



Tecnológico
de Monterrey

Seguimiento/monitoreo de la calidad

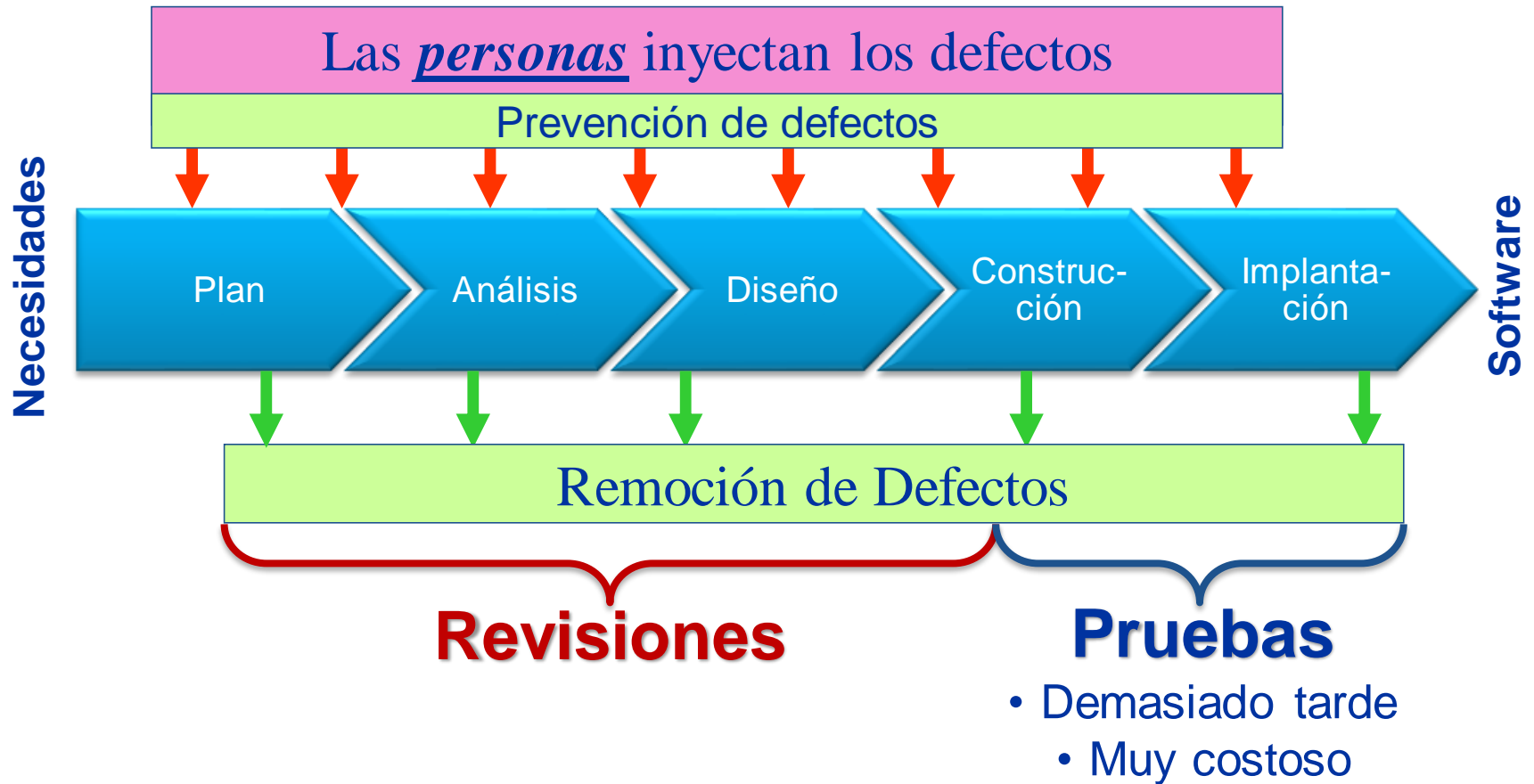
Módulo 3: Modelos de calidad
para equipos y personas



