Tecnológico de Monterrey

Introducción a la Calidad en el Software

¿Qué es Calidad?

Según la Norma ISO 8402 (vocabulario):

"Es la totalidad de propiedades y características de un producto o servicio que le confieren la capacidad de satisfacer las necesidades expresas o implícitas"

Calidad es…

Superar las expectativas de nuestro cliente

¿Qué esperan nuestros clientes?

Un software:

- Entregado a tiempo
- Dentro del presupuesto
- Con la funcionalidad que necesita
- Que no falle (cero defectos)

Retos a Nivel Mundial

- En una conferencia se concluyó que el desarrollo de software era…
  - Poco confiable
  - Entregas tardías
  - Altos costos de modificación
  - Retos para su mantenimiento
  - Desempeño inadecuado
  - Excesos en presupuesto

OTAN
International Software Engineering Conference

Munich, Alemania, ***1968***

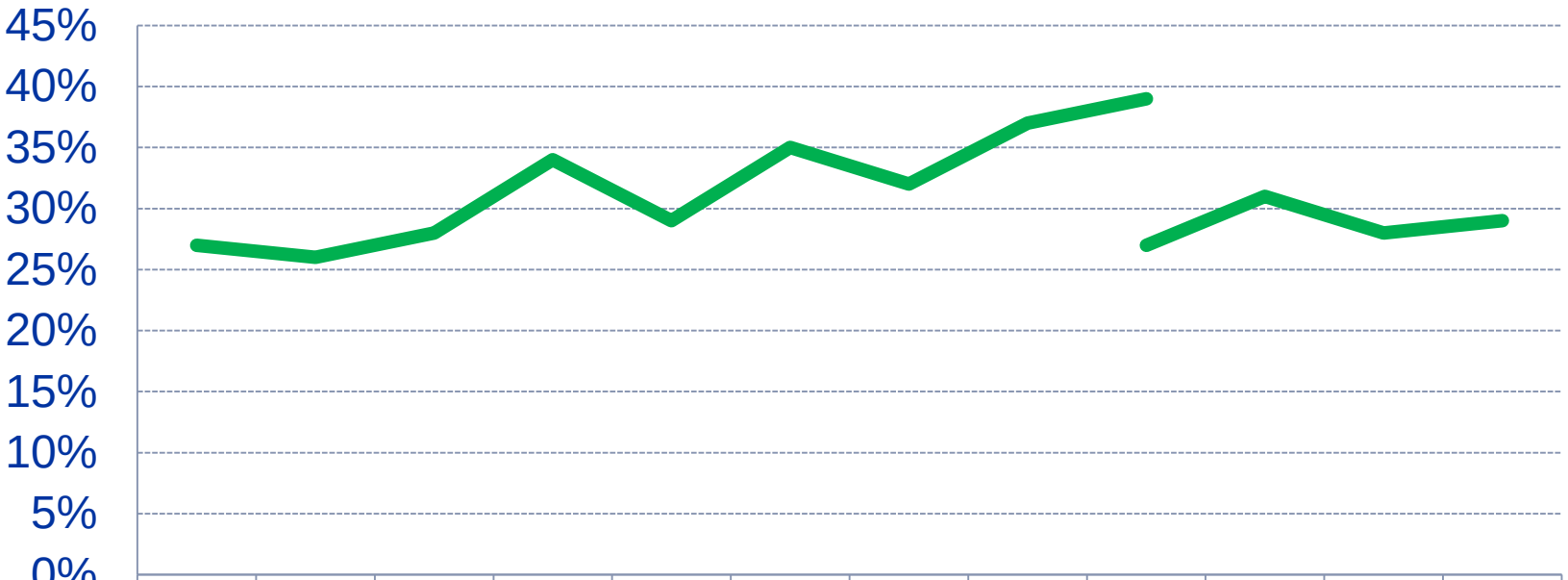
40 años después...

