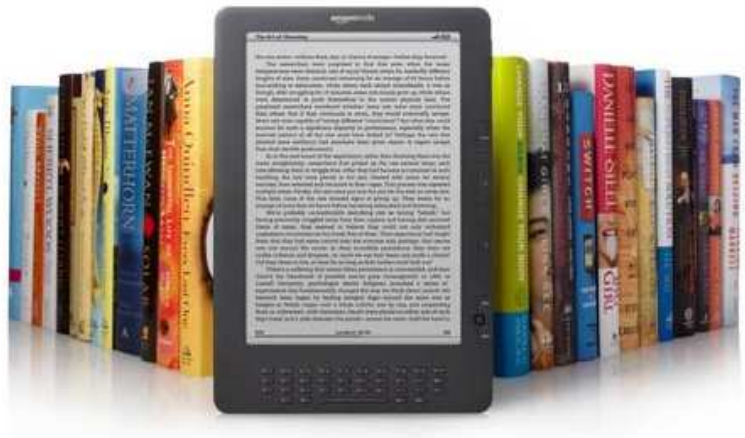
2010s: …

fiuba algo3

Software hoy

"El software se está comiendo al mundo"

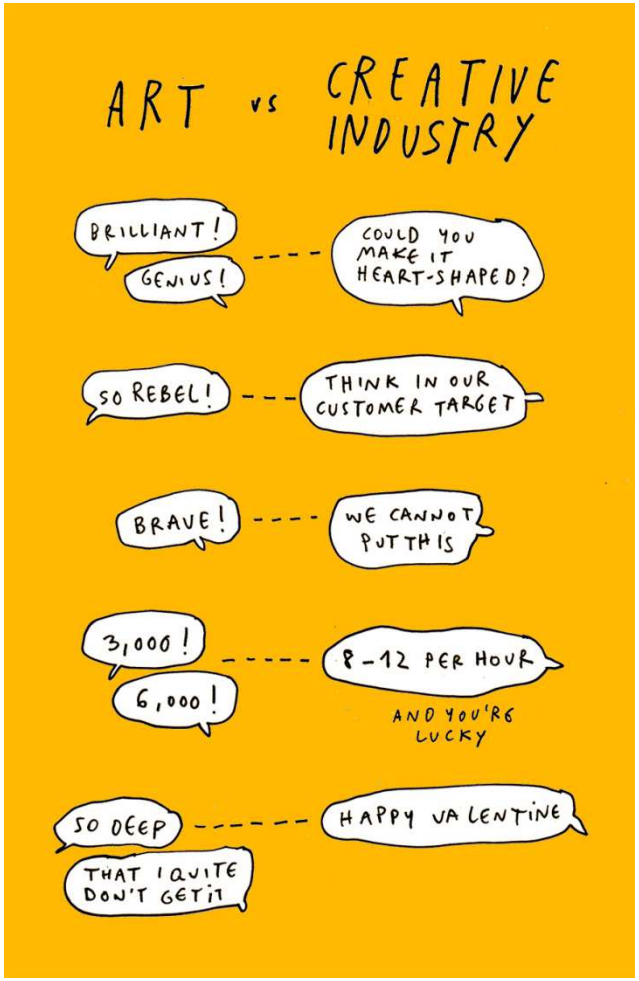
Marc Andreesen, 2011



fiuba algo3

Características del software

- Intangible
- Maleable: posibilidad de cambio
- Se desarrolla por proyectos o se mantiene en el tiempo como producto
  - Diferencia con manufactura industrial
- Alto contenido intelectual
  - Disperso y difícil de reunir
- El software no se “fabrica”
  - Se **construye** o se **desarrolla**
- Mantenimiento constante
  - desde su construcción



fiuba algo3

Fracasos del desarrollo de software (1)

