Architectural Design: Designing before Design

Buenos Aires, Argentina Junio de 2006

Dr. J. Fernando Naveda
Department of Software Engineering
Rochester Institute of Technology
Rochester New York, USA

F.Naveda@rit.edu

Agenda

- ✓ Una corta historia del software
- ✓ ¿Por que arquitecturas?
- ✓ “Stakeholders”
- ✓ Dependencias y calidad
- ✓ Líneas de producto
- ✓ Arquitecturas en pre-grado y post-grado
- ✓ Conclusiones
- ✓ Preguntas

Reconocimiento

- ✓**Partes de esta charla están basadas en el libro *Software Architecture in Practice (2nd Ed.) by L. Bass, P. Clements, and R. Kazman.***

  SEI Series in Software Engineering, Addison Wesley, 2003

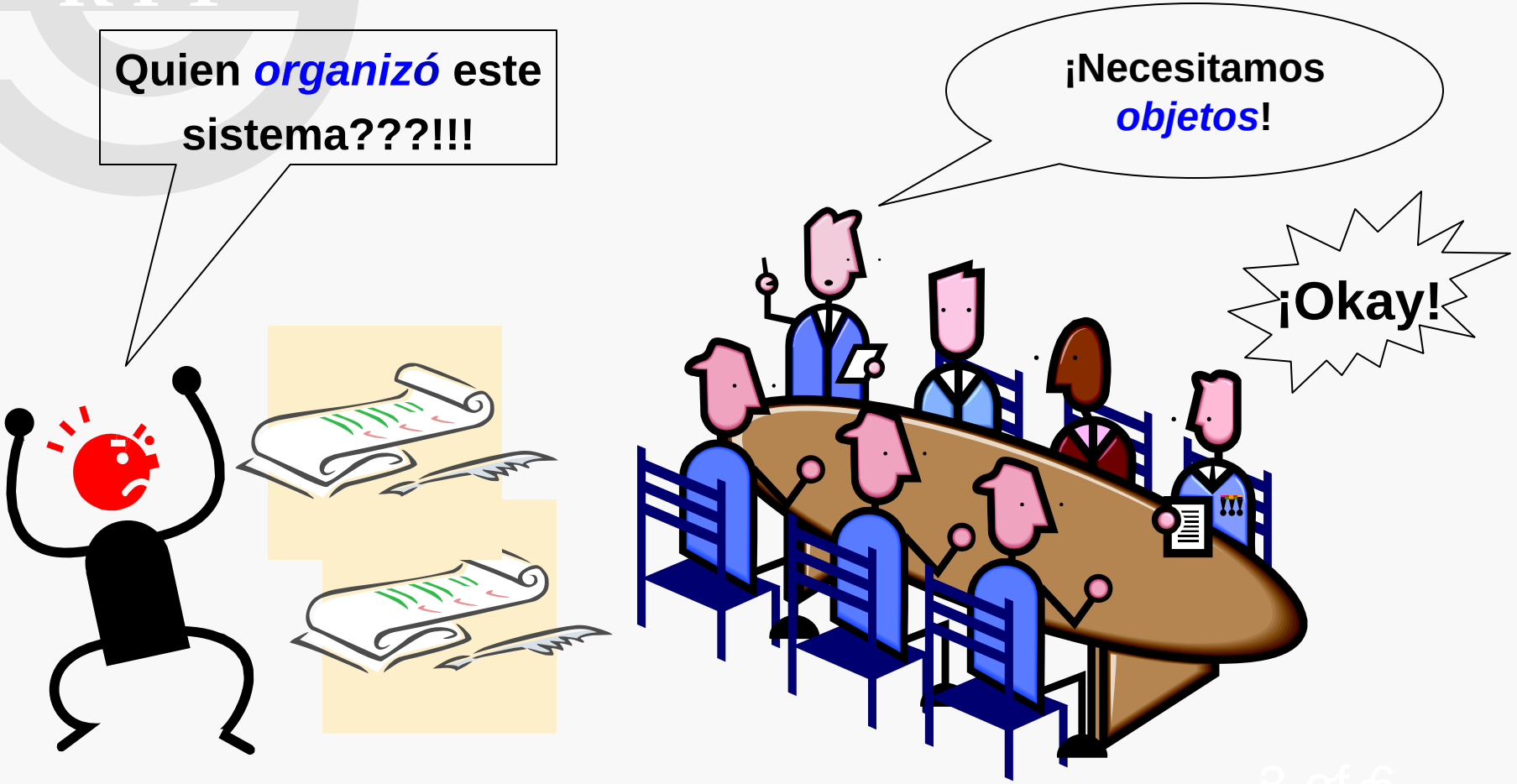
Una corta historia del software

1 of 6

Una corta historia del software

Una corta historia del software

Una corta historia del software

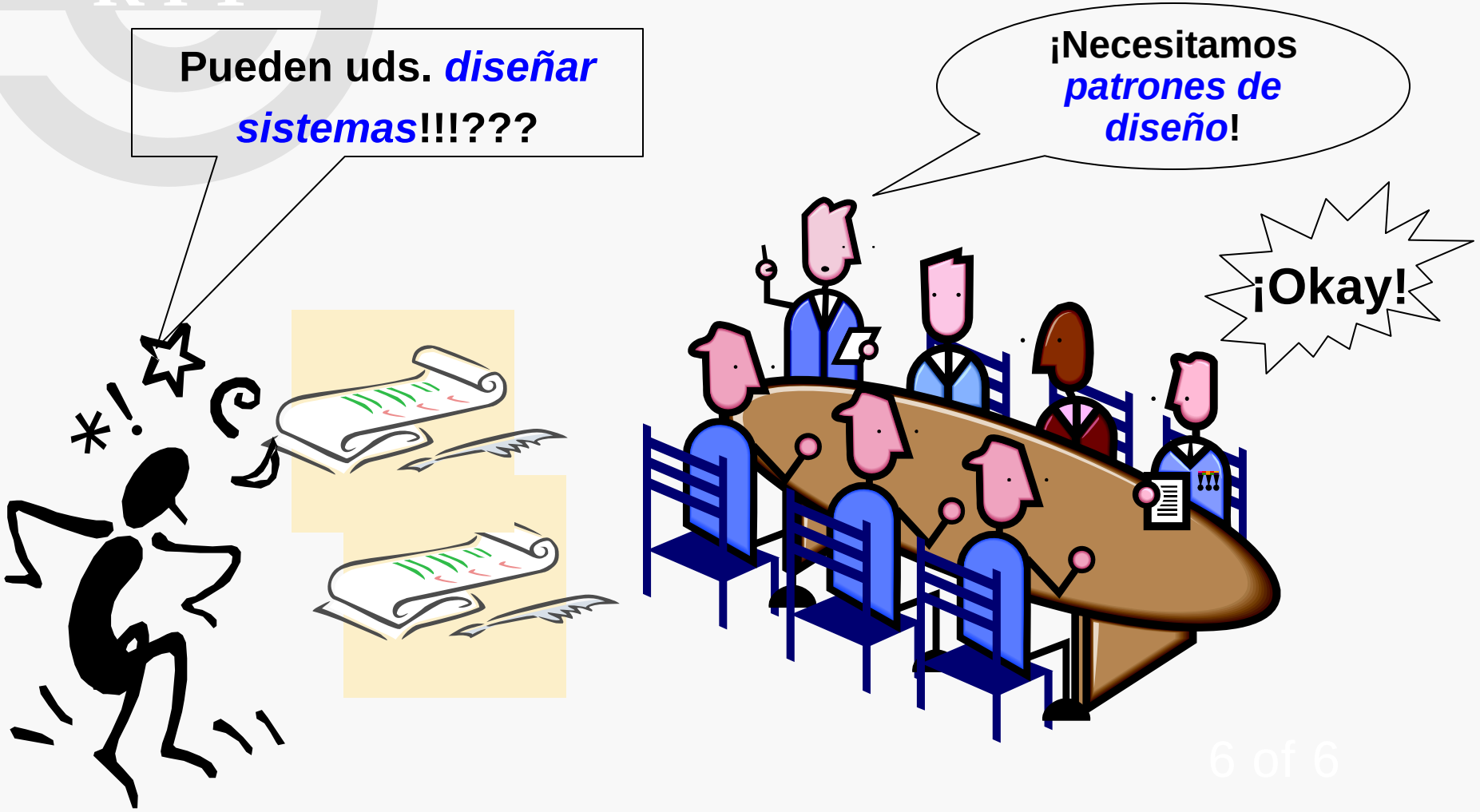


Una corta historia del software

- ✓ Abstracción
- ✓ Encapsulación de información y función
- ✓ Diseño de la interfase
- ✓ Re-uso
- ✓ Nos permiten documentar las ideas que funcionan (o no)