Administración de la Calidad

- Objetivo:
 - Entregar cero defectos al cliente
- Para entregar cero defectos:
 - Administramos la cantidad de defectos
 - Eso significa:
 - Estimar defectos a inyectar (basado en datos históricos)
 - Hacer plan de cómo y cuando removerlos
 - Darle seguimiento al plan de calidad
 - Realizar los ajustes necesarios para cumplir

Calidad = prevenir y remover defectos





Principales tipos de revisión del software

Nombre	Cuantos revisan	Descripción	Objetivo
Revisión	1 persona	Análisis atento y cuidadoso individual <ul style="list-style-type: none">• Rev. personal = realizada por el autor• Rev. por colega (peer review) = realizada por alguien distinto al autor	Encontrar defectos
Recorrido (Walkthrough)	2 o más personas	Presentación guiada a un grupo	Múltiples
Inspección	2 o más personas	Revisión sistemática por un grupo entrenado	Encontrar defectos

Ver estándar IEEE 1028 “Standard for Software Reviews and Audits”

Queremos utilizar los métodos más efectivos

	Efficiency	Effectiveness	Predictability
Activity	Average removal rate (defects/hr)	Phase yields (% of defects removed)	Estimated effort
Design Review	1.5	50% to 70%	Based on product size
Design Inspection	0.5	70% to 90%	
Code Review	4	50% to 70%	
Code Inspection	1	70% to 90%	
Unit Test	0.2	35% to 50%	Based on time to find a solution to a problem
Integration Test	0.1	35% to 50%	
System Test	0.05	35% to 50%	

Seguimiento al Plan de Calidad

Contamos con un plan de calidad

Phase	Defects Injected Per Hour	Phase Yields	Defects Injected	Defects Removed
Planning				
Requirements	0.55		52.48	
System Test Plan				
REQ Inspection		85%		44.60
High-Level Design	0.70		100.80	
Integration Test Plan	0.50		43.44	
HLD Inspection		50%		76.05
Detailed Design	0.91		91.38	
DLD Review		50%		83.72
Test Development				
DLD Inspection		60%		50.23
Code	2.00		200.84	
Code Review		70%		164.03
Compile	0.05	50%	0.48	35.39
Code Inspection		60%		21.23
Unit Test	0.07	50%	2.57	8.36
Build and Integration Test	0.04	60%	15.73	14.46
System Test	0.04	60%	5.18	8.89



Junta semanal de avance

- El equipo da seguimiento a los planes en la junta semanal de avance
- Extracto del Script WEEK:

Paso	Actividad	Descripción
1	Roles de la Junta	-----
2	Agenda de la Junta	-----
3	Reporte de Gerencia	-----
4	Reporte de los Roles	Los miembros del equipo revisan las responsabilidades de sus roles asignados y el estado de cada uno contra el rol (especificación ROLE).

Ejemplo de junta semanal

- El "Quality Manager" informa que en la inspección de un componente no se encontraron los defectos que se esperaban
- Estos son los datos de la inspección del componente:

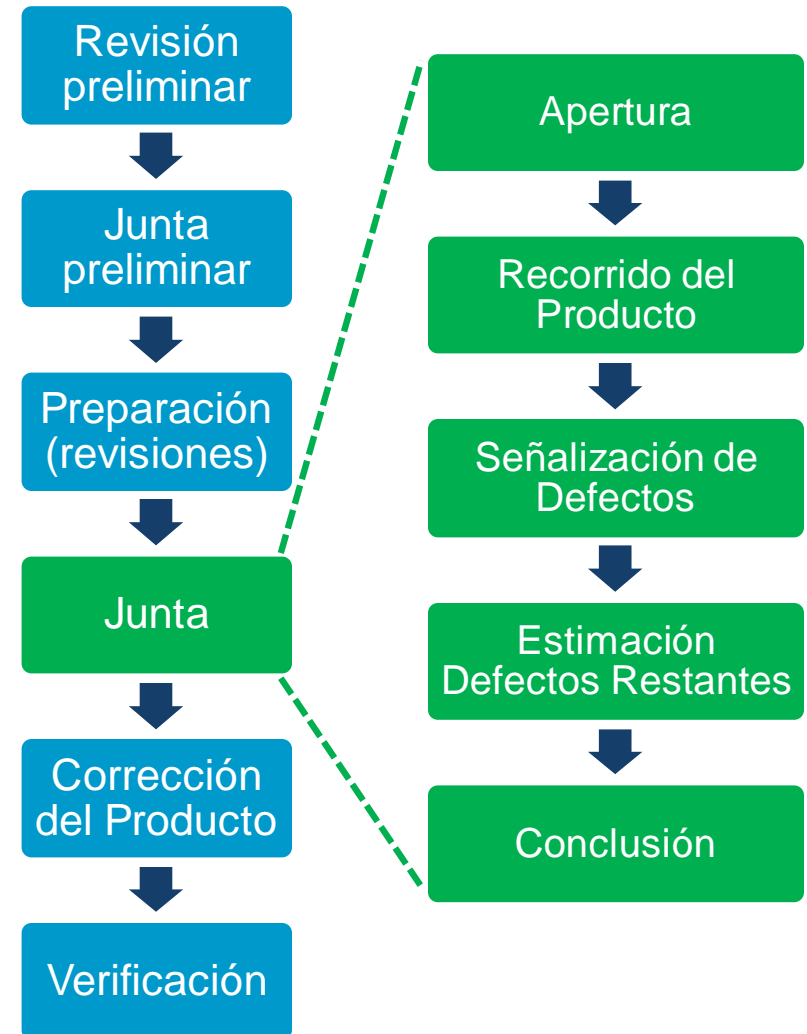
	<u>Plan</u>	<u>Real</u>
Tiempo de construcción.....	12.4 hr.....	15.2 hr
Defectos en inspección.....	13.....	12
Yield de la inspección.....	80%.....	¿?

- ¿Cuál fue el Yield real?
 - Originalmente se planeaban inyectar 16.25 defectos ($13 / 0.8$)
 - Regla de 3:
 - Si en 12.4 hr \rightarrow 16.25 def... en 15.2 hr \rightarrow ¿Cuántos defectos?
 - $15.2 * 16.25 / 12.4 = 19.92$ defectos
 - Muy probablemente el Yield real sea de solo un **60.2%** ($12 / 19.92$)
- El equipo discute como solucionarlo ($60.2\% \ll 80\%$)

