

¿Qué es la Ingeniería de Software ?

¿Para que sirve?

¿Por qué es necesaria?

La esencia de la Ingeniería de Software

- ▶ Modelación: transformación del concepto en requerimientos, de requerimientos en diseño, del diseño en código fuente y de este al código ejecutable
- ▶ Diseño: encontrar la mejor solución que responda a requisitos sujeto a restricciones
- ▶ Optimización: encontrar las transformaciones de mejor calidad y más económicas

Escribir un Programa

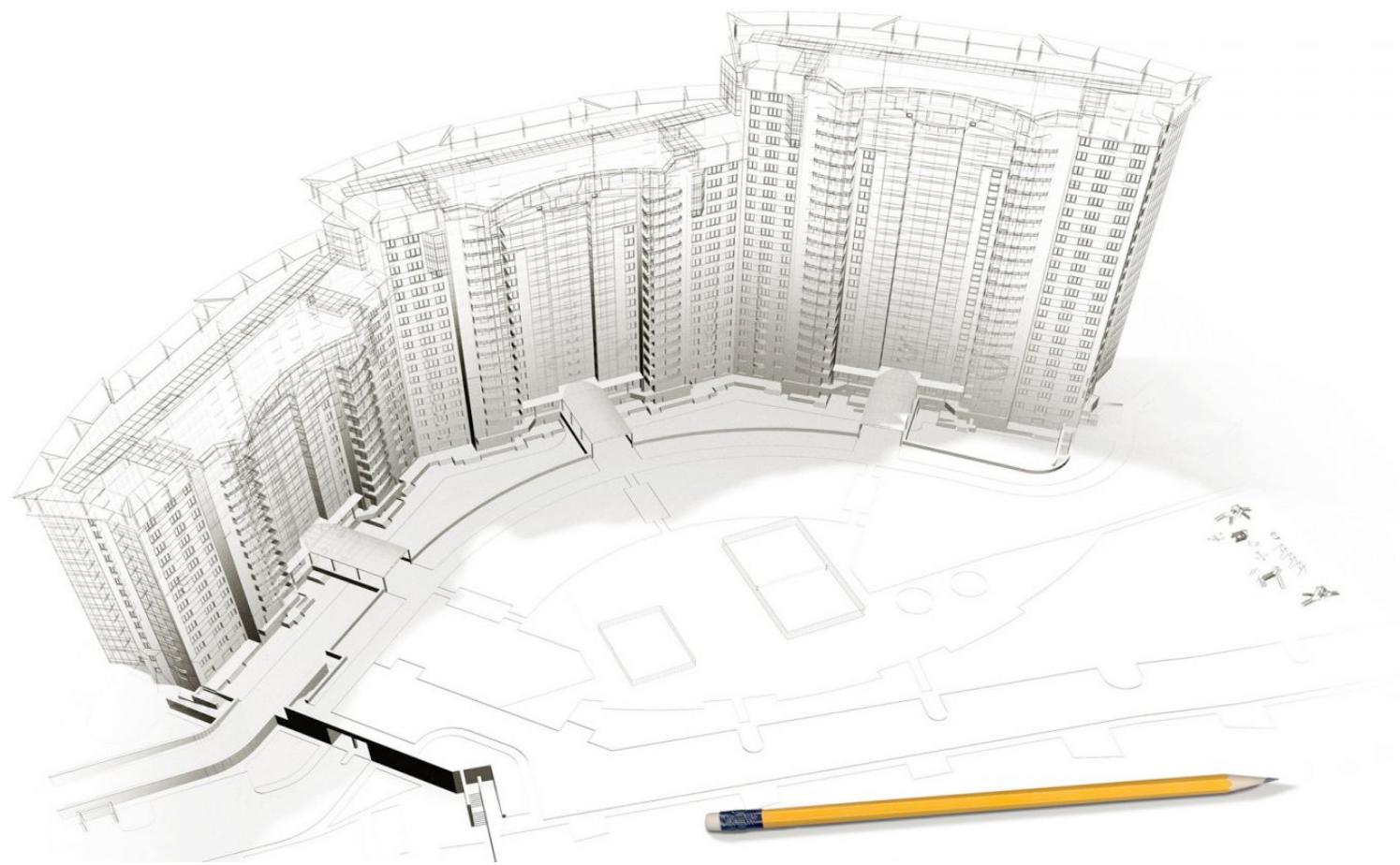
Introducción a la Programación



Programación Avanzada



Desarrollar un Software Real



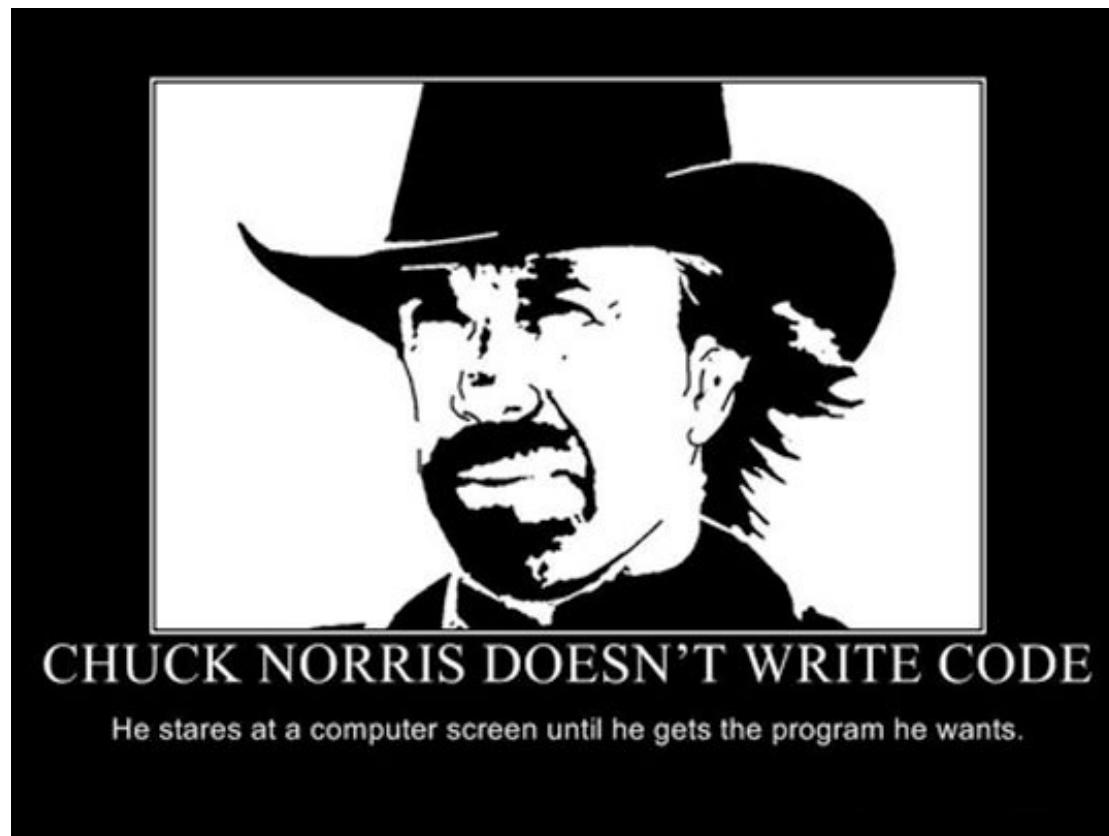
Diferencias

- ▶ participan muchas personas
- ▶ cuidadosa planeación de cada etapa
- ▶ cuidadoso diseño antes de edificar (planos, maquetas)
- ▶ cuidadosos cálculos para asegurar diseño
- ▶ revisiones y controles
- ▶ monitoreo y control del proyecto

La Necesidad de Ingeniería

