



# Reflexión inicial ...

Si en su compañía actual se desarrollara software para dosificación automática y suministro de medicamentos ....



Estaría tranquilo de recibir un tratamiento utilizando una máquina que opera con el software desarrollado en su empresa ?  
(con los procedimientos y métodos que actualmente utilizan)

# Reflexión inicial ...

Si en su compañía actual se desarrollara software de navegación para aviones ....



Estaría tranquilo de volar en un avión que utiliza el software desarrollado en su empresa ?  
(con los procedimientos y métodos que actualmente utilizan)

# Por que es VITAL la calidad?

En sistemas críticos, un *error* puede traer como consecuencia, desde pérdidas de reputación, de mercado, de dinero, hasta vidas humanas, y/o sanciones económicas, incluso penales severas.



**American Airlines** Flight 965, a Boeing 757 registered N651AA, was a scheduled flight ... Airport in **Colombia**, which crashed into a mountain in Buga, **Colombia** on ... and determined it was caused by navigational errors. ....

Otros en:

<http://www.cs.tau.ac.il/~nachumd/horror.html>

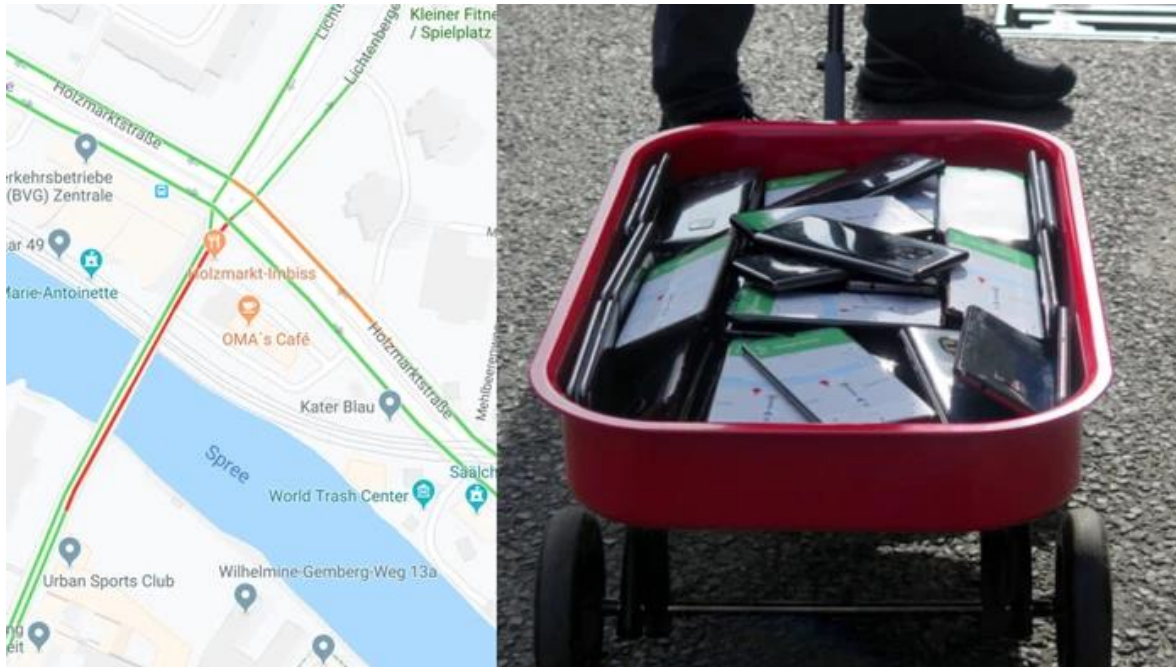
# Por que es VITAL la calidad?

Retrasos en salidas de vuelos de American Airlines debido a falla en el software de los dispositivos Ipad que almacena información sobre rutas aereas.