- La industria mundial del desarrollo del software continúa en problemas.
- En el 2008:
  - Sólo 32% de los proyectos fueron exitosos (en costo, a tiempo y con la funcionalidad prometida)
  - 24% de los proyectos se cancelaron por problemas
    - Costo $81,000 millones USD
  - 44% de los proyectos se entregaron tarde o con menos funcionalidad
    - Costo del 189% del estimado inicial
    - Este exceso cuesta $59,000 millones USD

Fuente: Chaos Report 2009, Standish Group

No existe una mejora sustancial

% de proyectos “exitosos”



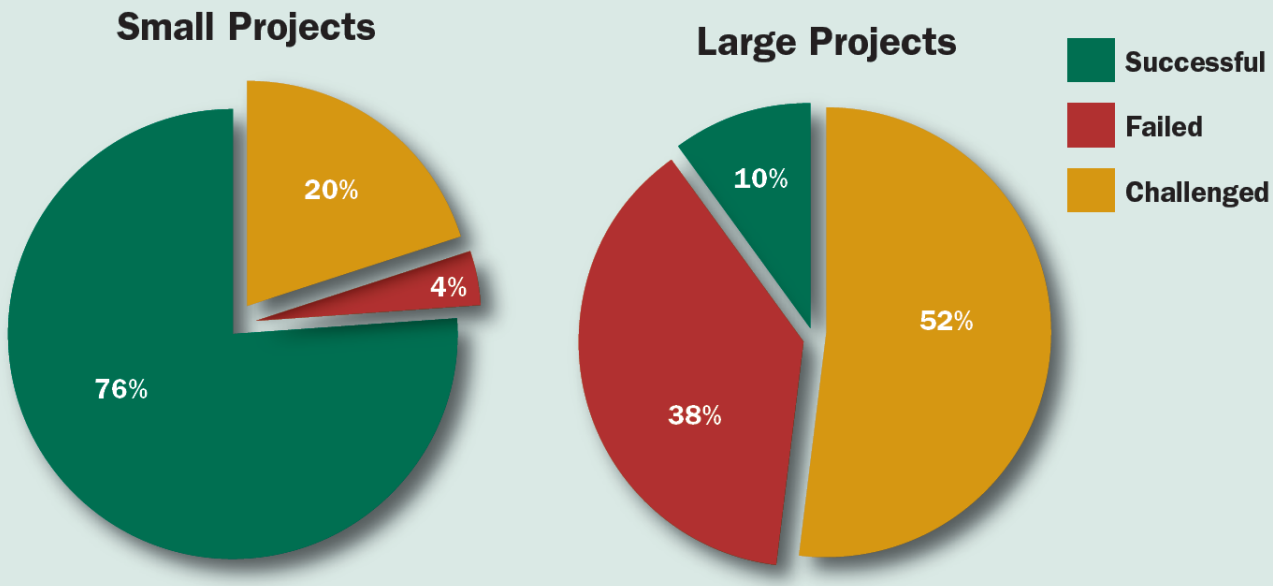
1996	1998	2000	2002	2004	2006	2008	2010	2012	2013	2014	2015
27%	26%	28%	34%	29%	35%	32%	37%	39%			
								27%	31%	28%	29%

Fuente: Chaos Manifesto 2013 & 2015, Standish Group

No somos buenos con proyectos grandes

CHAOS RESOLUTION BY LARGE AND SMALL PROJECTS

Project resolution for the calendar year 2012 in the new CHAOS database. Small projects are defined as projects with less than $1 million in labor content and large projects are considered projects with more than $10 million in labor content.