- Proyectos que no terminan a tiempo
  - Aeropuerto de Denver: sistema de administración de equipajes
  - agosto 1994 => diciembre => marzo => mayo
  - pérdidas de U$S 1 M por día de atraso
- Proyectos que cuestan más de lo estimado
  - PPARS (proyecto de administración de personal, Irlanda)
  - Estimado en € 8,8 M => € 140 M

fiuba algo3

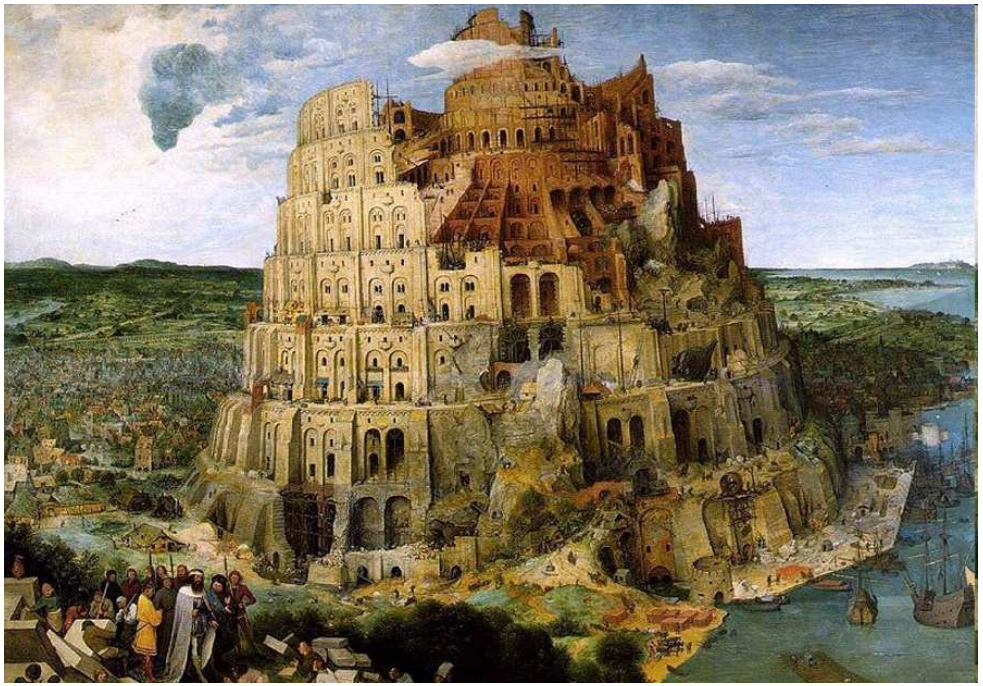
Fracasos del desarrollo de software (2)

- "Accidentes"
  - Software del Ariane 5
    - Explota a poco de salir por pérdida total de información de guiado y altitud
    - Origen: uso de software del Ariane 4
- Productos que no cumplen lo que el solicitante quiere
  - 45% del software contratado nunca es usado

fiuba algo3

Algunas reflexiones

- Los problemas del desarrollo no son sólo tecnológicos
- Ley de Brooks: agregar gente a un proyecto atrasado lo atrasa más
- Cuidar la comunicación



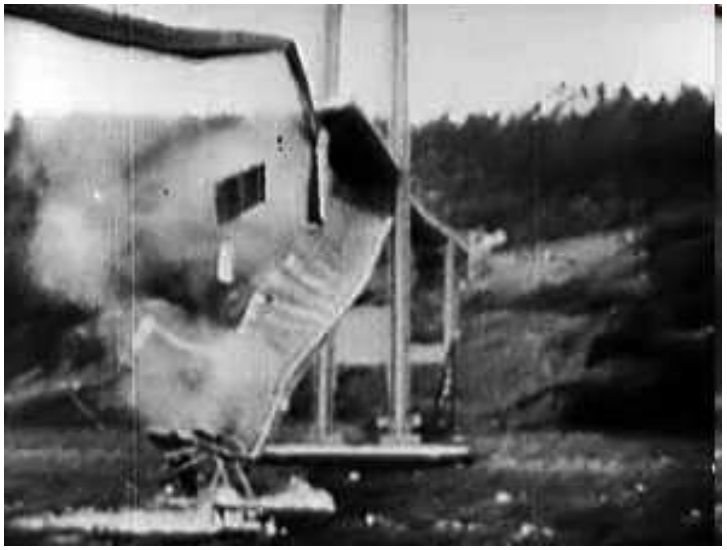
Trabajar con personas

fiuba algo3

Problemas en otras ingenierías de proyectos (1)

