

Frederick P. Brooks No Silver Bullet Essence and Accident in Software Engineering

Grupo 3

Ingeniería de Software II - 1er Cuatrimestre 2016

June 16, 2016

Qué vamos a ver

- 1 ¿Silver Bullet?
- 2 Dificultades esenciales vs. Dificultades accidentales
 - Propiedades inherentes a la esencia del software
- 3 Avances pasados sobre dificultades accidentales
- 4 Hopes for the Silver
- 5 Promising Attacks
- 6 ¿Qué pasó después del paper?

¿Silver Bullet?

- Un proyecto de software es como un hombre-lobo: usualmente inocentes y familiares, pero con la capacidad de convertirse en un monstruo.
- Buscamos una bala de plata que disminuya drásticamente los costos de producir software.
- Sin embargo, por la naturaleza misma del software, no hay ningún avance en la tecnología ni en las técnicas de gestión que por sí mismo prometa un aumento en productividad, confiabilidad y simplicidad de siquiera un orden de magnitud, de la misma manera que ocurre con el hardware.

Dificultades esenciales vs. Dificultades accidentales

Dos tipos de dificultades

- Esenciales
- Accidentales

Esencia del software

- Construcción de conceptos entrelazados
- Abstracta

La parte difícil de contruir software es la especificación, diseño y testeo de la construcción conceptual, no representar y probar la fidelidad de la representación.

Propiedades inherentes a la esencia del software

- Complejidad
- Conformidad
- Modificabilidad
- Invisibilidad

Avances pasados sobre dificultades accidentales

- Lenguajes de Alto Nivel
- Time-Sharing
- Ambientes de desarrollo unificados
- Ada y otros avances en lenguajes de alto nivel
- Programación Orientada a Objetos

Hopes for the Silver

- Lenguajes de Alto Nivel
- Programación Orientada a Objetos
- Inteligencia Artificial
- Sistemas Expertos
- Programación Automática
- Programación gráfica
- Verificación de Programas
- Entornos y Herramientas
- Workstations

Promising Attacks

- Buy, don't build
- Refinamiento de requerimientos y prototipado rápido
- Build vs Grow
- Mantener buenos diseñadores

¿Qué pasó después del paper?

“No Silver Bullet” Refired

- Mejoras en productividad, pero no en órdenes de magnitud.
- Buy, don't build: esto parece cumplirse. Uno no implementa todos los aspectos de un sistema.
- Desarrollo incremental, iterativo con feedback de usuarios: hoy se usa mucho. Diversas metodologías en la actualidad adoptan esta filosofía.
- Reusar código (buen diseño, buena documentación, vocabulario común): es mejora marginal, pero ayuda.
- OOP: ayuda, pero no cambió al mundo... todavía
- Mantener buenos diseñadores: son rockstars. Identificarlos, retenerlos, mimarlos. Todavía falta énfasis en este aspecto.

F.P. Brooks

La complejidad es el negocio en el que estamos, y la complejidad es lo que nos limita.