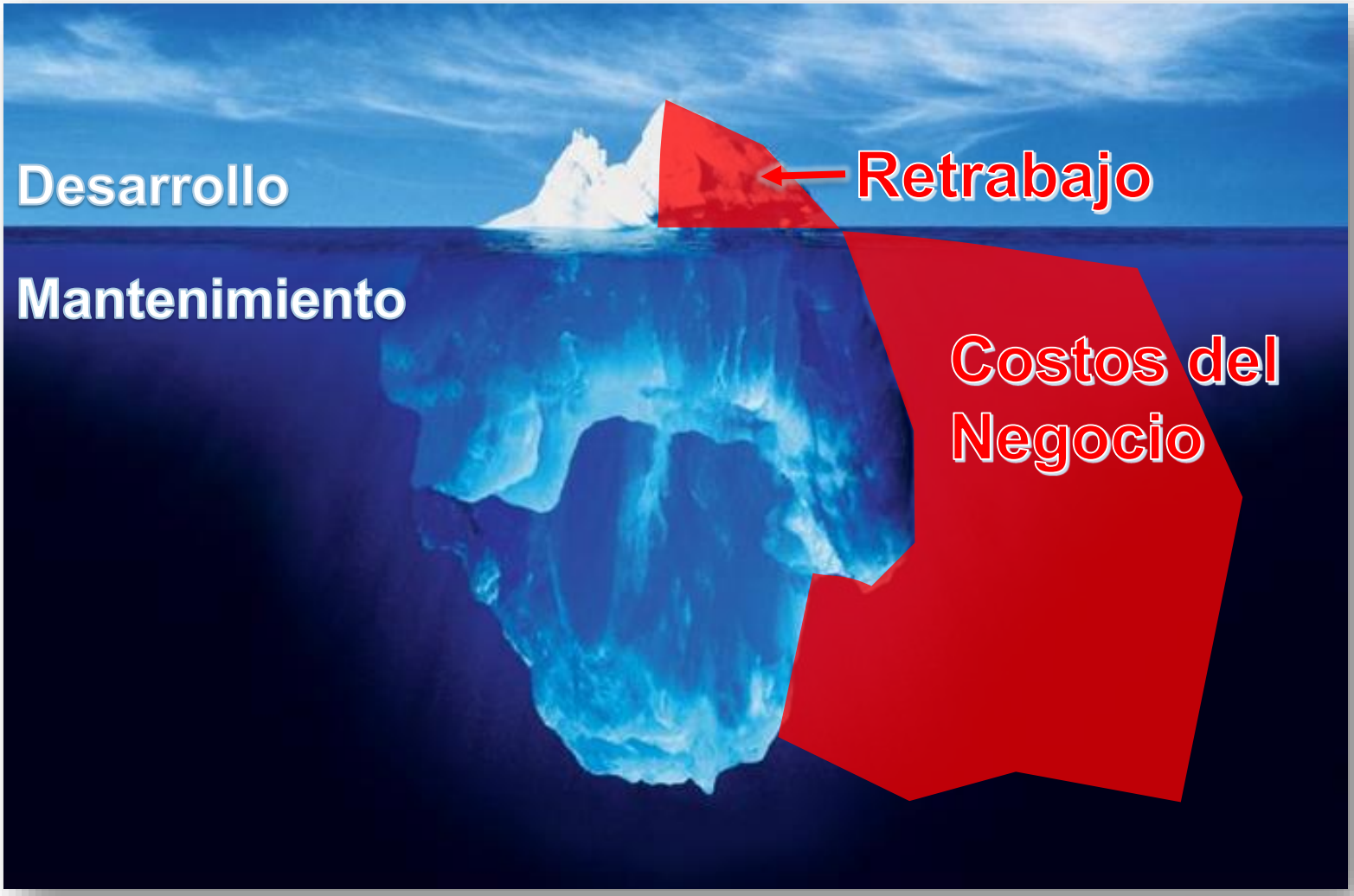


Fuente: Chaos Manifesto 2013, Standish Group

¿Proyectos exitosos?

- ¿Y los defectos entregados?
- Las empresas con CMMI nivel 5 entregan al cliente 1.05 defectos/KLDC
  - Fuente: Capers Jones
- ¿Eso es exitoso?
  - Sistema con 50 KLDC → ¡53 defectos!
  - ¿Marcapasos con 20 KLDC?
  - ¿Automóvil con 100,000 KLDC?

¿Y los costos escondidos?



Algo debemos estar haciendo mal

Industria de Manufactura (aprendizaje de Japón, 1950)	Industria del Desarrollo de Software
La calidad es lo más importante (luego fecha de entrega)	La fecha de entrega es lo más importante
La calidad depende de todos (en particular de la línea de producción)	La calidad depende de quien prueba
Decisiones basadas en datos	Decisiones basadas en intuiciones (no se cuenta con datos)
Enfoque al proceso	Enfoque al producto

¿Cómo medir la calidad del SW?

Atributos de calidad del SW

- Desempeño
- Usabilidad
- Mantenibilidad
- Seguridad
- Escalabilidad
- Etc. …idad

Definamos “defecto”