Tacoma – Narrows

- Colapsó en 1940



Puente de Aviñón

- Construido en 1171
- Destruido varias veces
- Último intento 1660

fiuba algo3

Problemas en otras ingenierías de proyectos (2)

- Accidentes en Three-Mile Island, Chernobyl, Fukushima
- Yacyretá-Apipé
  - US$ 11.000 M en 15 años
- Puente Chaco-Corrientes
  - Fallas de diseño (1973)
- Big-Dig, Boston, EEUU
  - 2,8 MM => 14 MM
- Catedral de Colonia, Alemania
  - 1248-1880
  - Hoy: reparación permanente

fiuba algo3

Recapitulación

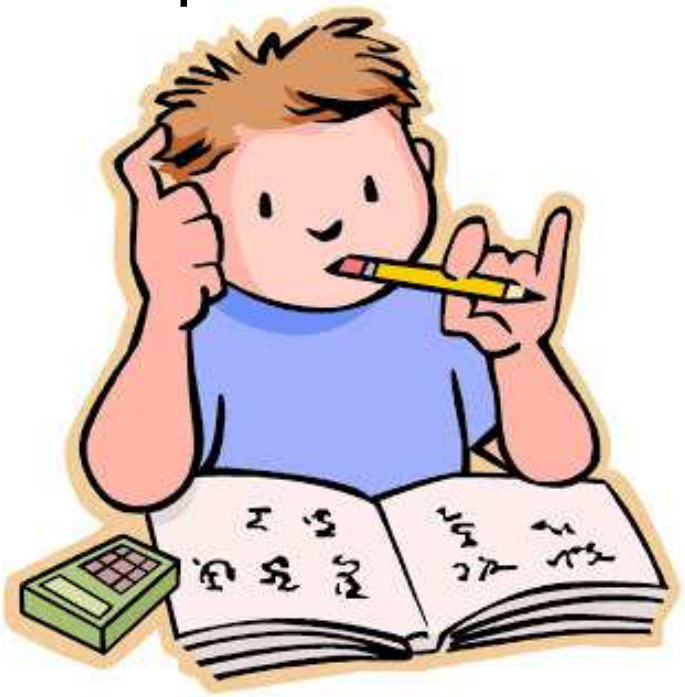
fiuba algo3

Recapitulación: preguntas

¿Por qué decimos que el software hace útil al hardware?

¿Por qué decimos que el software no se fabrica?

¿Qué verbo usamos para no decir “fabricar”?



Claves

Problemas desarrollo >> Problemas tecnológicos

Elegir y adaptar los métodos de desarrollo

- Esa es tarea de un ingeniero

fiuba algo3

Lecturas opcionales (1)

Básicos sobre métodos ágiles:

- http://agilemanifesto.org/iso/es/
- http://agilemanifesto.org/iso/es/principles.html
- http://www.mountaingoatsoftware.com/topics/scrum
- http://xprogramming.com/xpmag/whatisxp

Más avanzado:

http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf

fiuba algo3

Novedad 2014

fiuba algo3

Lecturas opcionales (2)

Artículo de Wayt Gibbs en Scientific American, “Software’s Chronic Crisis” en: http://www.cis.gsu.edu/~mmoore/CIS3300/handouts/SciAmSept1994.html

Paper de Fred Brooks, “The Mythical Man-Month”: buscar en la Web

Gojko Adzic, “Bridging the Communication Gap. Specification by example and agile acceptance testing”

Dan North, “Introducing BDD”, buscar en web

Rick Mugridge y Ward Cunningham, “Fit for Developing Software…”

Lasse Koskela, “Test Driven: TDD and Acceptance TDD for Java Developers”

fiuba algo3

Lectura obligatoria

Carlos Fontela, “Estado del arte y tendencias en Test-Driven development”

http://web.fi.uba.ar/~cfontela/Fontela_EstadoDelArteTDD_UNLP_EIS.pdf

(ojo que es largo)

fiuba algo3

Qué sigue

Concurrencia

Cierre de la materia

fiuba algo3