5 of 6

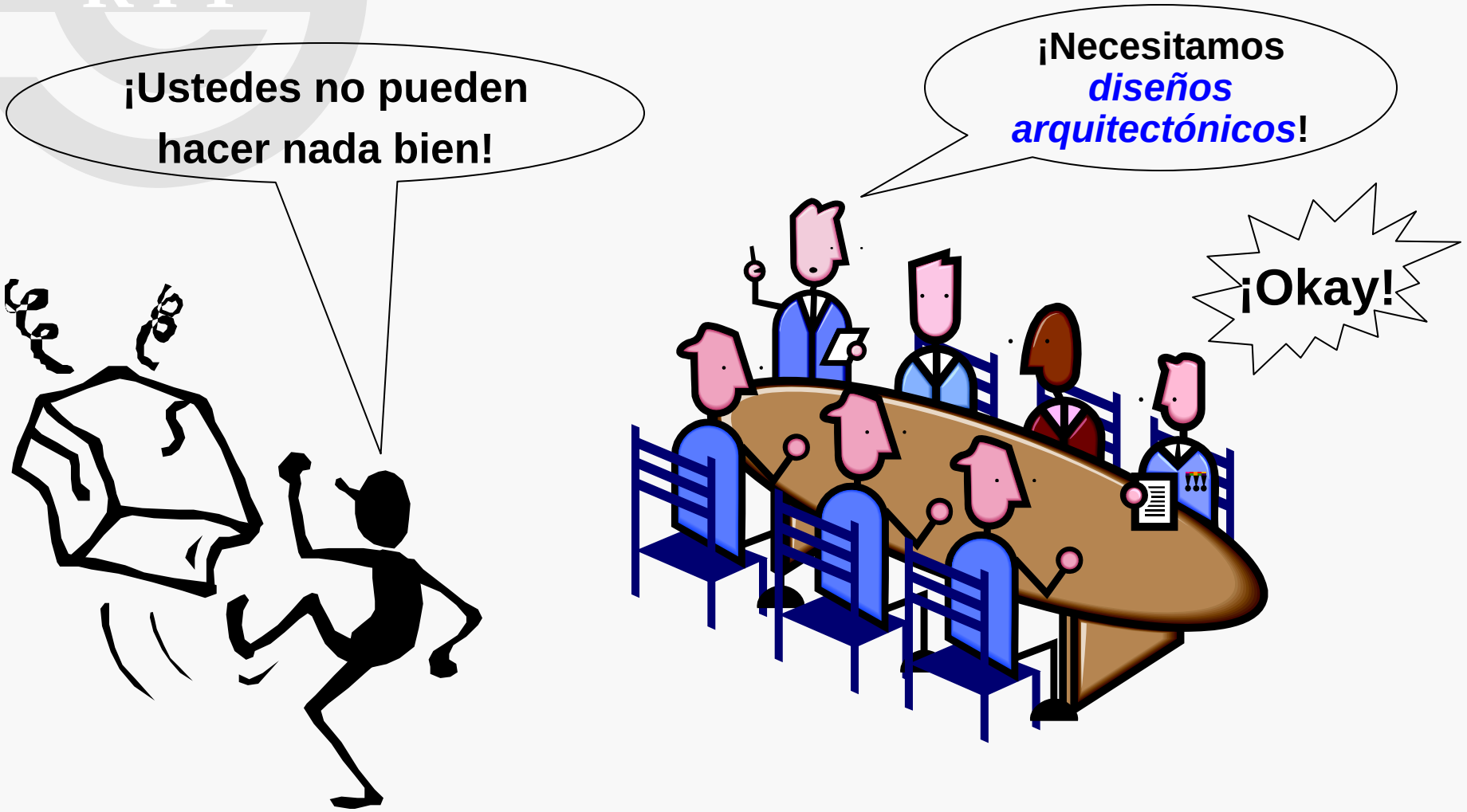
Una corta historia del software



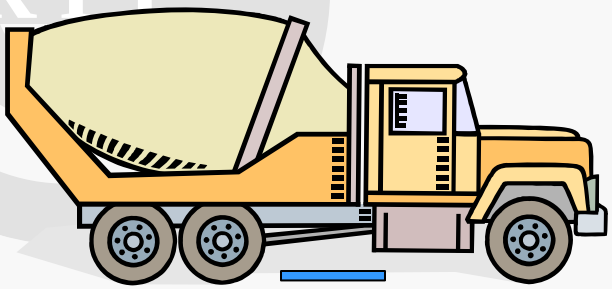
Una corta historia del software

- ✓ Facilitan el re-uso y comunicación de ideas de diseño que funcionan
- ✓ Facilitan la decisión de escoger el diseño mas adecuado
- ✓ No tenemos que adivinar
- ✓ Previenen la re-invención de lo que ya se ha inventado

Una corta historia del software



De lo concreto a lo abstracto



- ✓ **Prácticas de programación**
- ✓ **Lenguajes de programación**
  - ***ADTs (Información)***
  - ***Función***
- ✓ **Lenguajes de programación**
  - ***Orientados a objetos***
  - ***Orientados a aspectos***
- ✓ **Práctica del diseño**
  - ***Patrones de diseño***
- ✓ **Diseños arquitectónicos**

¿Por que arquitecturas?

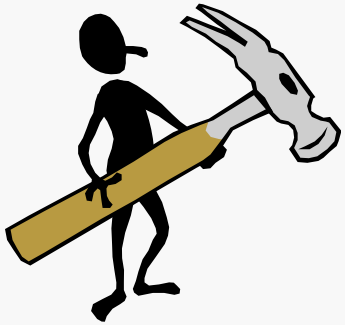
- ✓**Sistemas software son cada día mas complejos**
- ✓**Buenos diseños son difíciles de producir**
  - ***¿Por que tenemos que re-inventar y re-analizar diseños?***
  - ***Requieren una inversión fuerte de recursos tanto económicos como de tiempo***

Diseño arquitectónico

- ✓**Concierne con al estructura de sistemas software suficientemente grandes**