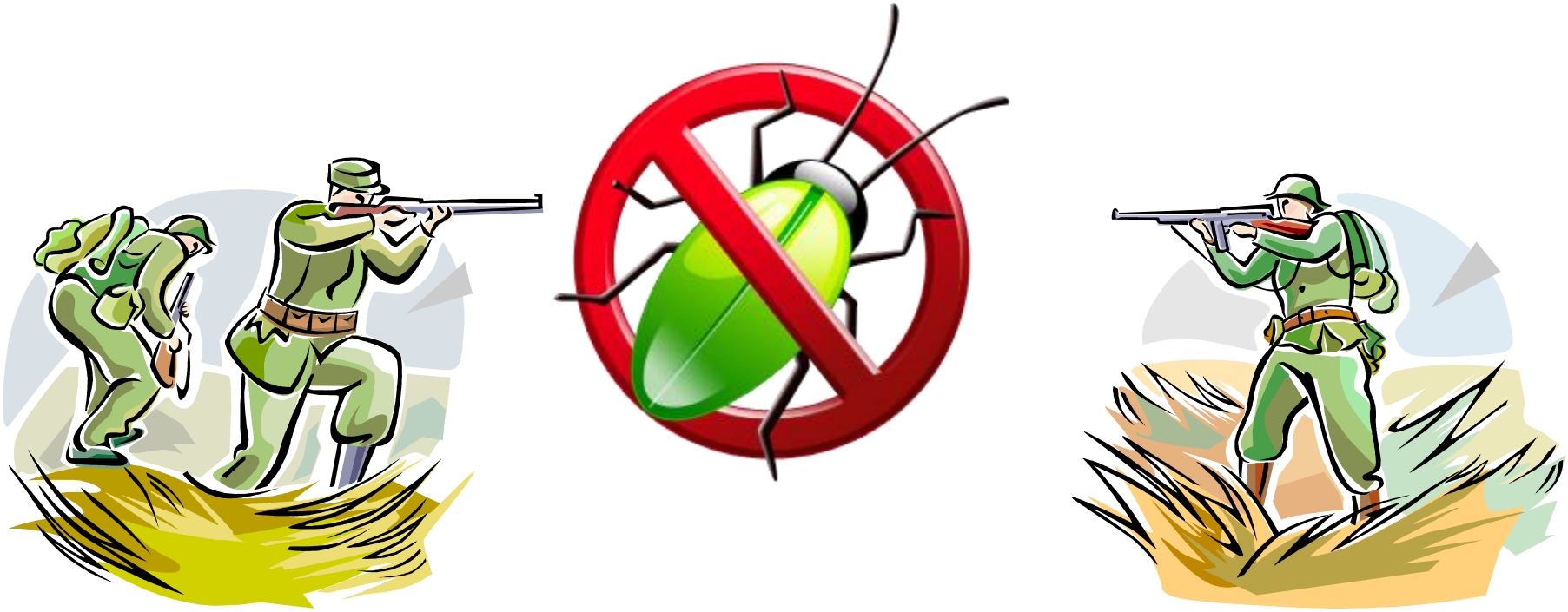
- Es todo aquello que tengo que cambiar en mi programa porque no quedó como el cliente lo quería
  - Defecto = retrabajo
  - Defecto > bug
  - Incluye mala usabilidad, pobre desempeño, huecos de seguridad, etc.
- ¿Qué significaría tener “0 defectos”?
  - 100% de trabajo productivo (cero retrabajo)
  - Producto entregado bien “a la primera”

Forma de medir la calidad del SW

- Calidad *podría* medirse como
  - "Cantidad de defectos entregados al cliente"

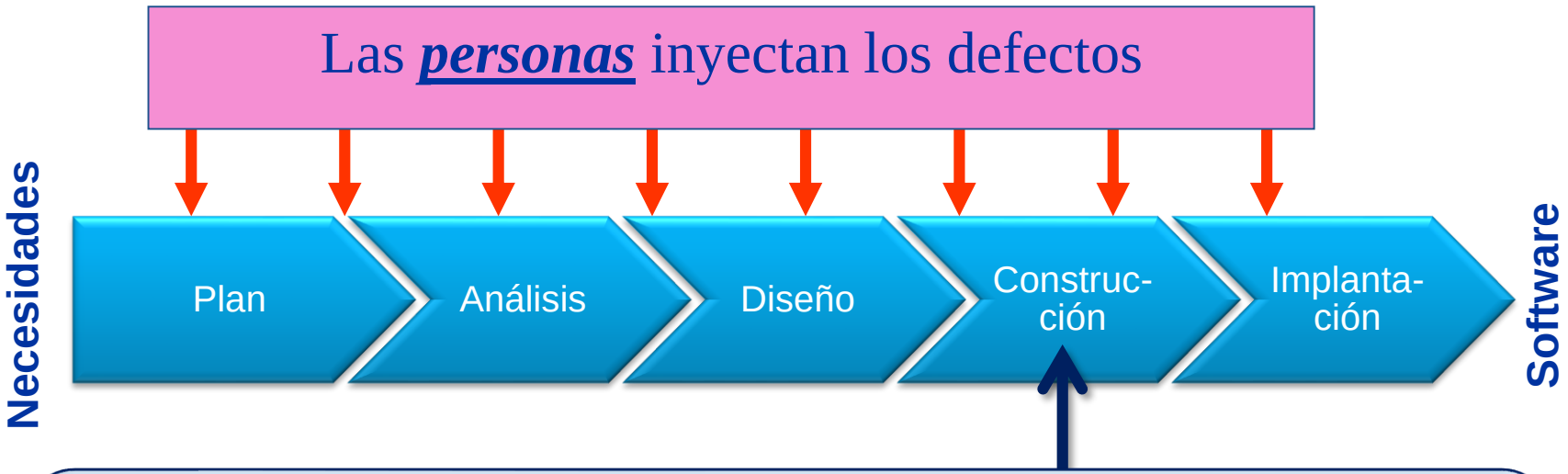
- Pero queremos poder comparar entre productos, entonces mediremos:
  - "**Densidad de defectos entregados al cliente**"
  - Normalmente → **defectos/KLDC**

