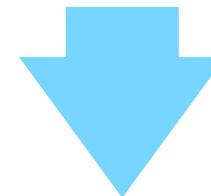
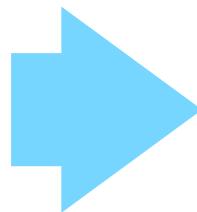


# Ingeniería de Software



Vamos profesor, no exageremos,  
no estamos haciendo edificios

La cantidad de líneas de código fuente es una de las métricas para la envergadura del software

# Líneas de Código Fuente



Google Chrome

5.000.000

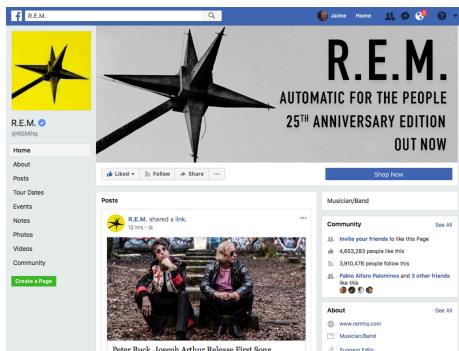


Windows 7

50.000.000

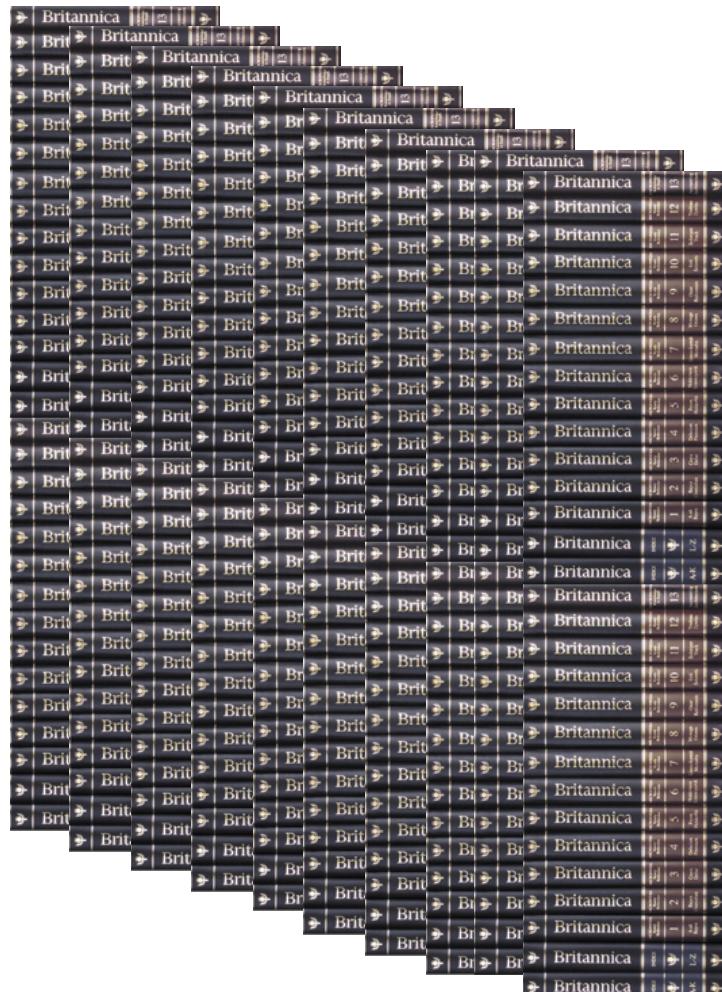
Facebook

60.000.000



Un millón de líneas de código equivale a  
60 tomos de 300 páginas de código cada uno

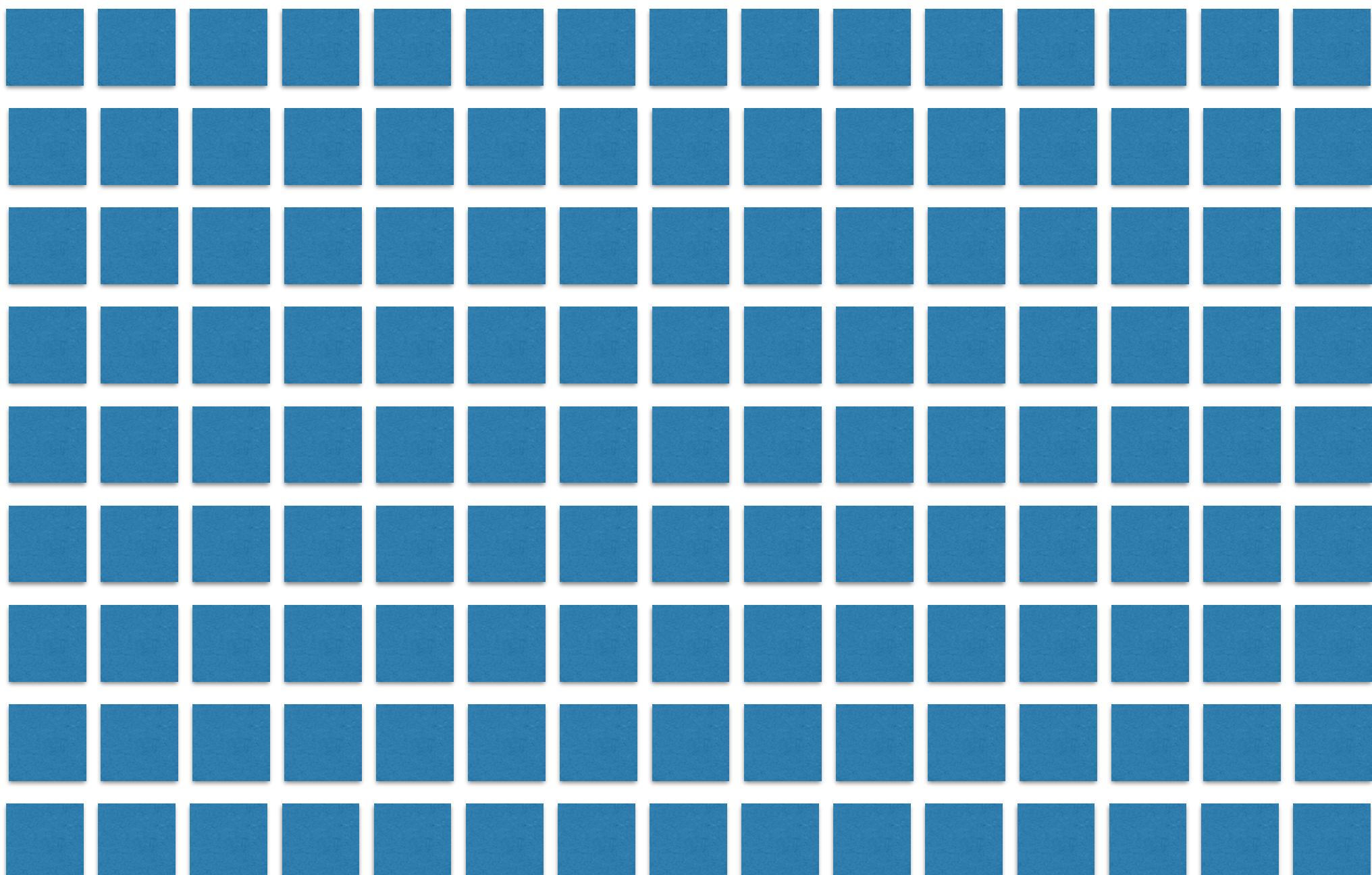
El código de chrome sería equivalente a 300  
tomas de 300 páginas cada uno !!