- ▶ Prácticamente todo (productos, servicios) incluye software
- ▶ Software puede ser extremadamente complejo
- ▶ Debe ser extendible, modificable, escalable
- ▶ Debe ser seguro
- ▶ Se requieren métodos, herramientas y técnicas porque enfoque ingenuo, intuitivo no funciona

... a menos que seas Chuck Norris



He stares at a computer screen until he gets the program he wants.

8. Evaluación

La nota final del curso (N'_f) se calcula de la siguiente manera:

$$N'_f = 0,5 * P + 0,3 * \bar{I} + 0,2 * E$$

$$P = 0,5 * \bar{EP} + 0,2 * EF + 0,3 * PF$$

Las condiciones para aprobar el curso son las siguientes:

$$P \geq 3,95$$

$$\bar{I} \geq 3,95$$

$$E \geq 2,95$$

$$N'_f \geq 3,95$$

Considerando lo anterior, la nota final será:

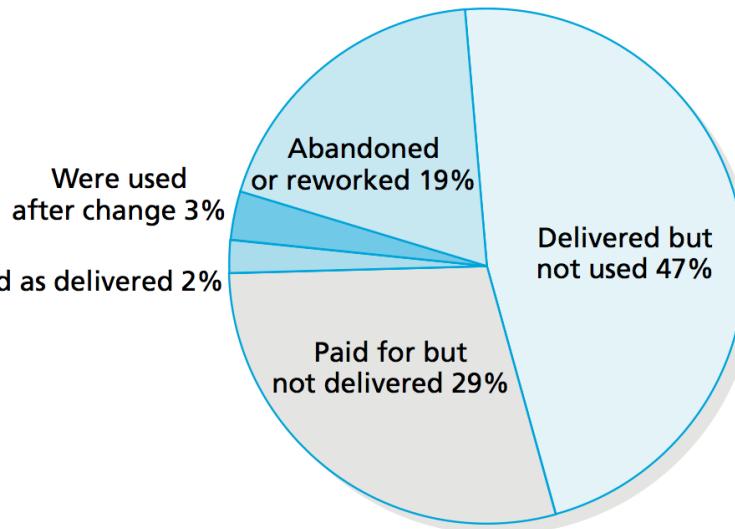
$$N_f = \begin{cases} N'_f & \text{si se cumplen todos los requisitos} \\ \min\{3,9, N'_f\} & \text{en caso contrario} \end{cases}$$