Hablemos sobre los defectos



Tecnológico de Monterrey

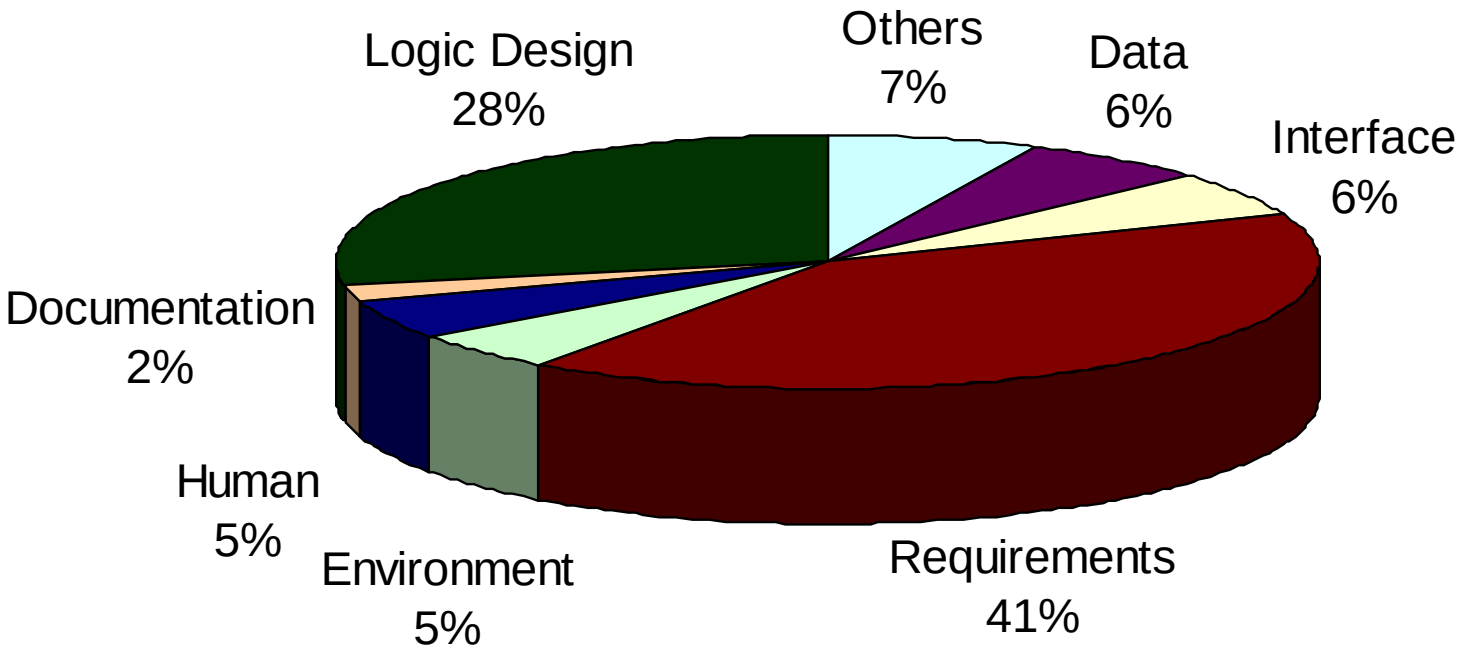
¿De dónde vienen los defectos?