<https://www.youtube.com/watch?v=9KUMmHTLonQ>

# Reflexión ...



**This Man Created Traffic Jams on Google Maps Using a Red Wagon Full of Phones**



# La llamada “Crisis del Software”

- ¿Por qué toma tanto tiempo desarrollar software?
- ¿Por qué es tan elevado su costo?
- ¿Por qué no se puede entregar programas libres de errores?
- ¿Por qué es tan costoso su mantenimiento?
- ¿Por qué resulta tan difícil constatar el progreso del desarrollo de software?

La crisis del software aparece en la segunda era de la evolución de los sistemas informáticos (alrededor de 1968).

Las actividades de mantenimiento del software (corrección de fallas, modificación por cambios de requerimientos de usuarios, y adaptación a nuevos dispositivos) y el esfuerzo empleado en dicho mantenimiento comenzó a absorber recursos en una medida alarmante.

# La llamada “Crisis del Software”

1970's - Crisis del software

- Exceso en la capacidad para desarrollar software grande y complejo, con la tecnología existente.
- Costo de las actividades de mantenimiento del software superior a los costos de nuevos desarrollos.
- Inicio el descenso en los costos del hardware.
- Aumento de proyectos que fallaron a causa del software.

## **Soluciones**

- Mejores lenguajes de programación (PL/1, APL, Jovial)
- Lenguajes formales para especificar requerimientos
- Lenguajes estructurales y multitareas.



# La llamada “Crisis del Software”

1985

- Crecimiento de la Industria del Software a \$ USD 300 MM
- Reconocimiento de la Ingeniería de Software como una disciplina de la Ingeniería.
- Los costos del hardware continúan disminuyendo.
- Surgen las herramientas CASE:
  - Implementación de un proceso específico de desarrollo
  - Representar gráficamente los diseños del software
  - Fácil de mantener, validar y entender.

## **Soluciones**

- Muchas compañías se dieron cuenta que necesitaban realizar mejoras significativas en su proceso de desarrollo.