Los Principales Desafíos

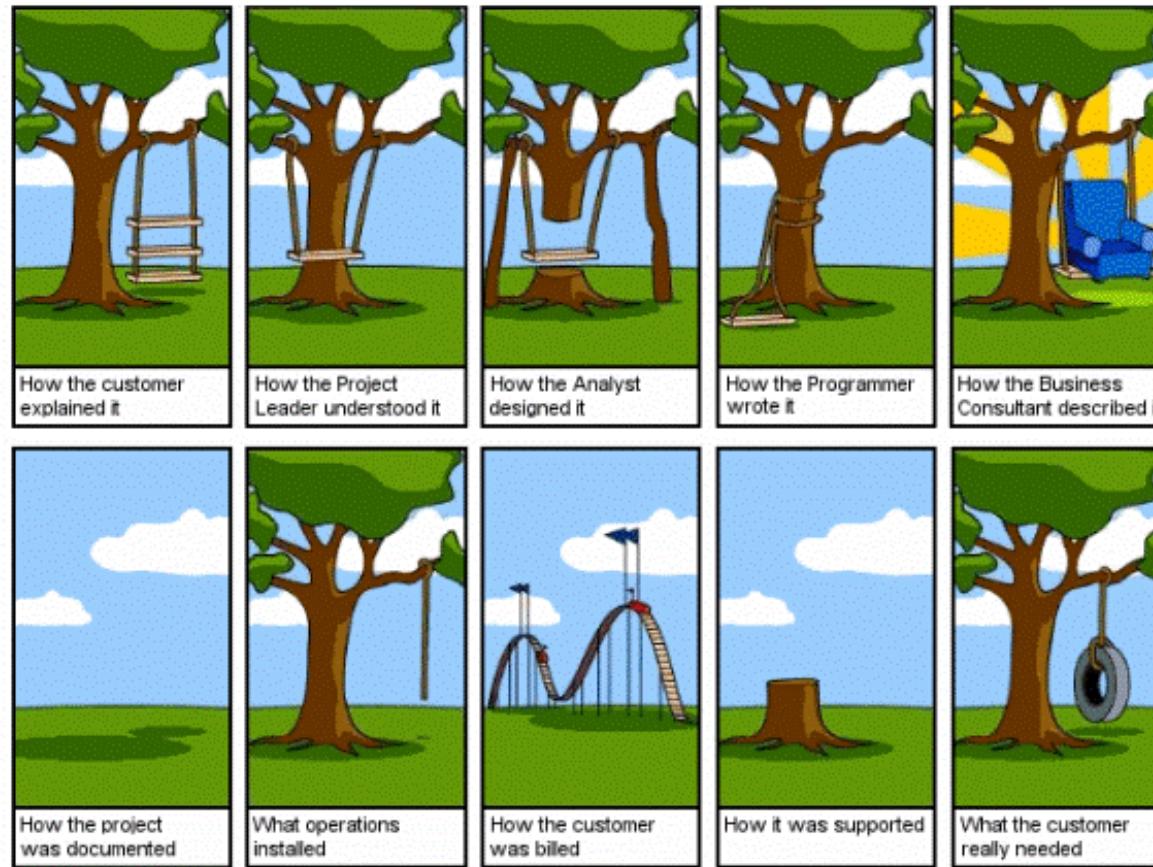
- ▶ responder a las necesidades de los usuarios
- ▶ costos reducidos
- ▶ exigencia de alto desempeño
- ▶ portabilidad
- ▶ bajo costo de mantenición
- ▶ confiabilidad
- ▶ entrega a tiempo

No es fácil responder a necesidades de los usuarios

- ▶ Análisis a de requisitos
- ▶ Se consideran críticos en el éxito de un proyecto
- ▶ Evidencia histórica muestra algunos problemas



El problema de comunicación

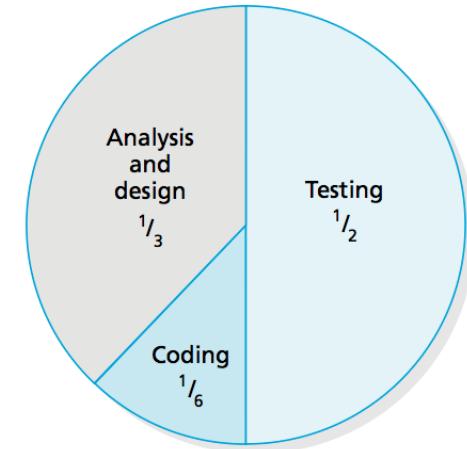


El problema de costos

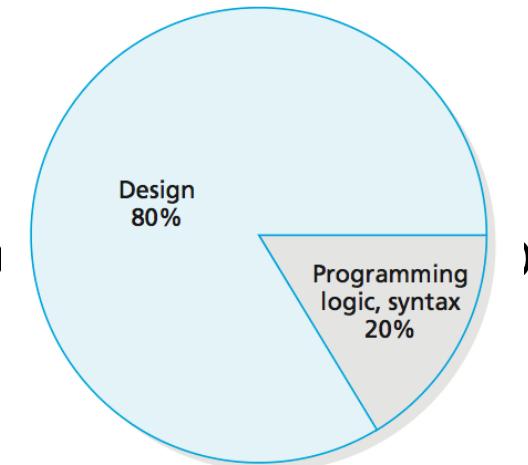
- ▶ El software puede representar mucho dinero en un proyecto
- ▶ Se gastan más de US\$1000 billones al año en el mundo
- ▶ Casi la totalidad se debe a salarios de los desarrolladores
- ▶ Productividad de programador es crucial (10 a 20 líneas de código al día)

¿ Por qué tan baja productividad ?