Inspecciones de Software

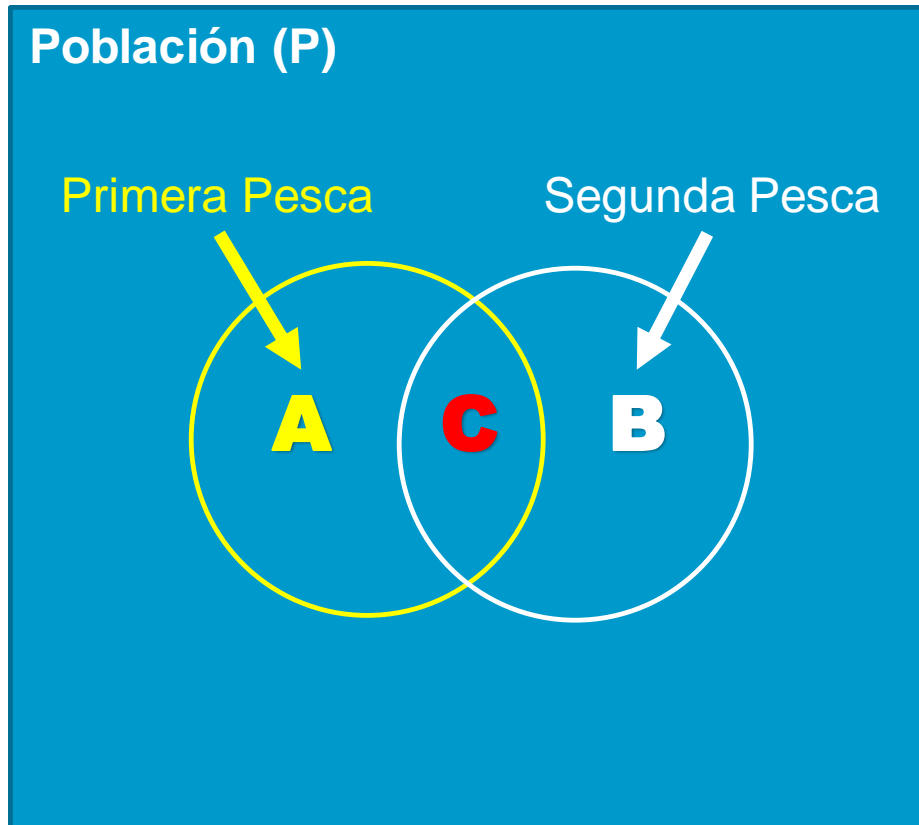
Inspecciones en el TSP

- Una **inspección** es la revisión de un producto de trabajo por 2 o más personas capacitadas que buscan defectos mediante un proceso bien definido
- En el TSP las inspecciones son formales
 - Roles definidos
 - Scripts, formas y recolección de métricas
 - Checklists para las revisiones individuales
 - Medición estadística de la efectividad



Las inspecciones permiten estimar los defectos escapados

- Método estadístico de “la pesca”



Regla de 3:

$$P \rightarrow A$$

$$B \rightarrow C$$

$$P = (A * B) / C$$

Ejemplo

Supongamos una inspección con 4 inspectores donde encuentran los siguientes defectos:

Defecto	Ins1	Ins2	Ins3	Ins4	TOTAL
Defecto 1	1	1		1	3
Defecto 2		1		1	2
Defecto 3		1	1		2
Defecto 4	1		1		2
Defecto 5			1	1	2
Defecto 6		1			1
Defecto 7				1	1
Defecto 8			1		1
Defecto 9	1			1	2
Defecto 10				1	1
Defecto 11		1	1		2
Defecto 12			1		1
Defecto 13				1	1
Defecto 14				1	1
TOTAL	3	5	6	8	22