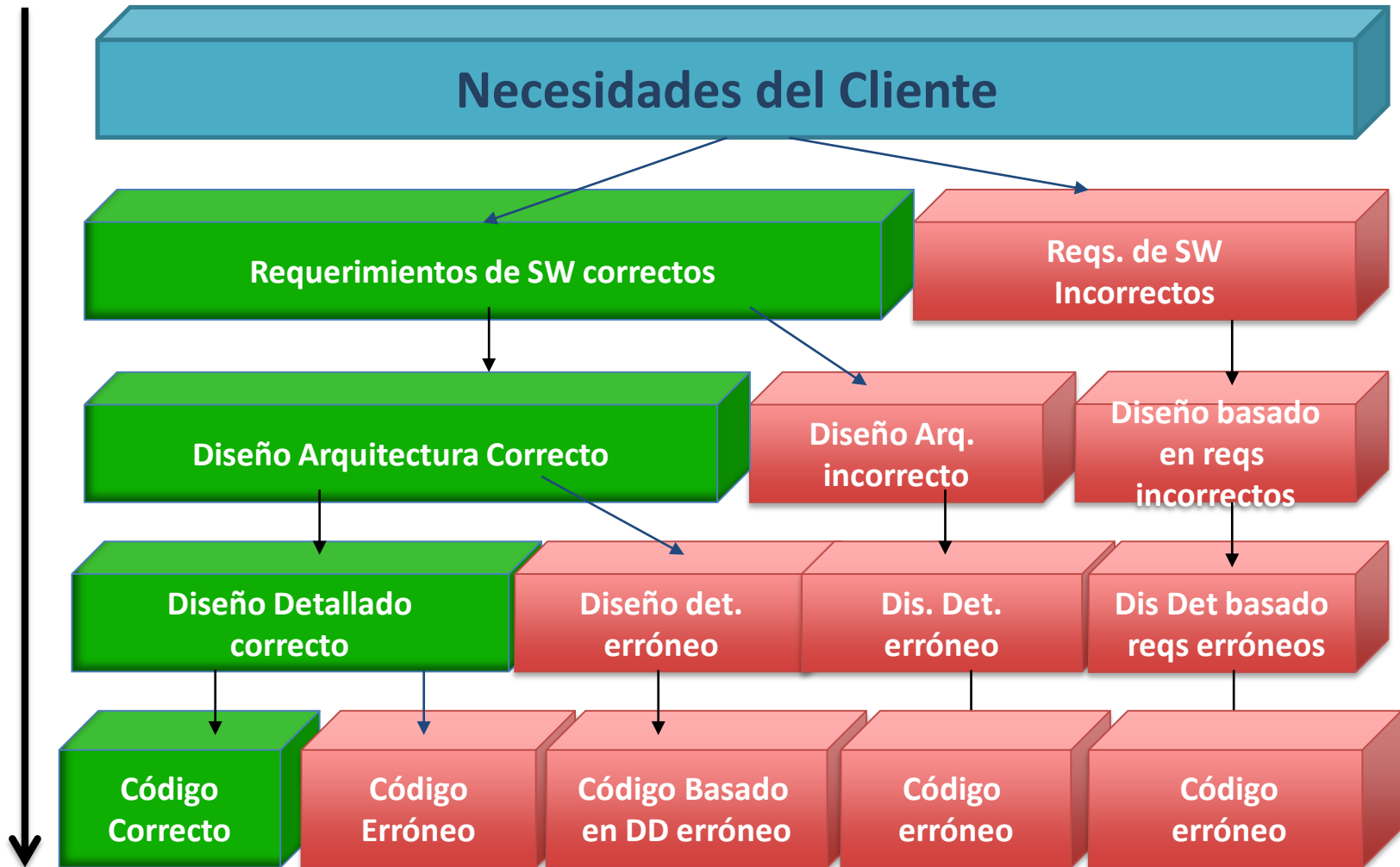
**Que es Calidad de  
Software?**



# Estándares de Calidad de Software ISO 25000



# Ciclo de vida de desarrollo



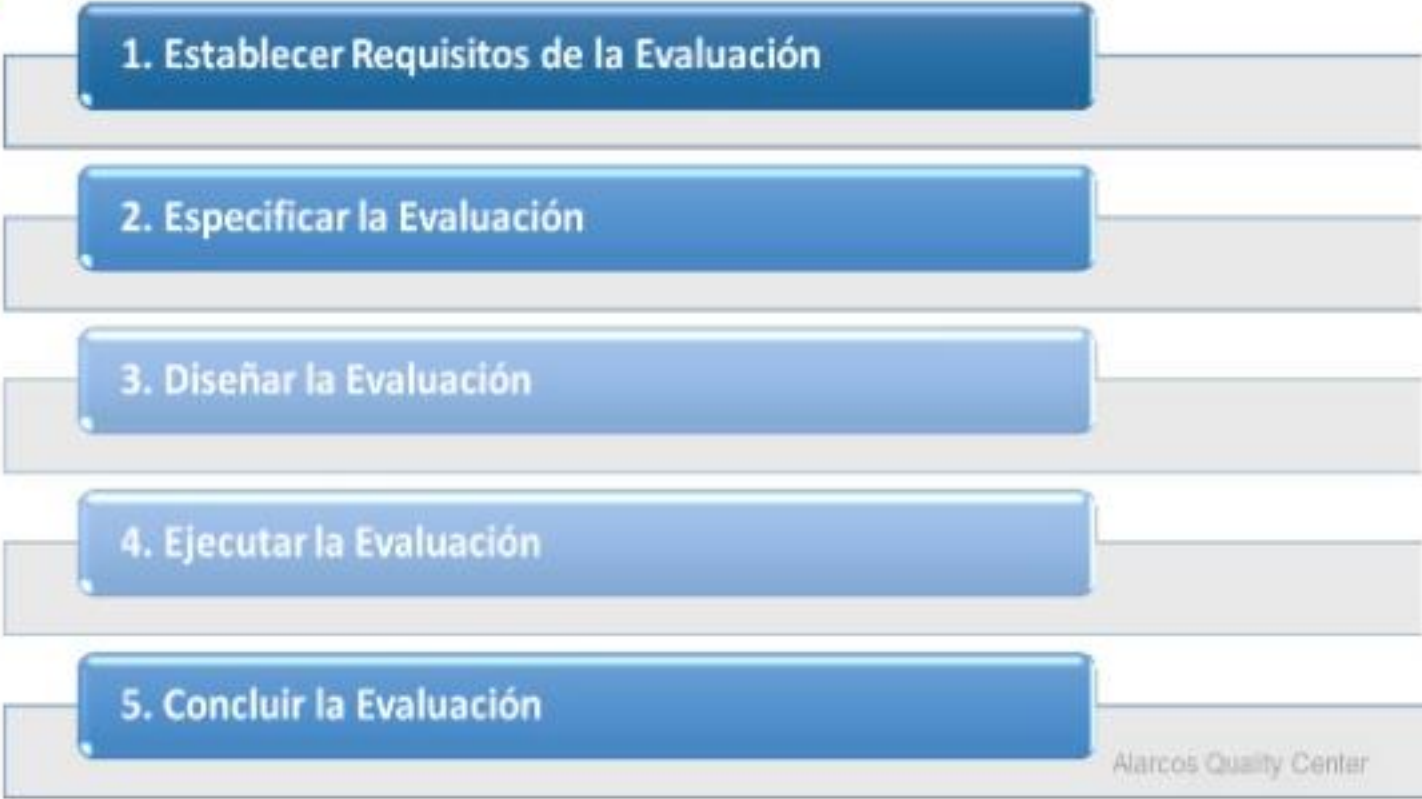
# Estándares de Calidad de Software ISO 25000