# El Repositorio de Google

- aprox 2 billones de líneas de código
- 2.000.000 de tomos de 1000 páginas
- 150 oficinas de 40 m<sup>2</sup> densamente llenas con los tomos

40 m<sup>2</sup>



7000 m<sup>2</sup>

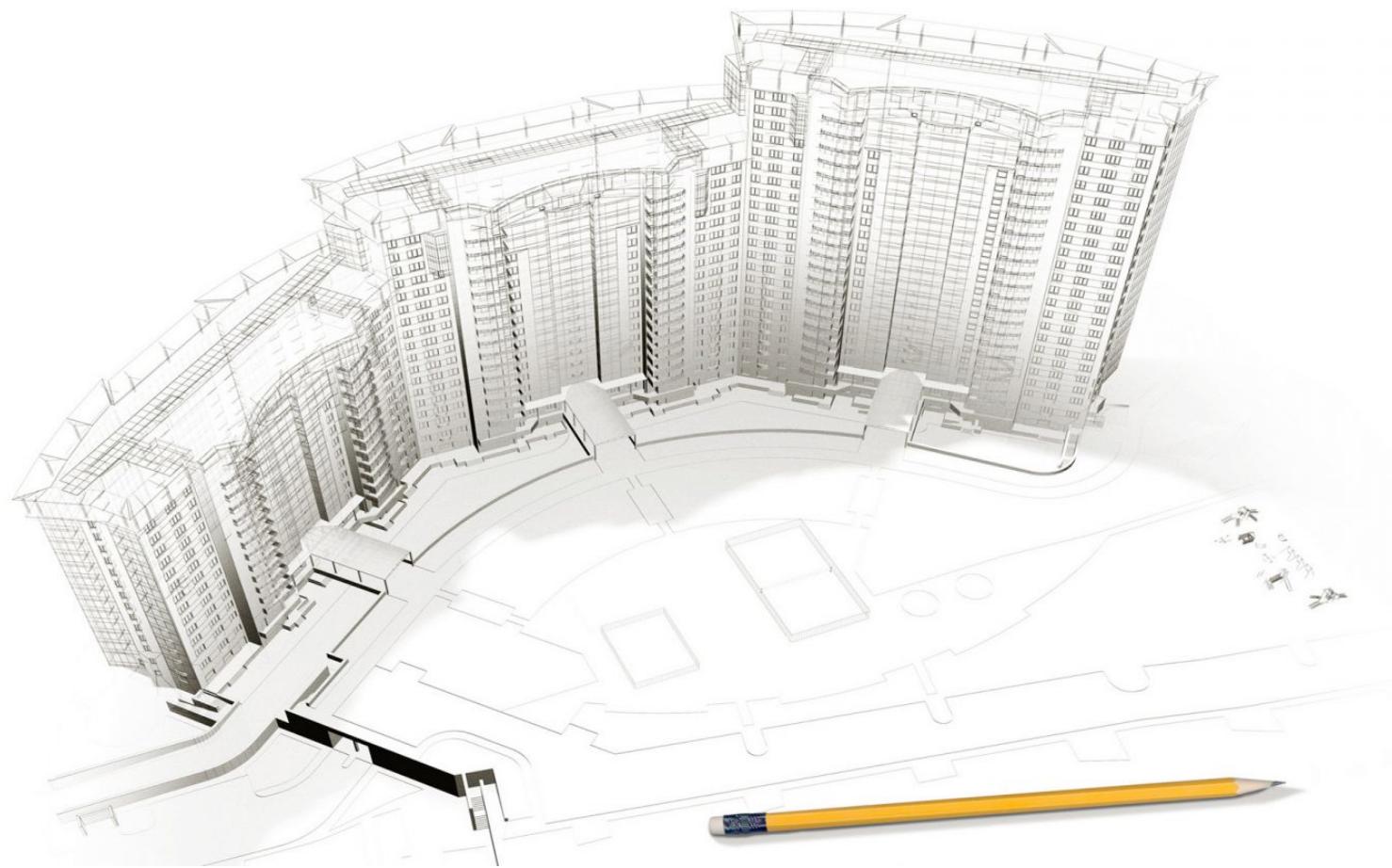
# Los principales problemas

- el software no tiene la funcionalidad necesaria (usuarios insatisfechos)
- el proyecto no logra terminar en la fecha planeada
- el proyecto termina costando mucho más que lo planeado
- aún hay muchos errores en el código

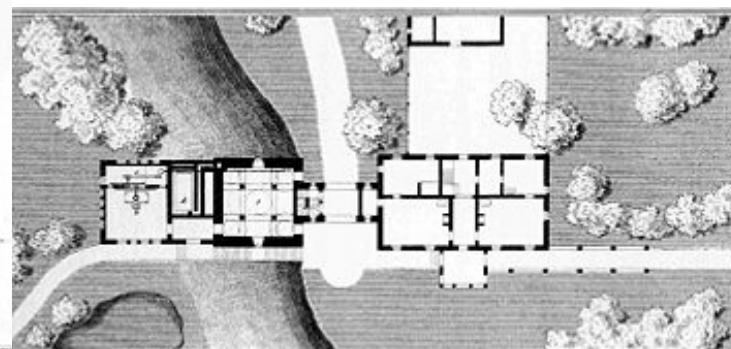
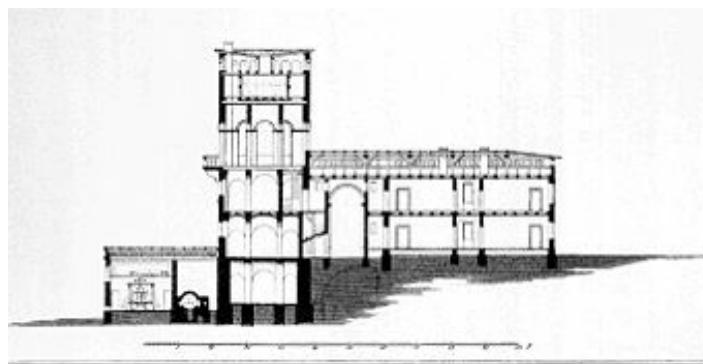
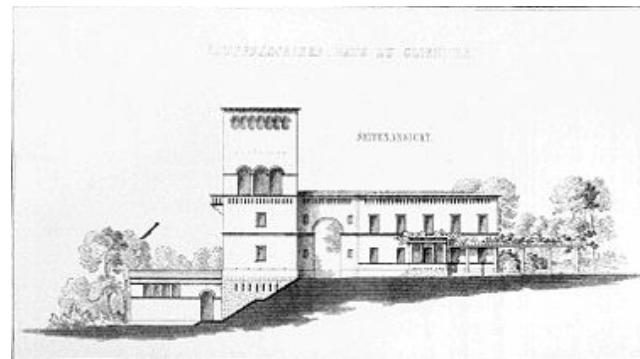
# Ingeniería de Software