"Los ingenieros de software inyectan en la fase de construcción un promedio de 100 d/KLDC (1 por cada 10 LDC)"

Fuente: SEI (30,000 programas escritos por ingenieros de todo el mundo) .

Fuente de los Errores



*From a U.S. Air Force Project, F. Sheldon, 1992
Reliability Measurement from Theory to Practice

Costo relativo para corregir un error

1	Requirements
3-6	Design
10	Coding
15-40	Development Testing
30-70	Acceptance Testing
50-100	Operation

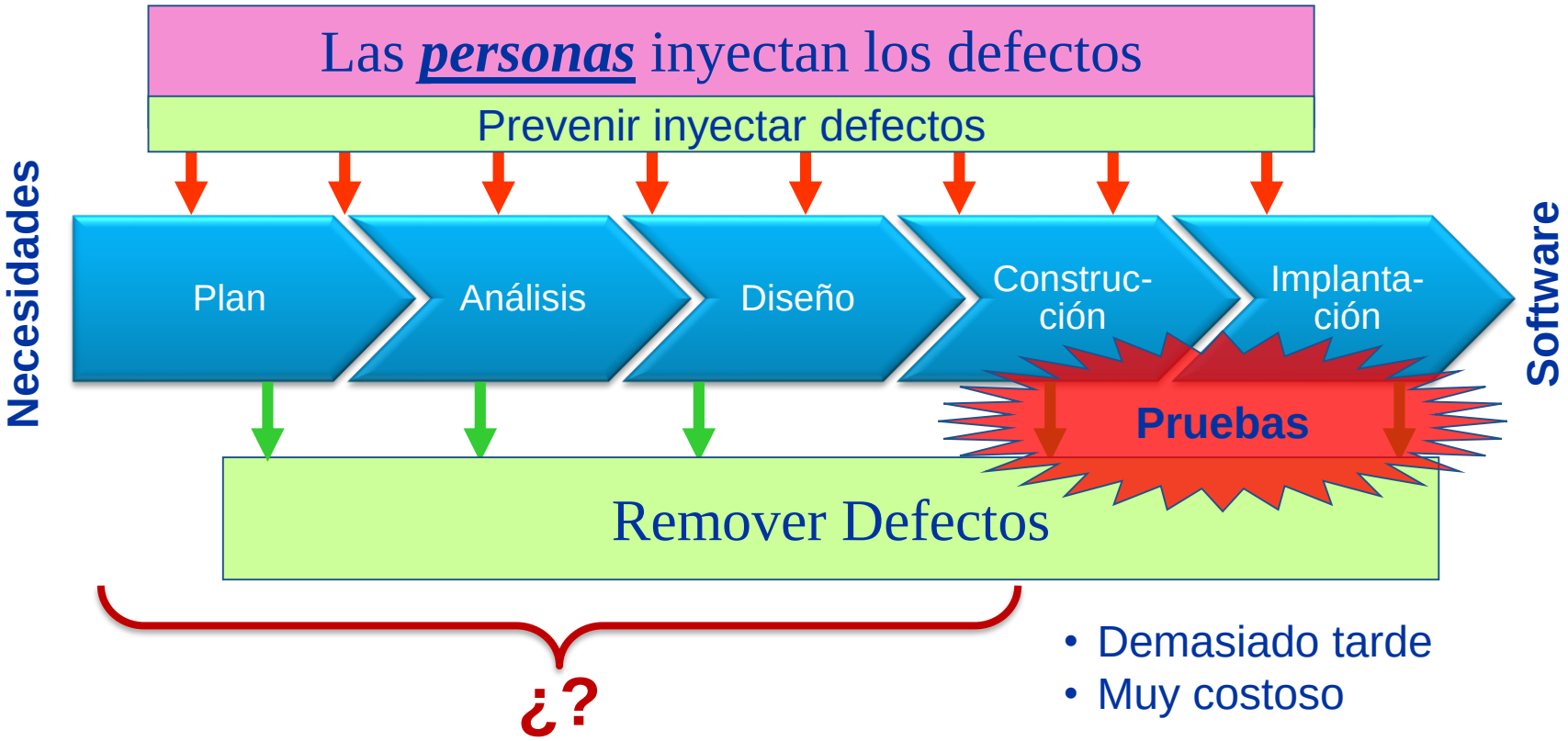
*From a U.S. Air Force Project, F. Sheldon, 1992
Reliability Measurement from Theory to Practice