- ▶ Distribución de Costos por Etapa



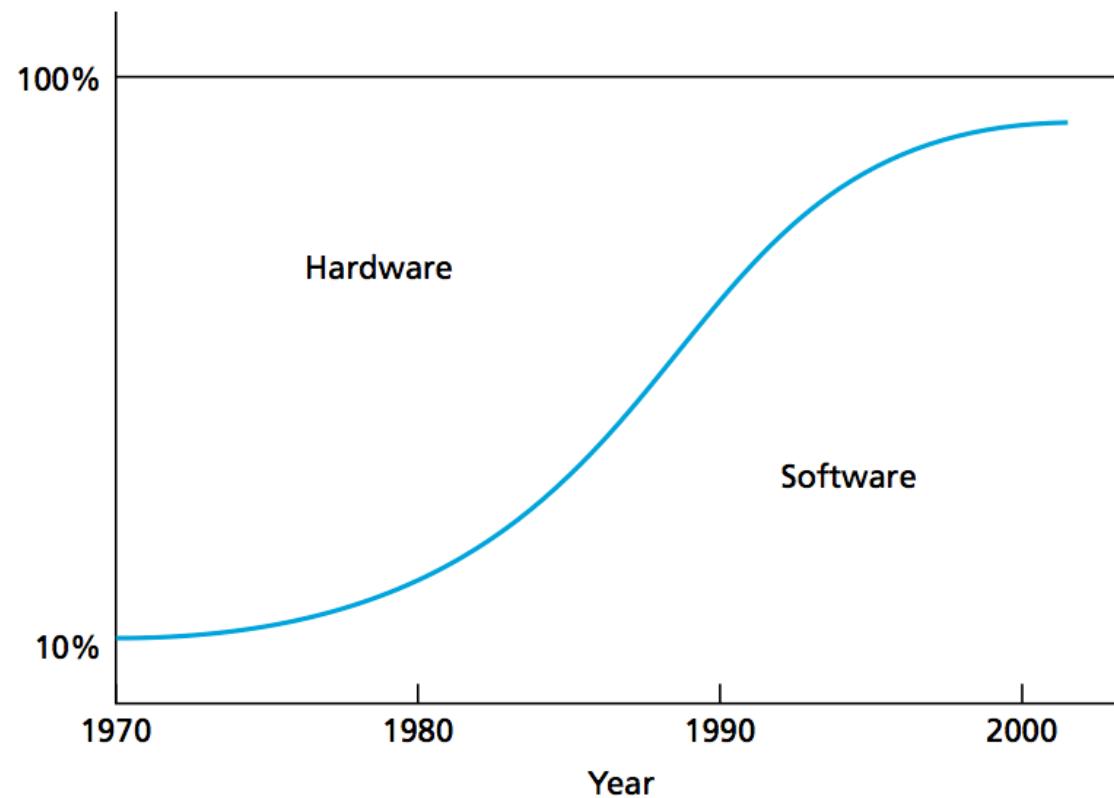
- ▶ Costo de reparar una falla de a



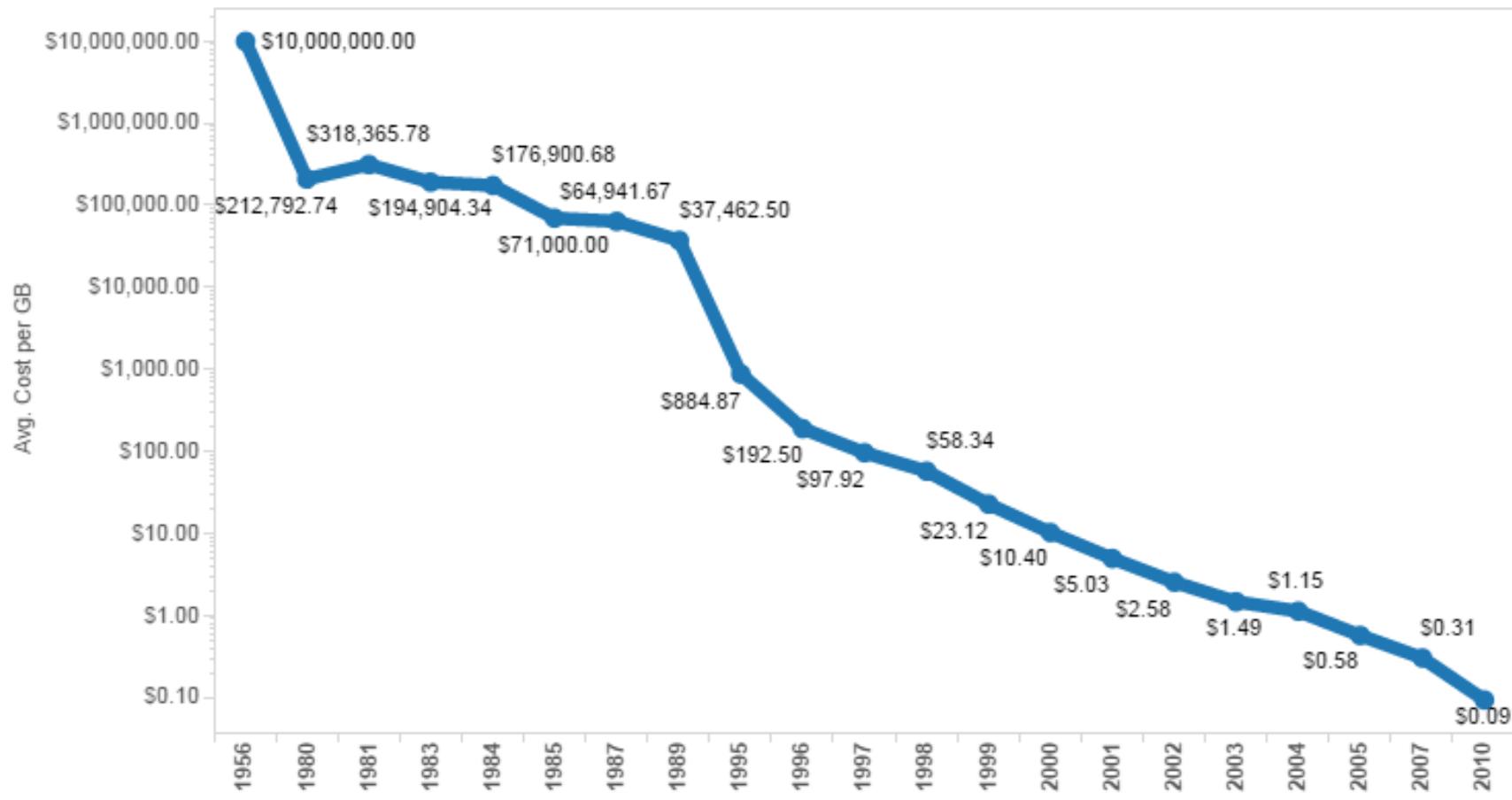
Poniendo números ...