- ✓**La perspectiva de la arquitectura es abstracta:**
  - ***Detalles de la implementación no son importantes***
  - ***Detalles algorítmicos no son importantes***
  - ***Representación de la información se concentra en el comportamiento y la interacción entre los componentes***

“Stakeholders” y prioridades

Costo y tiempo de lanzamiento al mercado



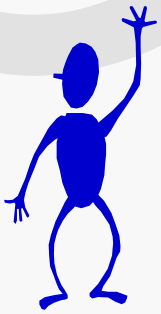
Mantenibilidad

Usabilidad y rendimiento

Tendencias técnicas

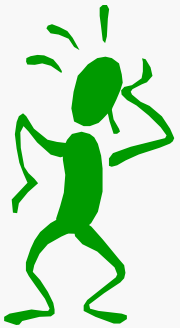
Conflictos

Tiempo al mercado



Confiabilidad

Mantenibilidad

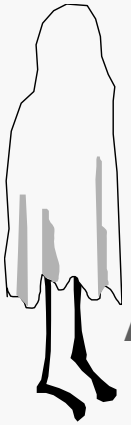


Costo

Seguridad

Usabilidad

Rendimiento



Adaptabilidad

Tendencias técnicas

Dependencias

- ✓**Modificabilidad (modifiability)**
  - *Como se dividen las funciones del sistema entre sus componentes (arquitectural)*
  - *Selección de algoritmos y técnicas de programación (no-arquitectural)*
- ✓**Usabilidad (usability)**
  - *Selección de artefactos de la interacción con el usuario (no-arquitectural)*
  - *Habilidad del usuario de cancelar o deshacer operaciones (arquitectural)*

