No hay balas de plata

Fredrick P. Brooks, Jr. Universidad de Carolina del Norte en Chapel





MODIFICABILIDAD

- El software debe cambiar
- La funcionalidad tiene mayor presión para cambiar

Dos factores principales:

- Extender funcionalidad
- Adaptarse al nuevo hardware



COMPLEJIDAD

Dificultades:

- comunicación entre miembros del equipo de desarrollo
- enumerar todos los estados posibles del software
- incorporar nuevas funcionalidades sin crear efectos laterales
- aprendizaje y entendimiento del software



AVANCES QUE RESOLVIERON DIFICULTADES ACCIDENTALES

Lenguajes de alto nivel

- Incrementaron la productividad, confiabilidad y simplicidad
- Evito pensar en cuestiones de bajo nivel que nunca fue inherente al desarrollo de software (bits, registros, discos, saltos, etc)
- Permitió representar construcciones que se desean en el programa abstracto



AVANCES QUE RESOLVIERON DIFICULTADES ACCIDENTALES

Tiempo compartido



- Mejoró de manera importante la productividad de los programadores y calidad del software
- Atacó el tiempo de respuesta bajo de la compilación y ejecución del software
- Permitió mantener control de la complejidad al bajar lo tiempos de respuesta

AVANCES QUE RESOLVIERON DIFICULTADES ACCIDENTALES

Sistemas de programación unificados

- Unix e Interlisp, son los primeros sistemas integrados de programación usados ampliamente
- Atacan las dificultades accidentales que resultan de usar programas individuales en conjunto.



Verificación de programas



Interesante, pero costoso, y poco aplicable

No asegura programas libres de error. Solo que satisfacen la especificación.

No solo de la ruta feliz se compone un sistema exitoso.

Programación orientada a objetos

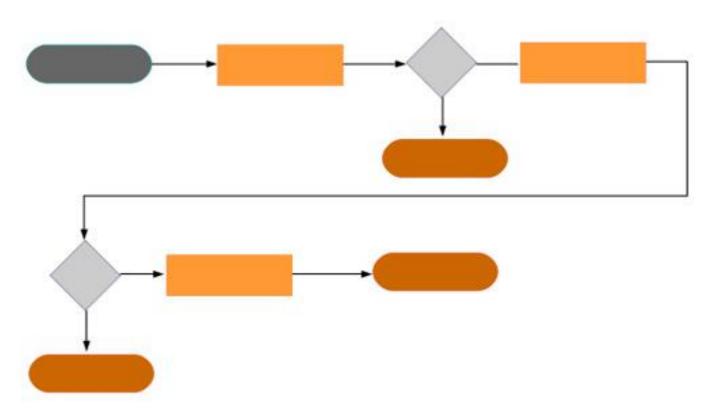


Inteligencia artificial



¿En qué consiste? ¿En qué consiste? ¿Tiene futuro en el desarrollo de ¿Tiene futuro en el desarrollo de ¿Tiene futuro en el desarrollo de ¿Tiene futuro en el desarrollo de

Programación gráfica



Programación "automática"

Ha sido siempre un eufemismo para la programación en un lenguaje de más alto nivel que el disponible actualmente.



El ambiente de trabajo Ambientes, herramientas y estaciones de trabajo



¿Objetivo?

Pelear menos con el código, Pensar más.

Sistemas expertos

Un consejero útil



ATAQUES A LAS CARACTERÍSTICAS ESENCIALES Comprar vs construir



¿Que es más conveniente?



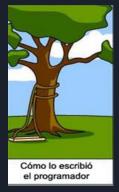
ATAQUES A LAS CARACTERÍSTICAS ESENCIALES

Refinamiento de requerimientos y prototipos rápidos

















ATAQUES A LAS CARACTERÍSTICAS ESENCIALES Desarrollo incremental



Que funcione, antes que nada

Más fácil de extender, adaptar, y prototipar. Más entusiasmo, por tener algo visual y tangible

Uno tiene <u>en cada momento</u> un sistema ejecutable