- ▶ Un producto como Word tiene cerca de 1 millón de líneas de código
- ▶ $1.000.000 / 20 = 50.000$ días de programador
- ▶ 300 días/año = 167 años
- ▶ 80 desarrolladores durante 2 años

Los costos de Software dominan los proyectos



El Costo del Hardware se Derrumba



XCOMP introduces a complete micro-size disk subsystem with more...

- MORE STORAGE
- MORE SPEED
- MORE VALUE
- MORE SUPPORT

5100 users. The XCOMP subsystem is now available with 10 megabytes of storage; 5 megabytes also available at \$2,849.00. Compare the price and features of any other 5½-inch — or even 8-inch system, and you'll agree that XCOMP's value is unbeatable.

OUTPERFORMS OTHER HARD DISKS

Floppy disk and larger, more expensive hard disks are no match for this powerful little system. More data is available on every track: 64K on 10MB and 32K on 5MB. Faster seek time too — an average of 70MS. It provides solid performance anywhere with only 20 watts of power. Data is protected in the sealed enclosure, and the latching zone for heads provides another margin of safety. The optional power board plugs directly into the 5100 bus and provides power for the drive.

FAST CONTROLLER

The XCOMP controller is the key to this system's high efficiency operation. Speed-up features include interleaved without table lookup, block-deblock with controller buffer, and read lookahead. OEMs world wide have already proved the outstanding performance of the XCOMP controller.

MORE SOFTWARE

Included with the system is software for testing, monitoring, I/O drivers for CP/M™, plus an automatic CP/M driver attach program. Support software and drivers for MP/M™ and CPM® are also available. Sophisticated formatting program assigns alternate sectors for weak sectors detected during formatting, assuring the lowest possible error rate — at least ten times better than floppies.

WARRANTY

The system has a full one-year warranty on parts and workmanship.

ALSO AVAILABLE FROM XCOMP

- General Purpose controllers (8 bit interface), with easy interface to microprocessor-based systems
- GP controller adapter that plugs directly into most Z80 computers.
- 512 GP controller for the SMB and 10MB drive above, with ST506 type interface.
- 512R GP controller for SA1000 interface.
- SMB GP controller for storage module drives.
- ST5, 10S, and 10M, same as above, for the 5100 bus

Quantity discounts available. Distributor, Dealer, and OEM inquiries invited.

See your local Dealer, or call:

XCOMP, Inc.
7565 Trade Street
San Diego, CA 92121
Tel (714) 271-6730
Fax: 182786

Circle 428 on Inquiry Card

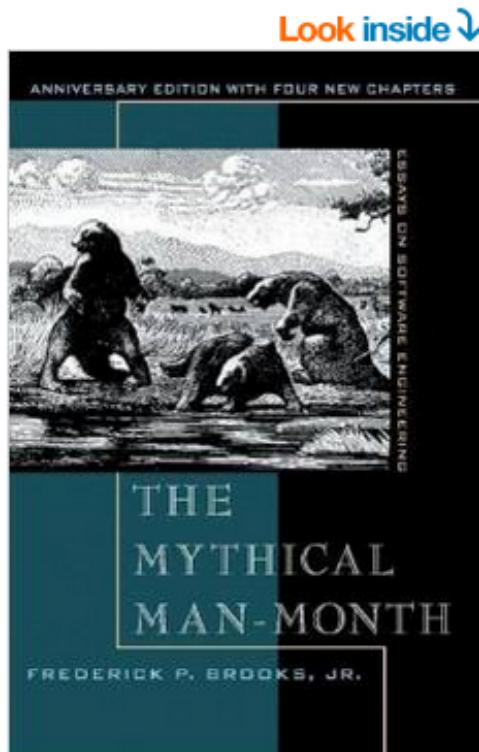


IBM 1620 EL computador de la UC

Cumpliendo con los plazos de entrega

- ▶ Un dolor de cabeza permanente
- ▶ Dificultad de hacer estimaciones de esfuerzo
- ▶ Dificultad para corregir atrasos una vez que se producen
- ▶ Desarrolladores tienden a subestimar el trabajo
- ▶ Area comercial presiona por plazos mas cortos