Tecnológico de Monterrey

La gran pregunta es…

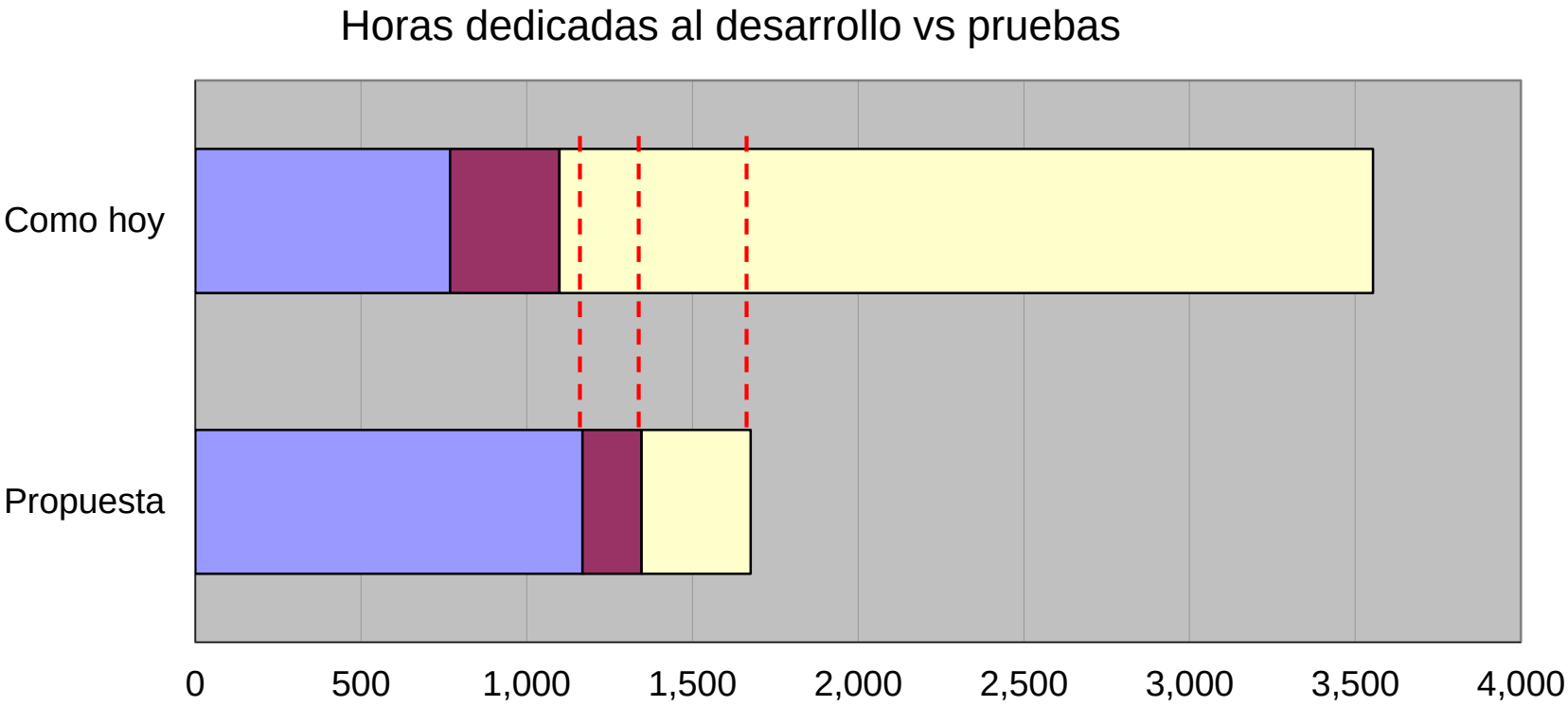
¿Qué podemos hacer para que al cliente le entreguemos muy pocos defectos (casi cero)?