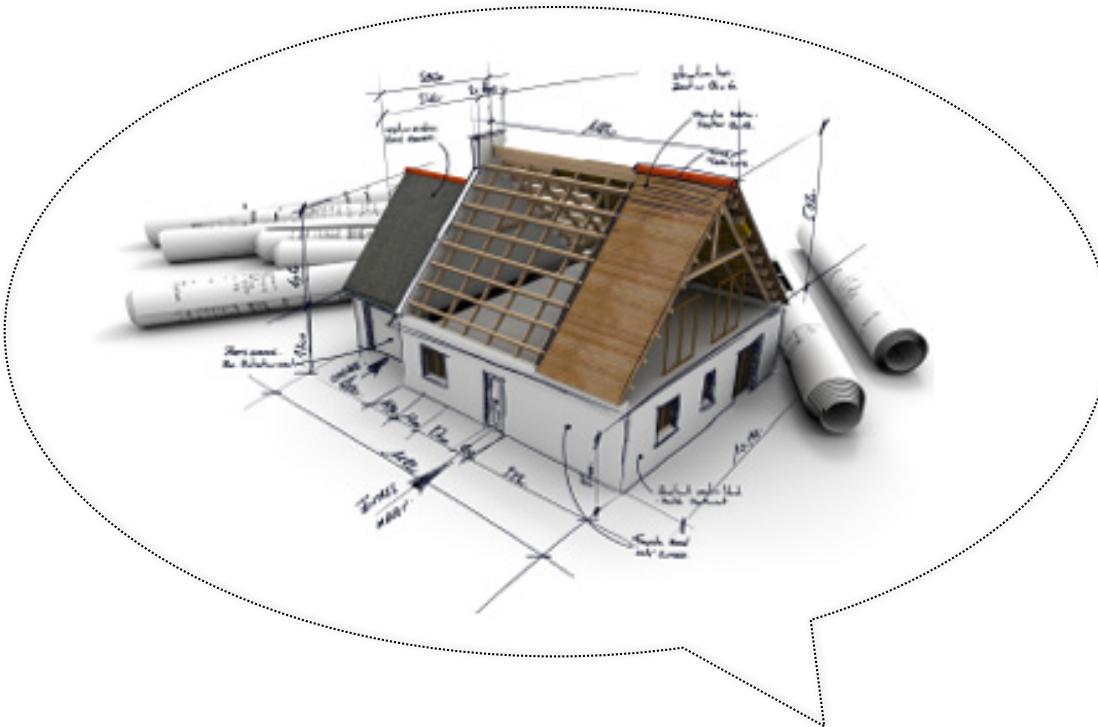
- Modelar antes de construir
- Proceso
- Planificación
- Relatos de Usuario
- Diseño
- Aseguramiento de Calidad

# Modelar



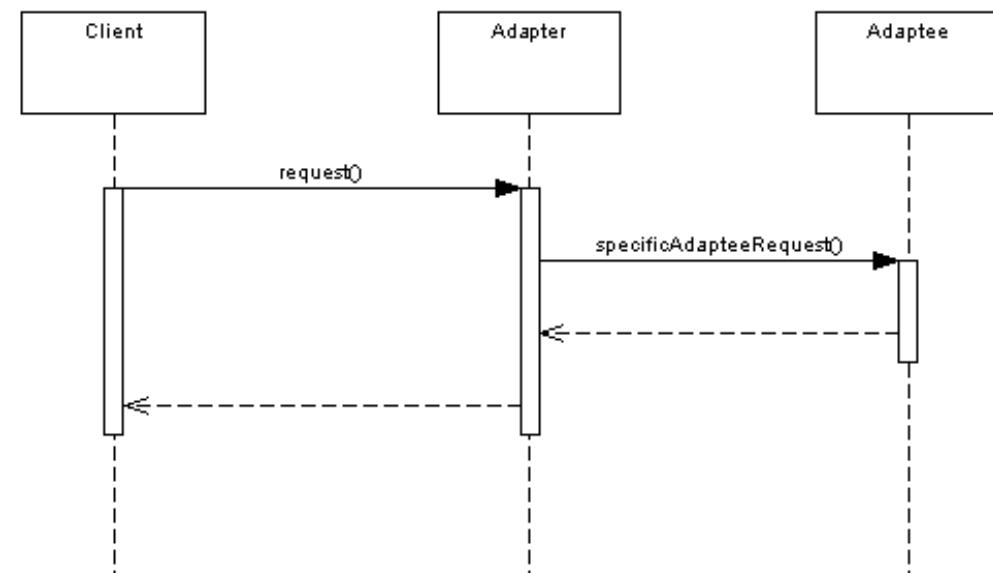
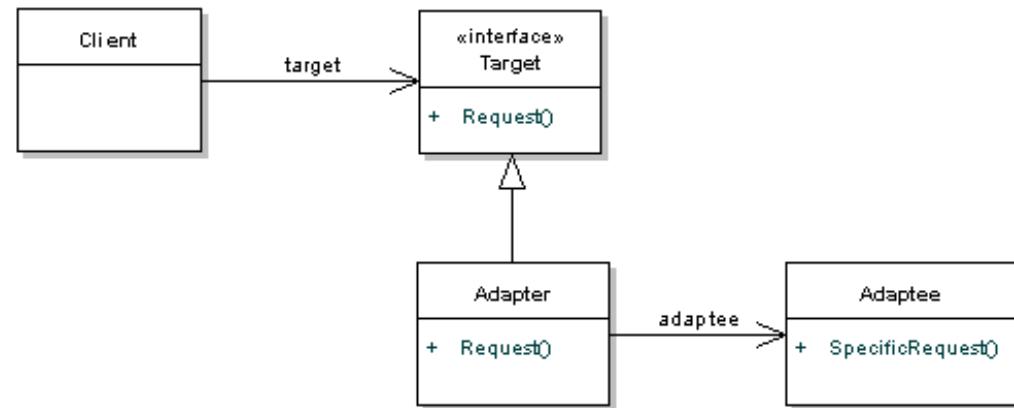
# Los Planos de una Casa