**Que es Calidad de  
Software?**



# Estándares de Calidad de Software ISO 25000



The diagram illustrates the five steps of the ISO 25000 software quality standards process. It consists of five horizontal blue bars with rounded ends, each containing a number and a description of a step. The bars are stacked vertically, with each subsequent bar starting further to the left, creating a descending staircase effect. The background is a light gray with a subtle grid pattern.

1. Establecer Requisitos de la Evaluación

2. Especificar la Evaluación

3. Diseñar la Evaluación

4. Ejecutar la Evaluación

5. Concluir la Evaluación

Aiacos Quality Center



# Estándares de Calidad de Software ISO 25000



# Bibliografía Principal

- International Organization for Standardization, ISO 9001:2000, Quality Systems-model for Quality Assurance in Design, Development, Production Installation and Servicing. ISO. 2000.
- Norma técnica Colombiana NTC-ISO 9004. Sistema de Gestión de Calidad. Directrices para la mejora del desempeño.
- Norma técnica IEC-ISO 12207. Tecnología de la información. Procesos del ciclo de vida del software.
- ISO/IEC DTR 29110-5-1-2:2010(E), ISO/IEC JTC 1/SC 07/WG 24
- Ingeniería de Software — Perfiles del Ciclo de Vida para organizaciones Pequeñas (OPs) — Parte 5-1-2: Guía de Administración e Ingeniería –Perfil Básico OP.
- An Update and Preliminary Results. Dennis R. Goldenson Diane L. Gibson. October 2003
- RPM-AEMES, VOL. 4, No 1 Enero 2007, ISSN: 1698-2029. DESDE ISO 9001 HACIA CMMI, PASOS PARA LA MEJORA DE LOS PROCESOS Y MÉTRICAS
- CMMI for Development, Version 1.3  
<http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>
- Demonstrating the Impact and Benefits of CMMI®:
- Rolando Armas Andrade<sup>1</sup>, Arturo Chamorro Gómez<sup>2</sup>, Maite Montes Beobide<sup>2</sup>, José Antonio, Gutierrez de Mesa, Departamento de Ciencias de la Computación. Escuela Superior de Informática.Universidad Alcalá Edif. Politécnico. Campus Universitario.