Arquitectura y calidad del sistema

- ✓**Hablar de arquitecturas es hablar de los atributos cualitativos del sistema**
  - ***Disponibilidad (Availability)***
  - ***Seguridad (Security)***
  - ***Modificabilidad (Modifiability)***
  - ***Rendimiento (Performance)***
  - ***Habilidad de conducir pruebas (Testability)***
  - ***Usabilidad (Usability)***
  - ***Etc. (Etc.)***

Arquitecturas y calidad del negocio

- ✓**Hablar de arquitecturas es hablar del aspecto cualitativo del negocio de producir software**
  - *Tiempo de lanzamiento al mercado*
  - *Costo y beneficio*
  - *Vida proyectada del sistema*
  - *Mercado*
  - *Calendario de lanzamiento*
  - *Integración con sistemas ya existentes*
  - *Etc.*

Calidad de la arquitectura

- ✓**Integridad conceptual**

  La integridad conceptual del sistema es la consideración mas importante durante la etapa del diseño.

  Fred Brooks, 1975

- ✓**Correcto y completo**

- ✓**Constructibilidad (Buildability)**

¿Entonces que paso?

“Algo que no se puede medir, no se puede mejorar”

Lord Kelvin

Midiendo mantenibilidad

- ✓**Fuerza de interacción entre módulos**
  - *¿Hay módulos que dependen mucho de otros?*
- ✓**Integridad conceptual de los módulos**
  - *¿Cada modulo captura un concepto de diseño único?*
- ✓**Probabilidad de cambios específicos**
- ✓**Costo estimado de implementación de cambios**

La calidad se inyecta al sistema

- ✓**Calidad no es algo que es simplemente agregado en algún momento dado**
  - ***Es inyectada (o no) a medida que el producto se desarrolla***

- ✓**Decisiones tácticas se deben tomar a través del ciclo de desarrollo**
  - ***Bass, Clements, Kazman hablan de "tactics"***

Líneas de producto

- ✓ **La arquitectura del sistema representa una inversión significativa**
- ✓ **Re-usar la arquitectura puede ayudar a la organización:**
  - ***Incremento de ingresos***
  - ***Reducción de costos de producción***
  - ***Incremento de oferta de productos***
  - ***Incremento de número de clientes***

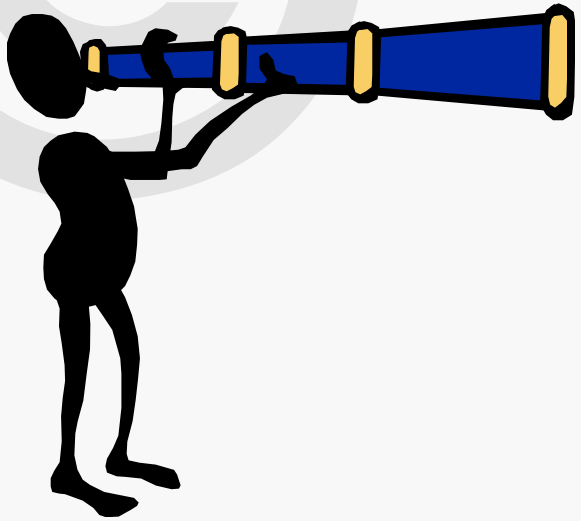
Manéjese con cuidado

- ✓No hay tal cosa como la arquitectura perfecta
- ✓La arquitectura esta íntimamente ligada al aspecto de negocios y a los requerimientos de los “stakeholders”
- ✓El arquitecto debe aprender a balancear los requerimientos del sistema contra las metas y recursos de la organización

¿Donde enseñar arquitecturas?

- ✓**A nivel de graduados**
  - *El libro completo de Bass, Clements y Kazman en un semestre*
  - *Un curso exclusivo de líneas de producto*
- ✓**A nivel de pre-grado**
  - *En la parte alta del programa*
  - *No es posible cubrir el libro completo*
  - *Líneas de producto: Solo una introducción*
  - *Uso de viejos proyectos como malos ejemplos*
  - *Re-implementación de sistemas existentes*

Conclusiones



- ✓ **Hemos caminado un largo trecho desde enfocarnos en lo concreto a identificar el valor de la arquitectura de los sistemas software**
- ✓ **Desarrollar la arquitectura vale la pena si el sistema es suficientemente grande y se espera una vida útil suficientemente larga**
- ✓ **Es importante enseñar este material en la escuela**