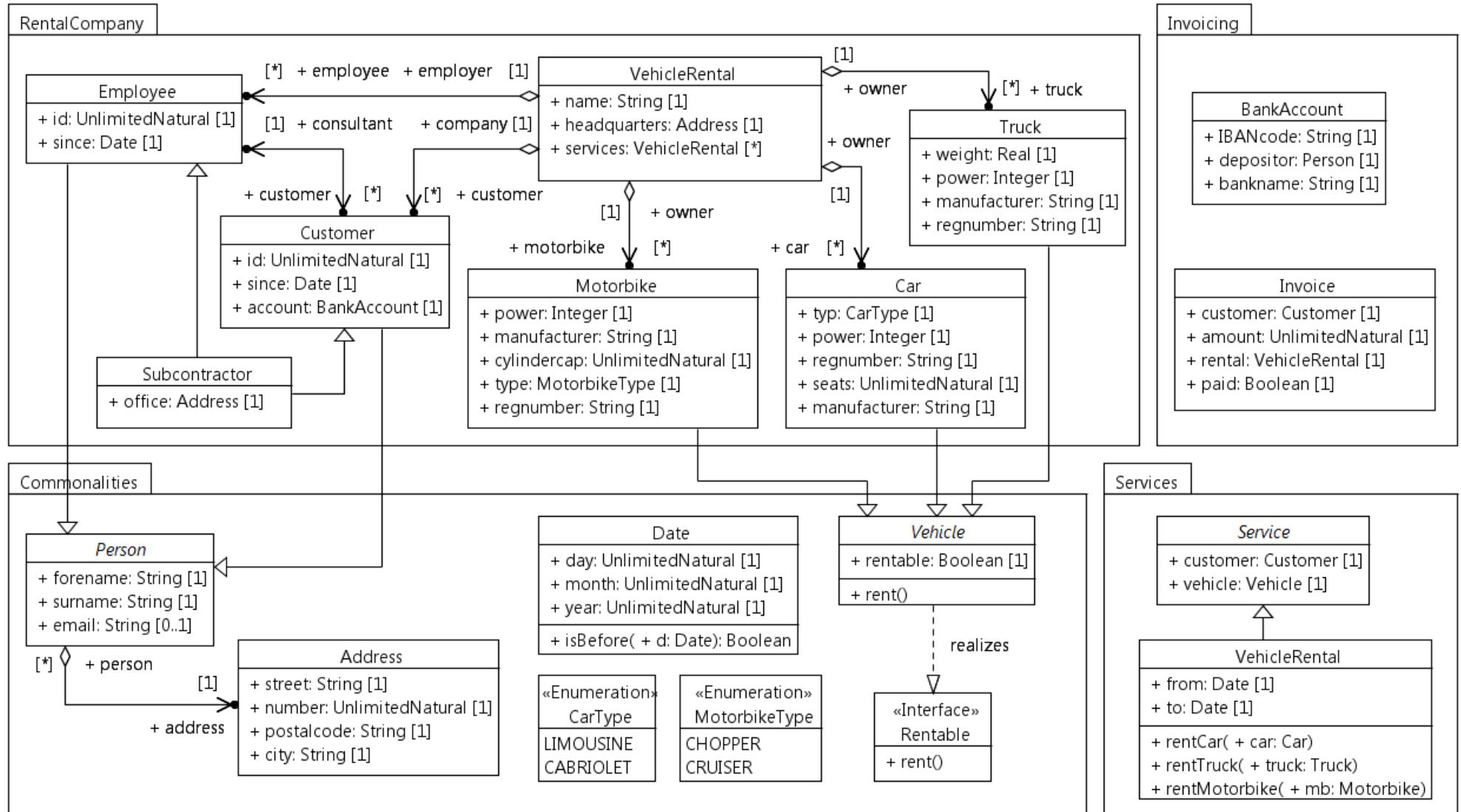




# Los Planos del Software

Un Adaptador !





# Diseño

¿ Por qué preferir una solución a otra si ambas satisfacen todos los requisitos ?

El factor clave es la mantenibilidad

¿ se puede modificar con facilidad ?

¿ se le pueden agregar nuevas funcionalidades ?



En el año 2000 la mayor parte del software tuvo que repararse porque el año consideraba solo dos dígitos

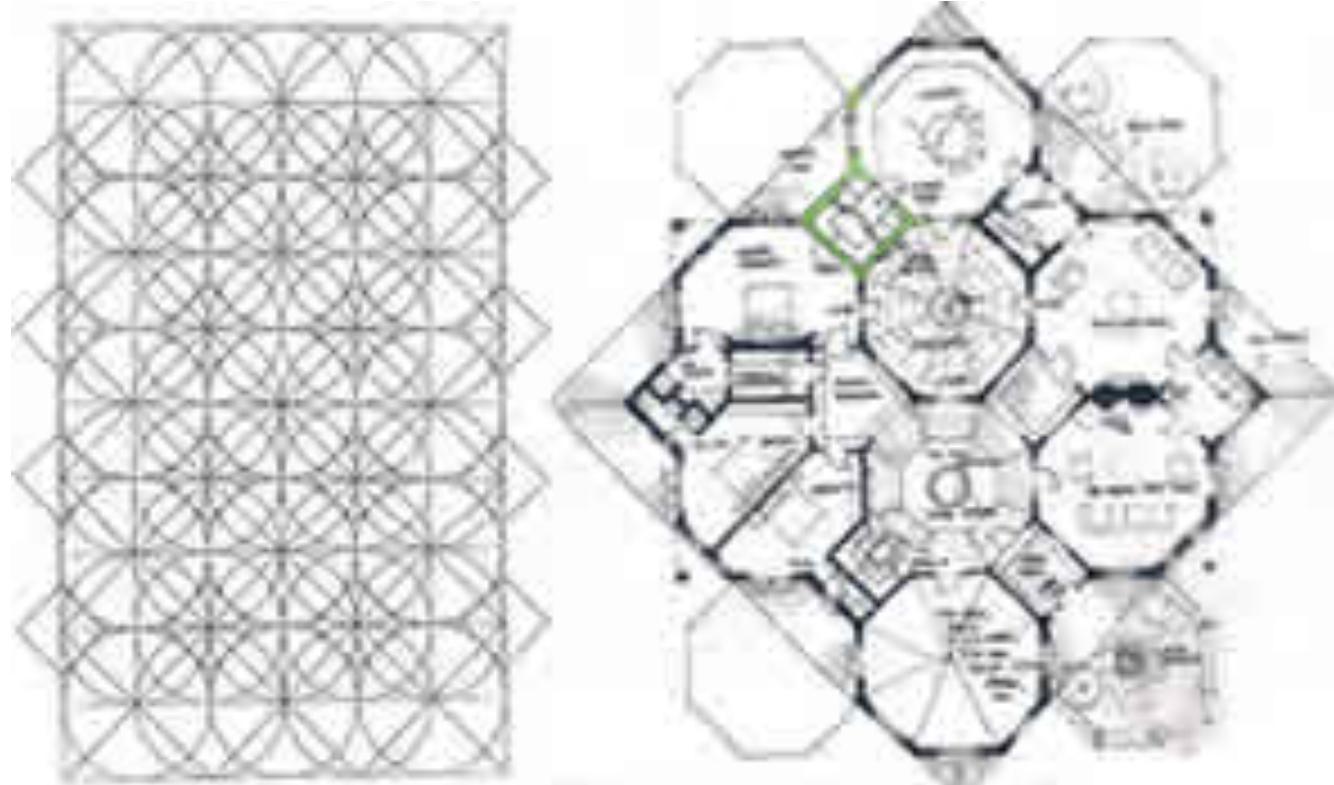
La mayoría de esos programas se escribió 20 o 30 años antes del 2000

Hoy existe software en la banca que se construyó hace 30 años

# Patrones de Diseño

- Idea sacada de la arquitectura
- Los arquitectos reusan buenas soluciones conceptuales una y otra vez
- En software hay 23 patrones que cubren la mayoría de las situaciones conocidas

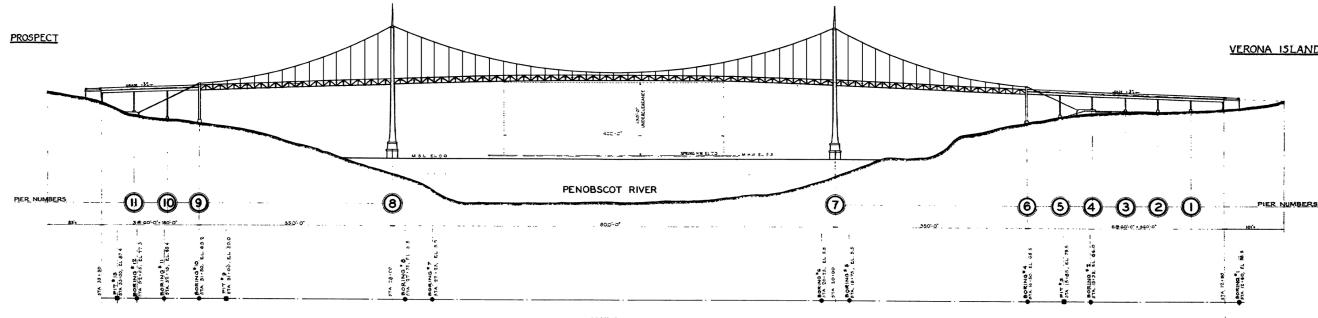
# Patrón Octogonal para un Centro de Salud



# Patrones de Diseño de Software

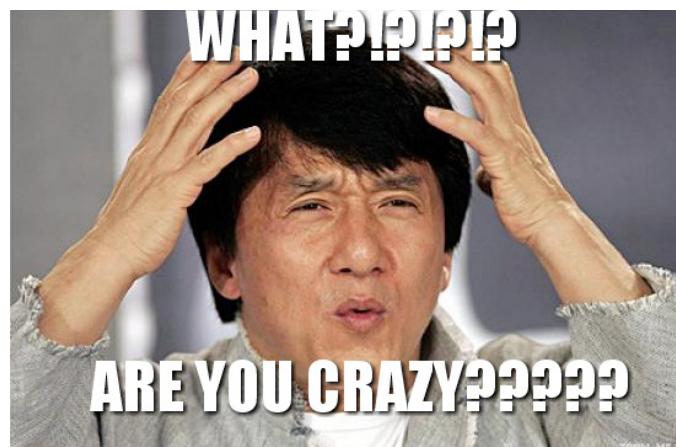
- adaptador
- fachada
- comando
- fábrica abstracta
- memento
- composite
- decorator
- ...