Necesitamos remover y prevenir los defectos



- Demasiado tarde
- Muy costoso

Economía de la Calidad



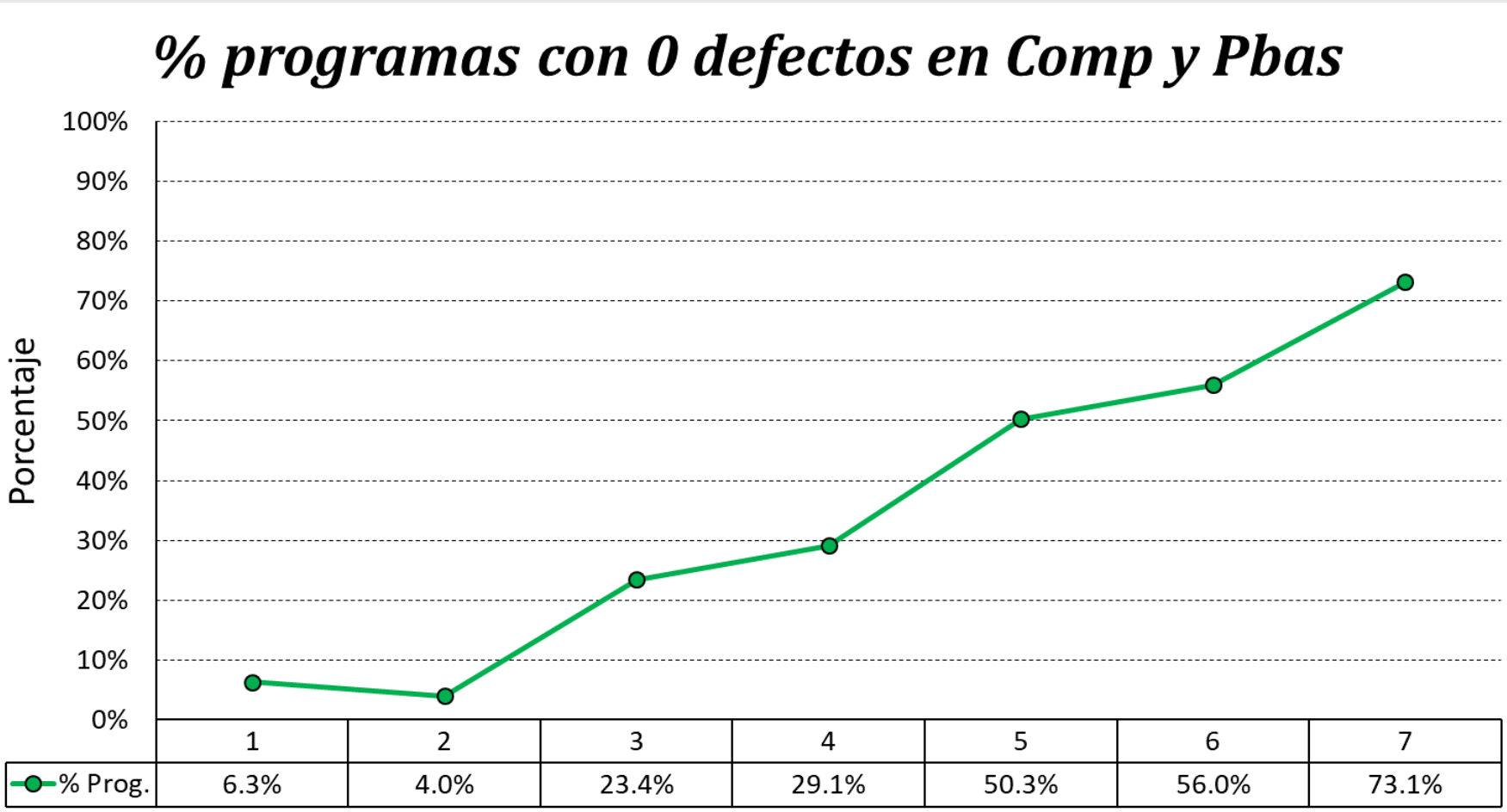
	Propuesta	Como hoy
Desarrollo	1168.6	768.6
Pruebas Unit.	178.3	330.4
Pruebas Finales	329.3	2454.9

Reflexión

- ¿Cuántos programas has desarrollado donde:
  - Al compilar tengas cero defectos y
  - Al probar tengas también cero defectos?

- ¿Es posible hacerlo siempre o casi siempre?

Resultados de 175 alumnos del Tec



Conclusión

- La industria del software tiene problemas entregando software de calidad
  - Inclusive se piensa que es imposible hacerlo
- Sus únicos esfuerzos se enfocan a probar bien el producto
  - Pero eso no es hacer las cosas “con calidad”
  - Cuando pruebas un producto, este ya tiene “construida la calidad”
- Si queremos hacer software de calidad debemos construirlo con calidad
  - O sea, deben llegar muy pocos defectos a las fases de pruebas
- Existen técnicas comprobadas para hacerlo