El libro mas citado de Ing de Software



The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2nd Edition)

Paperback – August 12, 1995

by Frederick P. Brooks Jr. (Author)

★★★★★ 243 customer reviews

#1 Best Seller in Microprocessor Design

ISBN-13: 858-0001065793 | ISBN-10: 0201835959 | Edition: Anniversary

Buy New

Price: \$27.73

65 New from \$23.10 | 69 Used from \$7.87

	Amazon Price	New from	Used from
Kindle	\$19.24	—	—
Hardcover	—	—	—
► Paperback	\$27.73	\$23.10	\$7.87



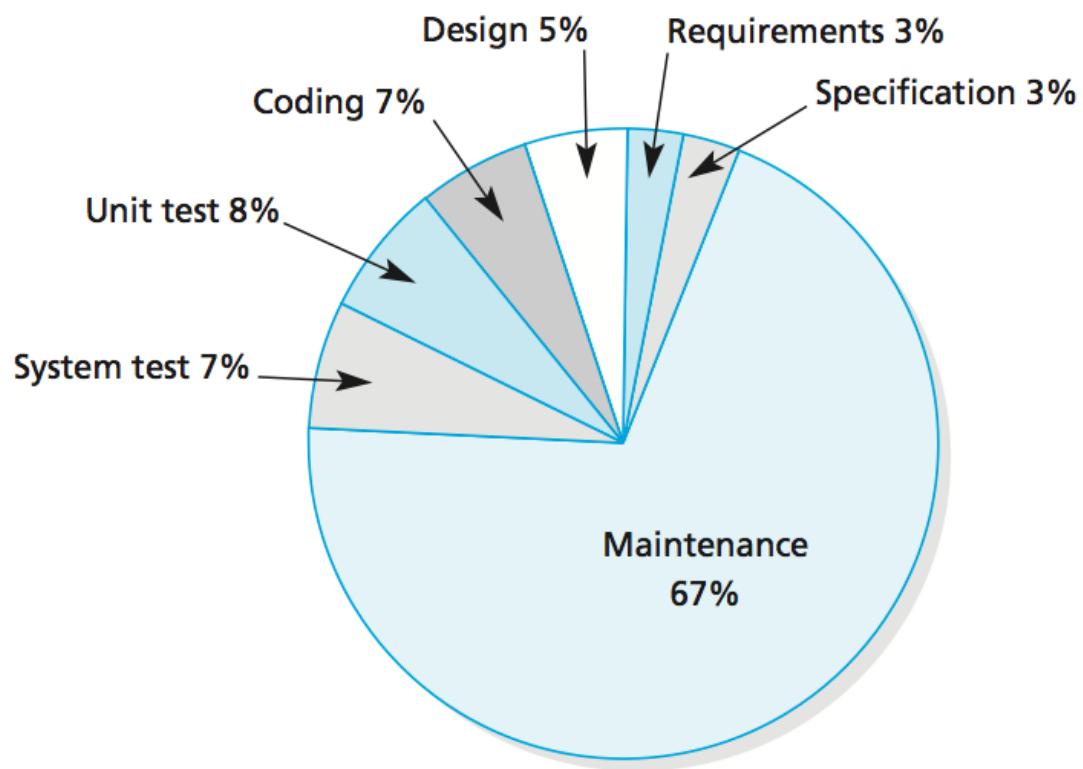
Performance

- ▶ Ha disminuido el foco debido a disponibilidad de hardware rápido y barato
- ▶ Hay proyectos en que sigue siendo crucial
 - ▶ software interactivo
 - ▶ juegos
 - ▶ señales de control
 - ▶ restricciones de ventana de tiempo (banco)

Portabilidad

- ▶ Ya que se invierte tanto en el Software tiene sentido poder llevarlo a otro hardware
- ▶ Lenguajes standard de alto nivel permiten avanzar
- ▶ Casi nunca es tan sencillo

Costos de Mantención



¿ Mantención ?