# Desafío de Escalabilidad



Diseño inicial: carga máxima 10 toneladas

¿ Se podrá adaptar para 1000 toneladas ?



**facebook**

Comenzó como idea de un joven en una universidad

Hoy tiene 2.000 millones de usuarios

**twitter** 

500.000.000 de tweets por día

**Google**

40.000 búsquedas por segundo

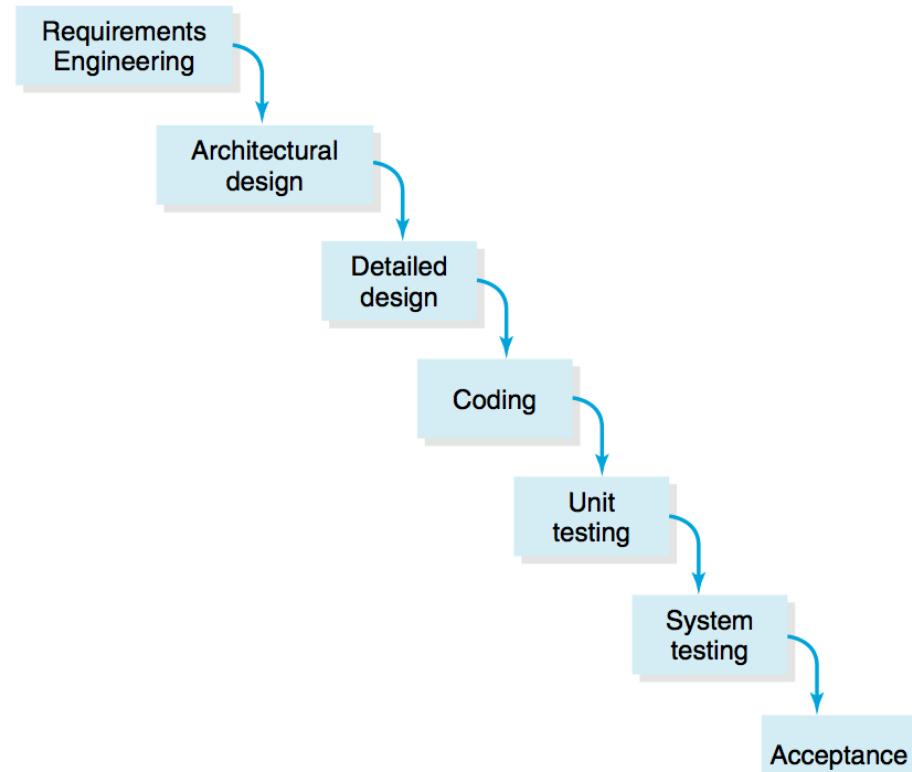
# Proceso de Desarrollo

- Estrategia para construir el software
- Quien hace qué cosa y en qué orden
- Cuales son las entradas y las salidas de cada etapa

# Proceso de Construcción

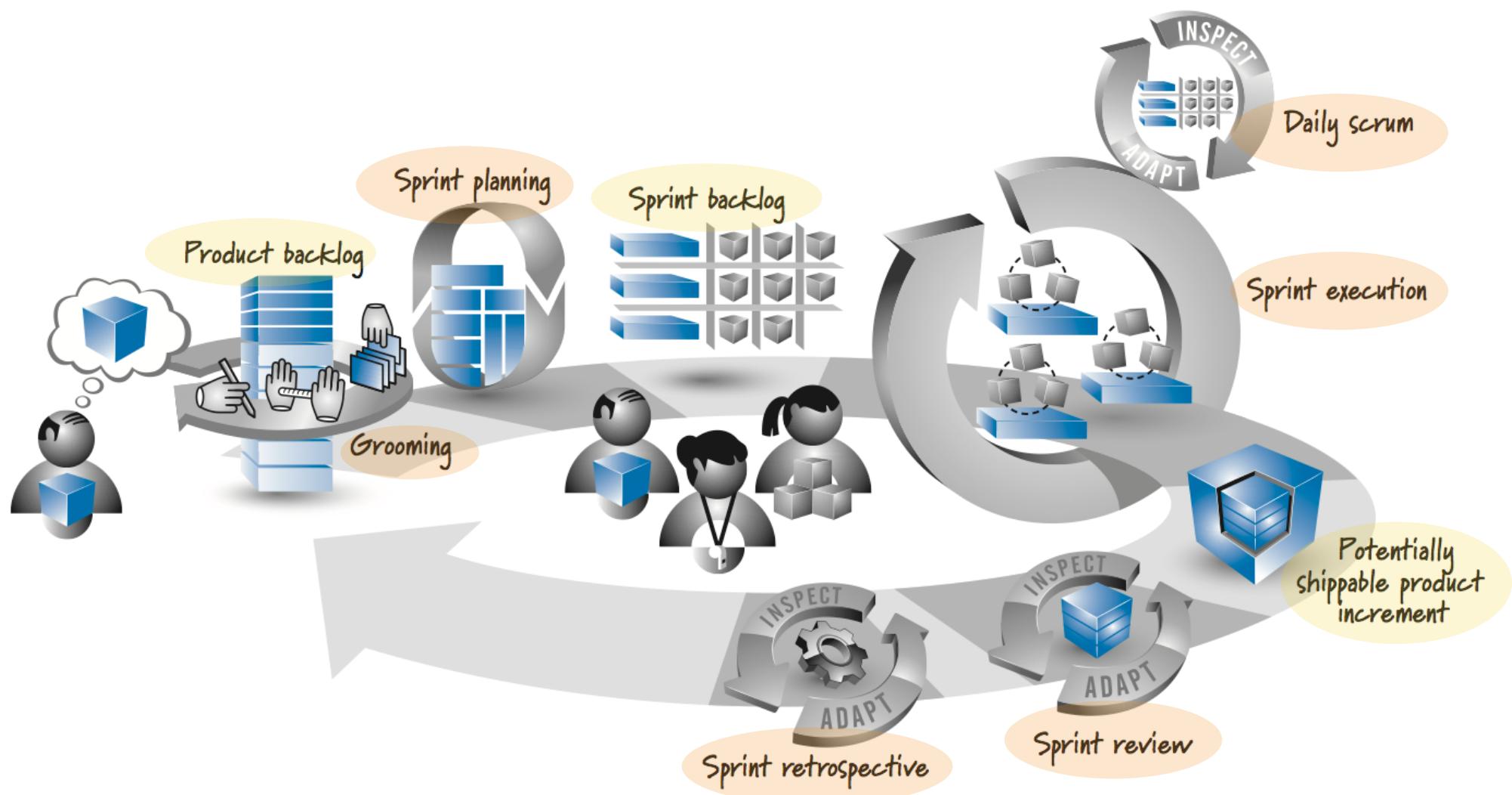
- Break Ground
- Excavation
- Foundation
  - Concrete Forms
  - Pour Foundation Wall
  - Concrete Slab Pour
    - Gravel
    - Water Barrier
    - Rigid Foam
    - Rebar (and PEX tubing for radiant floor heating)
    - Pour Concrete for Slab
- Utilities
  - Sewer, Electrical, Water, Gas
- Framing
  - Balloon Framing
  - Sheathing
  - Roof
  - Stairs
  - Windows
- Roofing
- Weather Resistant Barrier
  - Rain Screen
- Rough Plumbing
- Mechanical Systems
  - HVAC
- Lighting and Electrical
  - Ambient
  - Task
  - Accent
  - Perimeter
- Air Sealing
- Insulation
  - Spray Foam, Blown-in Insulation, or Batt Insulation
- Drywall
  - Sheetrock, Mudding, Sanding, Primer
- Siding
- Flooring
- Tiling
- Painting
- Cabinets, Shelving
- Finish Plumbing
- Finish Electrical and Lighting
- Certificate of Occupancy
- Modifications
- Moving In

# El Modelo de Cascada (Waterfall)



- Conceptualmente simple y fácil de entender
- Similar a procesos de manufactura o construcción
- Cada fase tiene un input y un output definido
- Se facilita asignar recursos
- Se puede saber cuanto llevamos y cuanto falta
- pero no funciona !!

# Scrum: Un proceso Agil



# Kanban board



# Hacer lo que realmente se necesitaba

- levantar los requisitos desde la perspectiva del usuario (relatos de usuario)
- incorporar a un representante del cliente en el equipo y hacerlo responsable del proyecto (product owner)
- aceptar que un documento siempre tendrá ambigüedades

# Ambigüedad en las especificaciones

*Customer:* Hello, I'd like to order a cake.

*Employee:* Sure, what would you like written on it?

*Customer:* Could you write "So long, Alicia" in purple?

*Employee:* Sure.

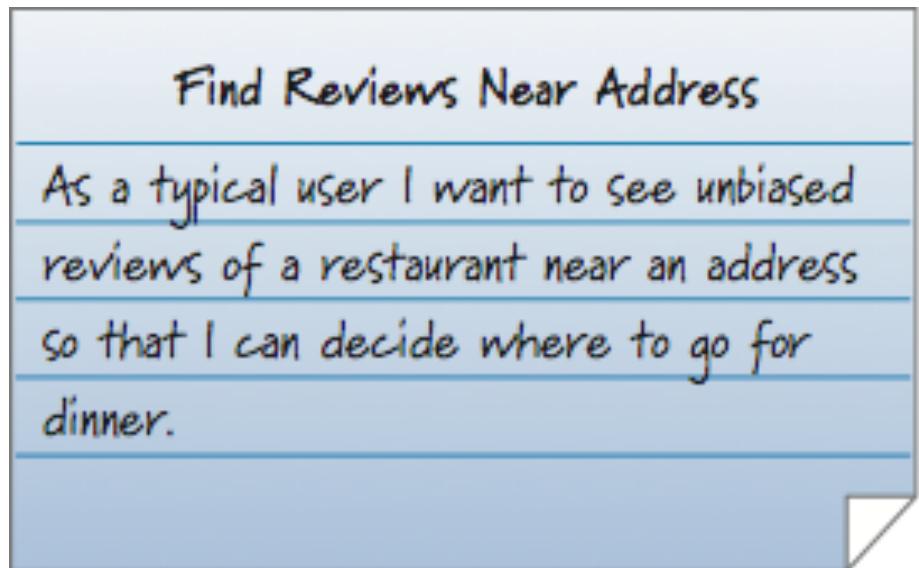
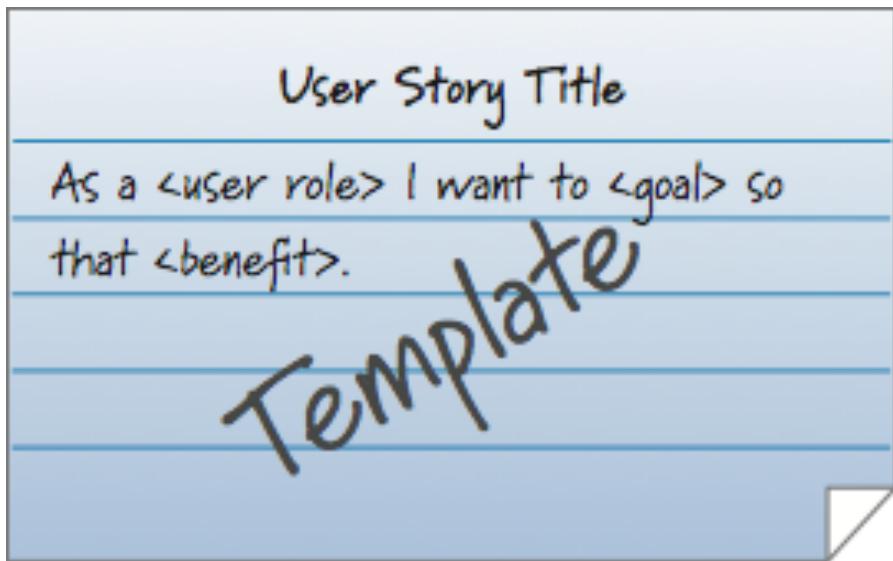
*Customer:* And put stars around it?

*Employee:* No problem. I've written this up, and will hand it to my cake decorator right away. We'll have it for you in the morning.



# Relatos de Usuario

- user role - categoría de usuario
- goal - meta de usuario, lo que quiere conseguir



# Relatos de usuario como Promesa de Conversación

- Las tarjetas son solo un punto de partida en el levantamiento de lo que se requiere
- Relato constituye una promesa de llevar a cabo una conversación (pueden ser varias) con los interesados
- Son independientes y pueden ser dimensionados en esfuerzo de desarrollo
- Priorización a cargo del product owner

# Aseguramiento de Calidad

- Errores (bugs) en el software causan miles de millones de dólares en pérdidas cada año
- Pueden afectar a vidas humanas (autos, equipos médicos, aviones)

# Software con Errores



Ariane 5  
Agencia Espacial  
Europea

*This shutdown occurred 36.7 seconds after launch, when the guidance system's own computer tried to convert one piece of data -- the sideways velocity of the rocket -- from a 64-bit format to a 16-bit format.*

*The number was too big, and an overflow error resulted.*

# La tragedia del Therac-25

- Falla de software hizo que se suministraran dosis de radiación de hasta 125 veces la requerida
- 6 casos documentados



*The defect was as follows: a one-byte counter in a testing routine frequently overflowed; if an operator provided manual input to the machine at the precise moment that this counter overflowed, the interlock would fail*

# Métodos de la Ing de Software

- ✓ programación en parejas
- ✓ revisión de código
- ✓ escribir los tests antes del código
- ✓ automatizar los tests
- ✓ cada vez que se hacen modificaciones correr hacerlas pasar por la batería de tests
- ✓ rigurosa etapa de testing final