Confiabilidad

- ▶ testing permite detectar errores
- ▶ testing no permite asegurar que ya no quedan errores
- ▶ nos conformamos con “good enough”
- ▶ “zero defect” software (safety crítical)

Bugs pueden ser muy costosos

- ▶ Sonda enviada al planeta Venus incluía el siguiente bug en código Fortran
 - ▶ DO 3 I = 1.3 en lugar de DO 3 I = 1,3
 - ▶ Se interpreta como asignación DO3I = 1.3
 - ▶ Sonda fue perdida

Otros bugs célebres

- ▶ Cohete Europeo Arianne 5 se estrelló en 1996 30 segundos después de lanzamiento (US\$500 millones a la basura)

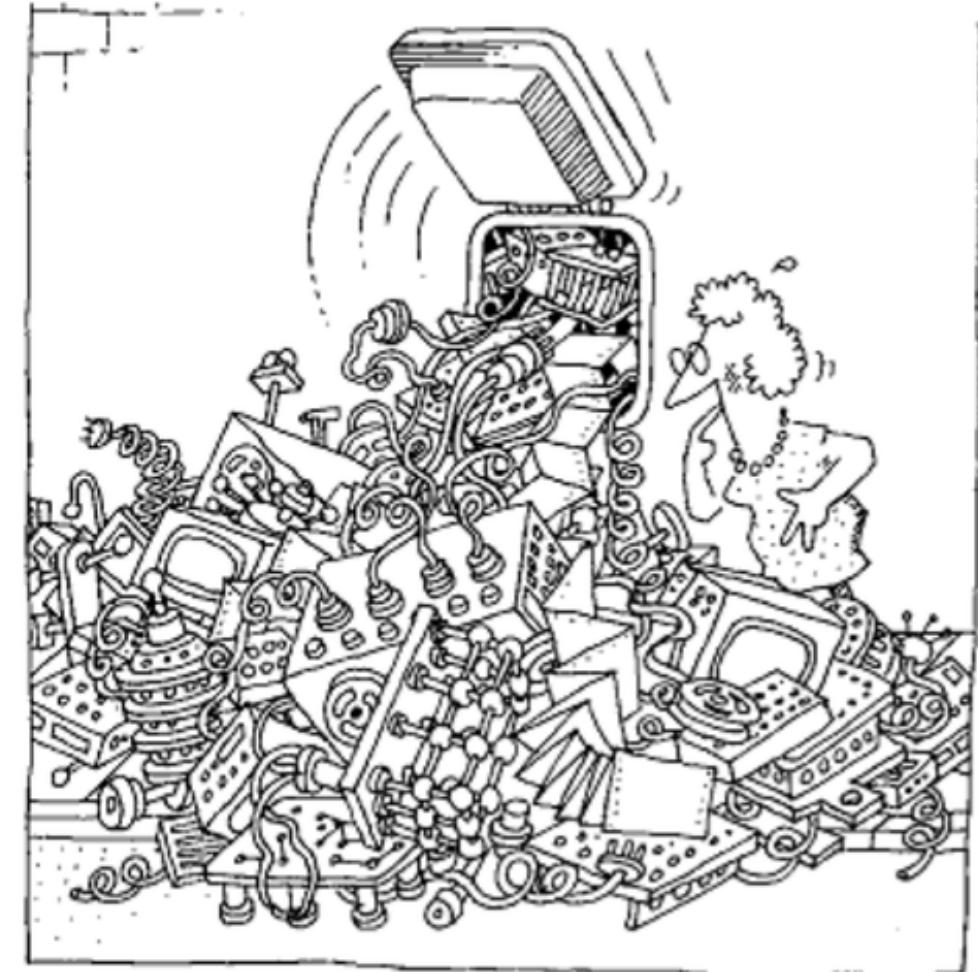
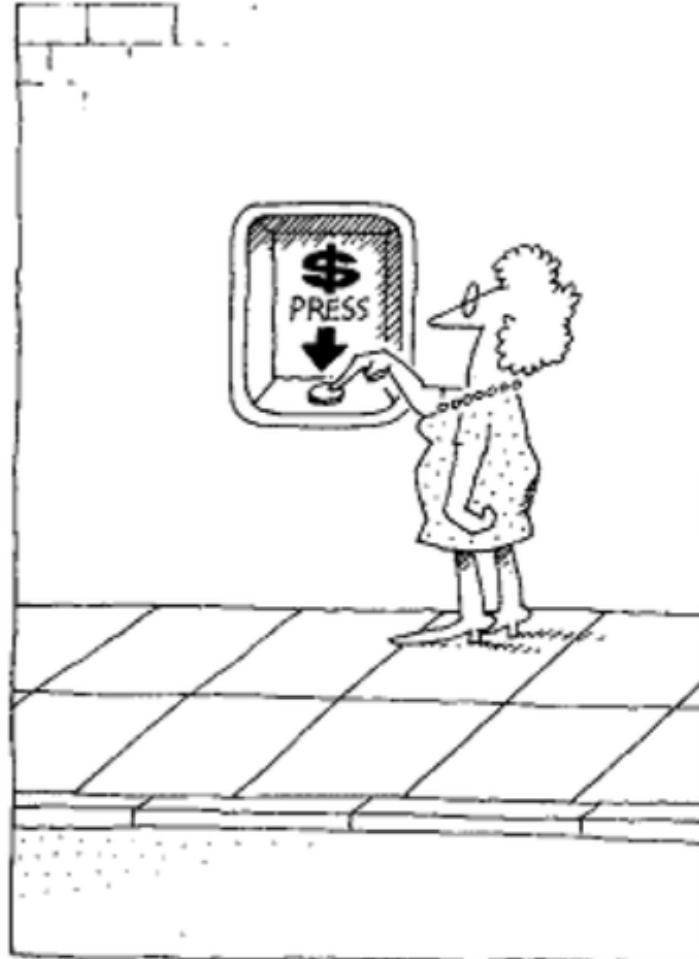