# Programación en Pares

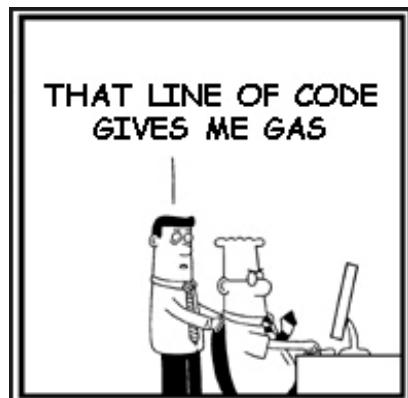




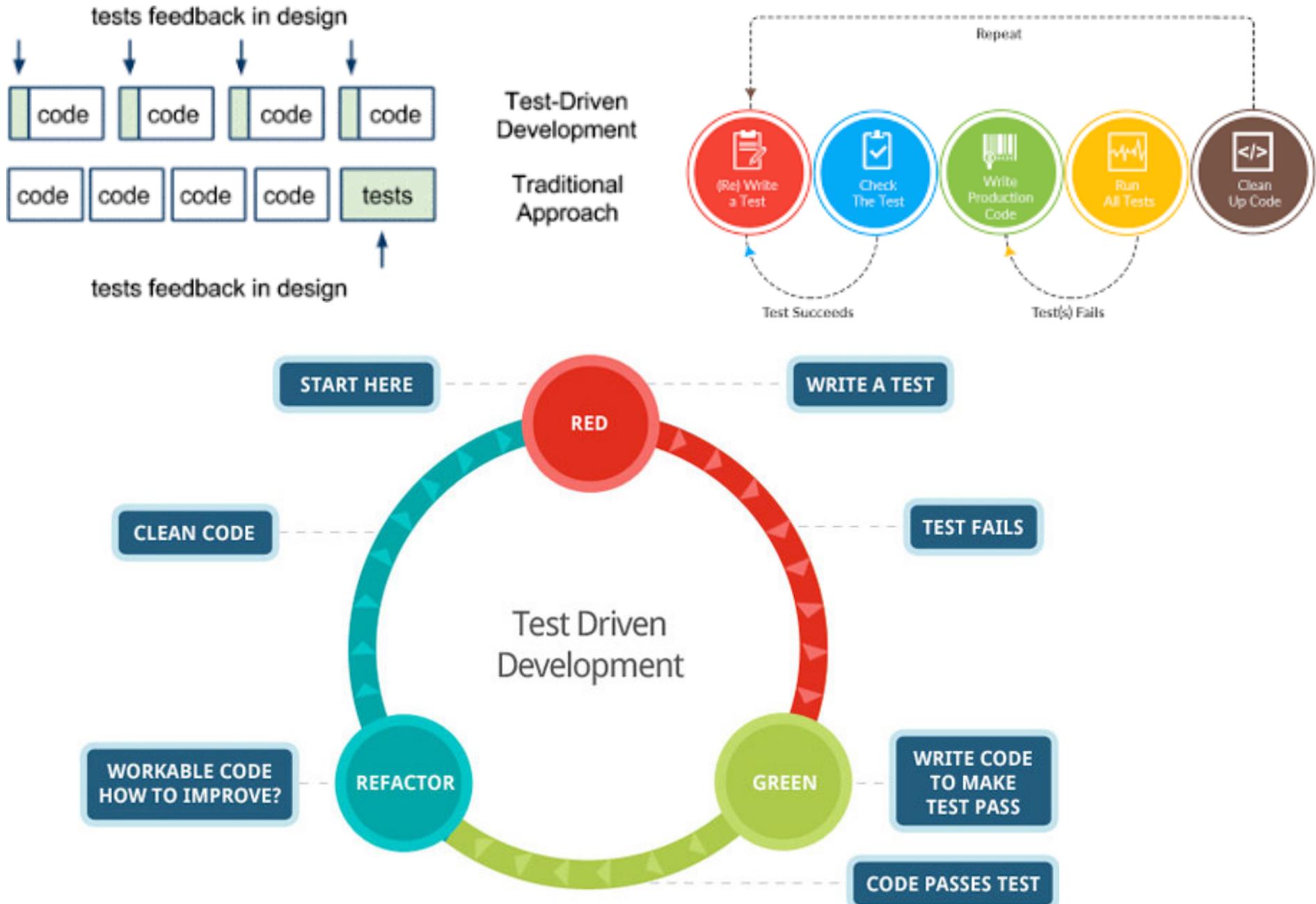
# Code Review

- Forma demostradamente efectiva para producir código de calidad
- Código es revisado por equipo de pares con el objetivo de detectar la mayor cantidad de defectos
- Equipo revisor se prepara antes de sesión de revisión
- Pueden usarse checklists para búsqueda mas dirigida

## Code Review

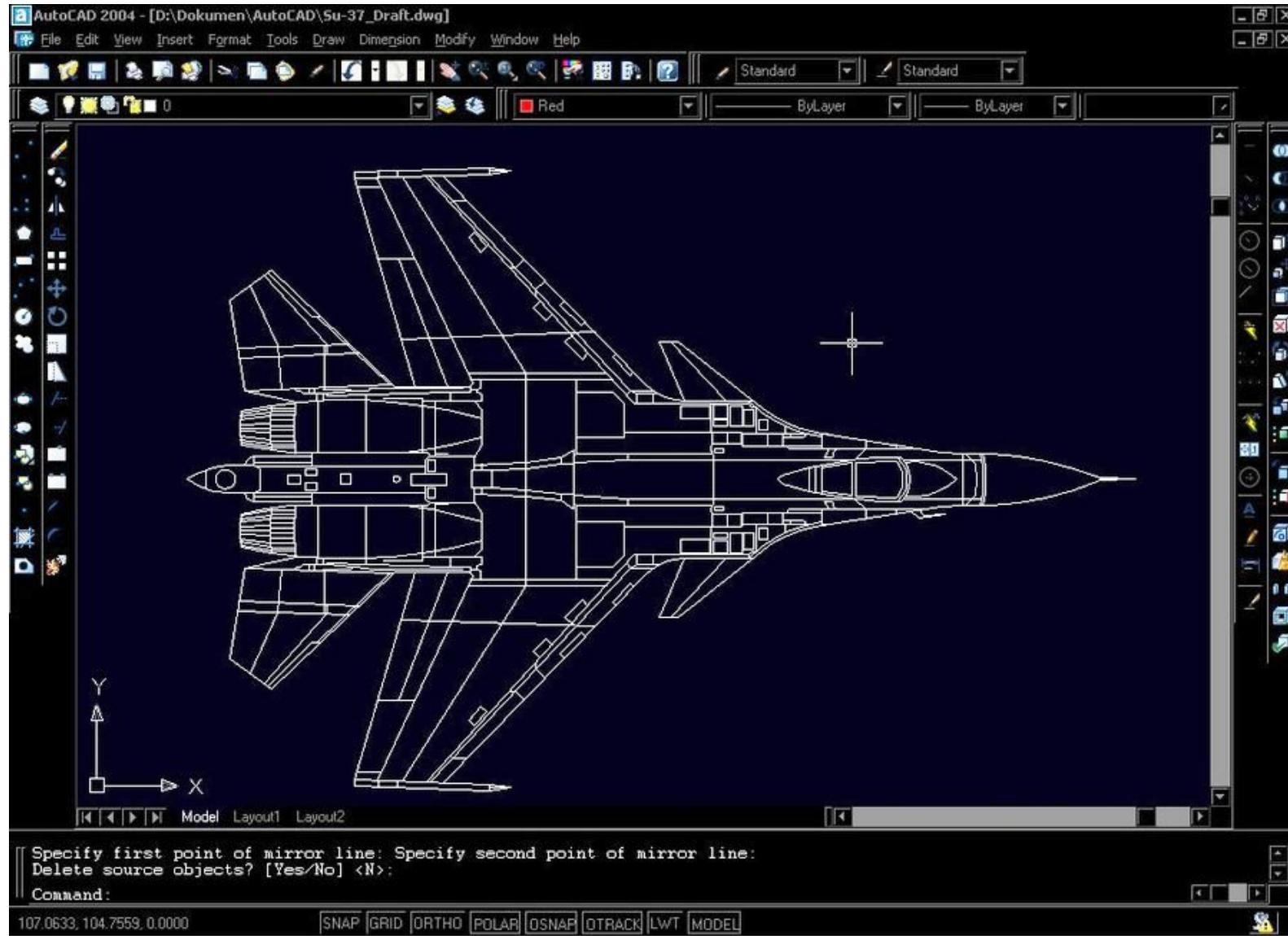


# Test Driven Development

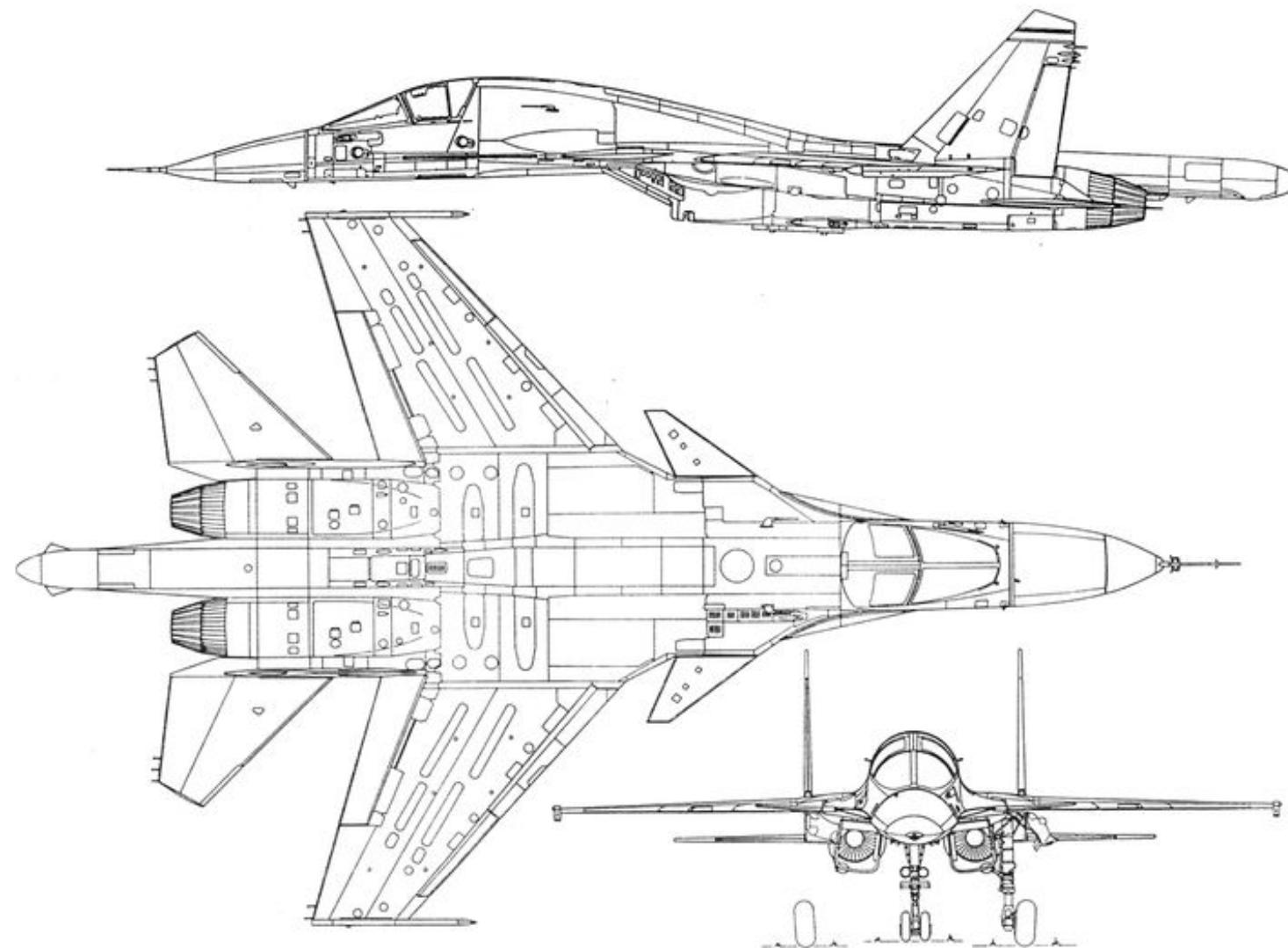


en resumen ...

# Construir un software para diseñar aviones ...



Puede ser tan complejo  
como construir el avión

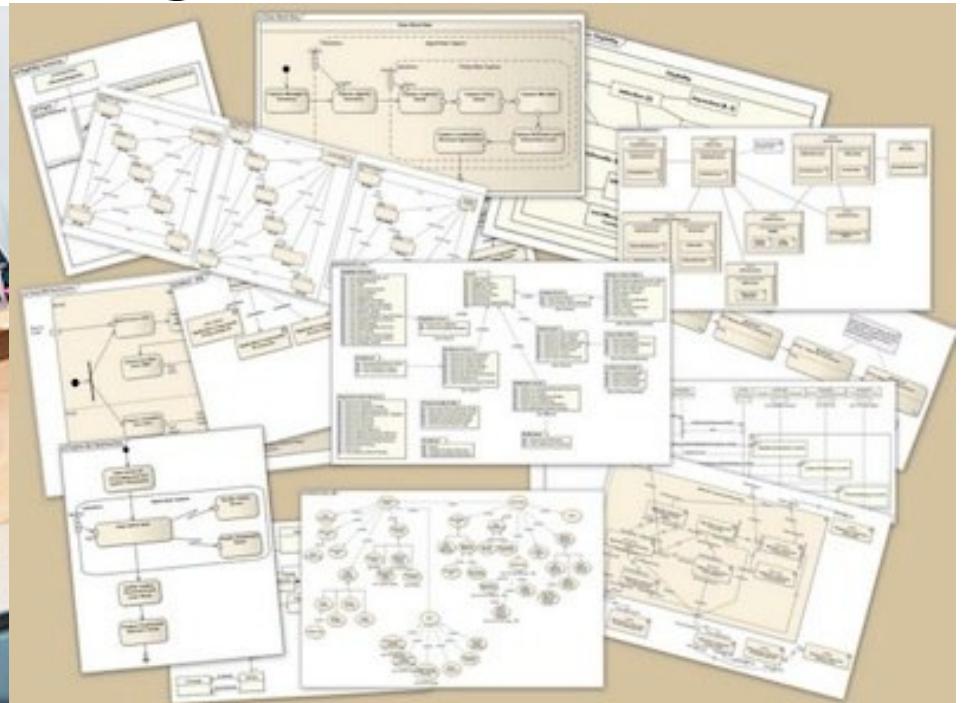


# Los métodos de la Ing de Software nos ayudan

- ✓ procesos ágiles (iterativos e incrementales)
- ✓ relatos de usuario (requisitos desde la perspectiva de los involucrados)
- ✓ modelar (lenguaje gráfico estándar UML)
- ✓ diseñar para reuso y escalabilidad (patrones)
- ✓ disminuir el número de errores (TDD, prog en pares, revisiones de código)

# El mundo funciona con software !!

se necesitan miles de ingenieros ...



# En la UC te ofrecemos 3 sabores

- Ingeniero Civil Industrial Computación
- Ingeniero Civil Industrial Tecnologías de Información
- Ingeniero Civil Computación

# La ruta lógica ...

- ✓ Major en Computación
- ✗ Major en Investigación Operativa

# Preguntas