- ▶ En 1999 eBay se cayó por 22 horas - US\$6 billones de pérdidas en la acción

Interacción Humano Computador

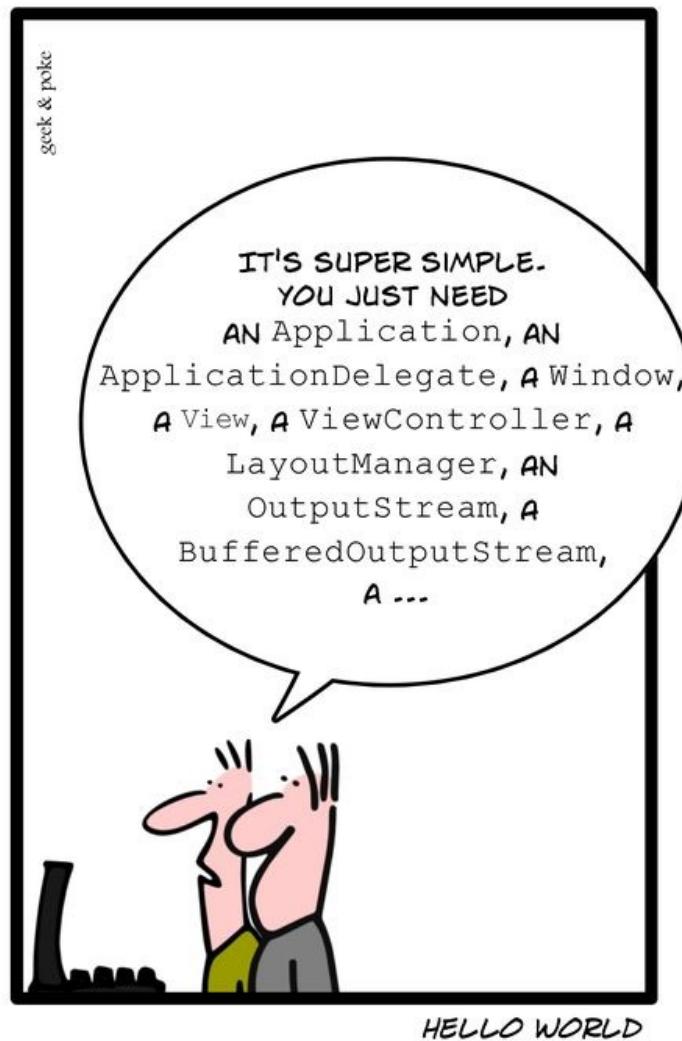
- ▶ Puede ser crítico en aceptación de un producto o servicio
- ▶ Algunos ejemplos
 - ▶ programación de un VCR
 - ▶ servicios de transferencia de llamadas
 - ▶ pago automatizado de estacionamiento
 - ▶ pagos por internet

Lo complejo debe verse simple



aunque a veces es a la inversa ...

SIMPLY EXPLAINED



Houston, we have a problem

- ▶ el software a menudo no hace lo que usuarios quieren
- ▶ el software es muy caro
- ▶ el software no es suficientemente rápido
- ▶ el software no puede ser portado
- ▶ el software es caro de mantener
- ▶ el software es poco confiable
- ▶ el software siempre se atrasa
- ▶ el software es difícil de usar

Ingeniería de Software al Rescate

- ▶ mayor énfasis en llevar a cabo el desarrollo en forma sistemática
- ▶ herramientas de apoyo
- ▶ mayor énfasis en averiguar lo que el usuario exactamente requiere
- ▶ demostrar en forma temprana el sistema a los clientes
- ▶ mayor énfasis en asegurar que no hay errores
- ▶ desarrollo incremental

¿ Que aprenderemos ?

1. Motivación

- la necesidad de ingeniería de software
- software como servicio
- desafíos y oportunidades

2. Proceso

- la necesidad de proceso
- modelo de cascada
- procesos iterativos: prototipos y RUP
- procesos incrementales
- métodos ágiles
- Scrum y Kanban

3. Requisitos

- funcionales y no funcionales
- relatos de usuario
- casos de uso

4. Diseño

- modelo de dominio
- atributos de un buen diseño
- acoplamiento y cohesión
- diagramas UML de clases, secuencia y estados
- patrones de diseño

5. Arquitectura

- conceptos fundamentales
- atributos que impactan la arquitectura
- patrones arquitectónicos
- arquitecturas cliente servidor y mult capas
- arquitectura orientada a servicios
- microservicios

6. Gestión del Proyecto

- actividades de gestión
- estimaciones
- planeación de producto, release y sprint
- gestión de personas

7. Aseguramiento de Calidad (QA)

- definiciones de calidad
- prevención de defectos
- detección y eliminación de defectos
- testing

8. Métodos Estadísticos en Ingeniería de Software

- principales desafíos
- métricas
- métricas para ingeniería de software
- aplicaciones de distribución binomial y geométrica
- aplicación de teorema de Bayes

Bonus Track

- ▶ Arquitectura de una Aplicación Web (SaaS)
- ▶ Estándares de la Web (Http, Html, CSS, URI)
- ▶ Fundamentos del Lenguaje Ruby
- ▶ Utilización de un framework de desarrollo (Rails)
- ▶ Desarrollo de una aplicación en plataforma Ruby/Rails
- ▶ Pruebas unitarias en la plataforma Ruby/Rails