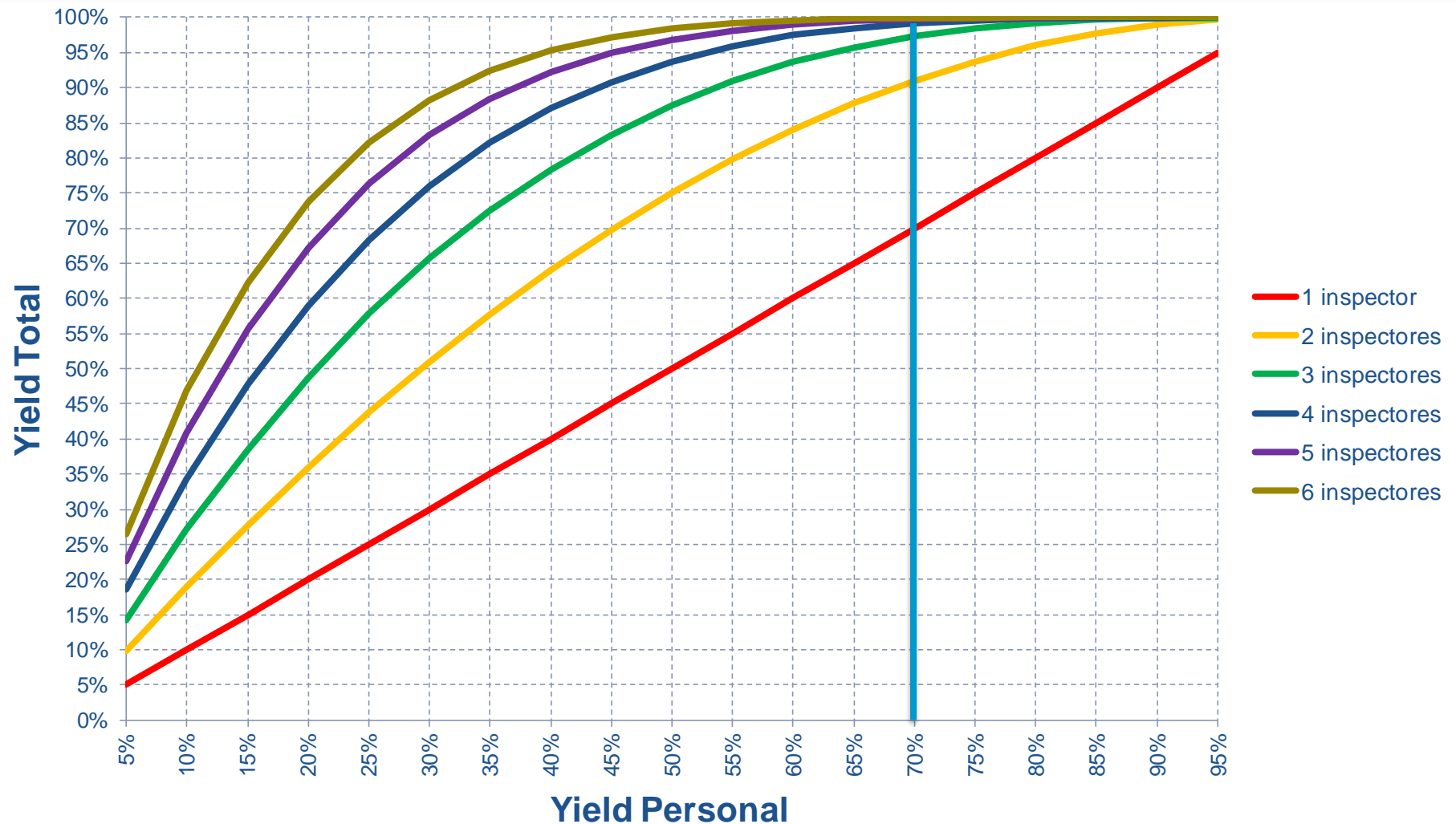
Ejemplo

Pasos:

1. Contar defectos únicos por inspector (en color verde)
2. $A \leftarrow$ inspector con más defectos únicos (inspector 4)
3. $B \leftarrow$ unión de los otros inspectores (insp. 1 a 3)
4. $C \leftarrow$ intersección de A y B
5. Defectos presentes = $A * B / C$
($8 * 10 / 4 = 20$)
6. Yield = def. encontrados / def. presentes ($14 / 20 = 70\%$)

Defecto	Ins1	Ins2	Ins3	Ins4	TOTAL	A	B	C
Defecto 1	1	1		1	3	1	1	1
Defecto 2		1		1	2	1	1	1
Defecto 3		1	1		2		1	
Defecto 4	1		1		2		1	
Defecto 5			1	1	2	1	1	1
Defecto 6		1			1		1	
Defecto 7				1	1	1		
Defecto 8			1		1		1	
Defecto 9	1			1	2	1	1	1
Defecto 10				1	1	1		
Defecto 11		1	1		2		1	
Defecto 12			1		1		1	
Defecto 13				1	1	1		
Defecto 14				1	1	1		
TOTAL	3	5	6	8	22	8	10	4
Únicos	0	1	2	4				
Yield Indiv.	15%	25%	30%	40%				

Yield de las inspecciones



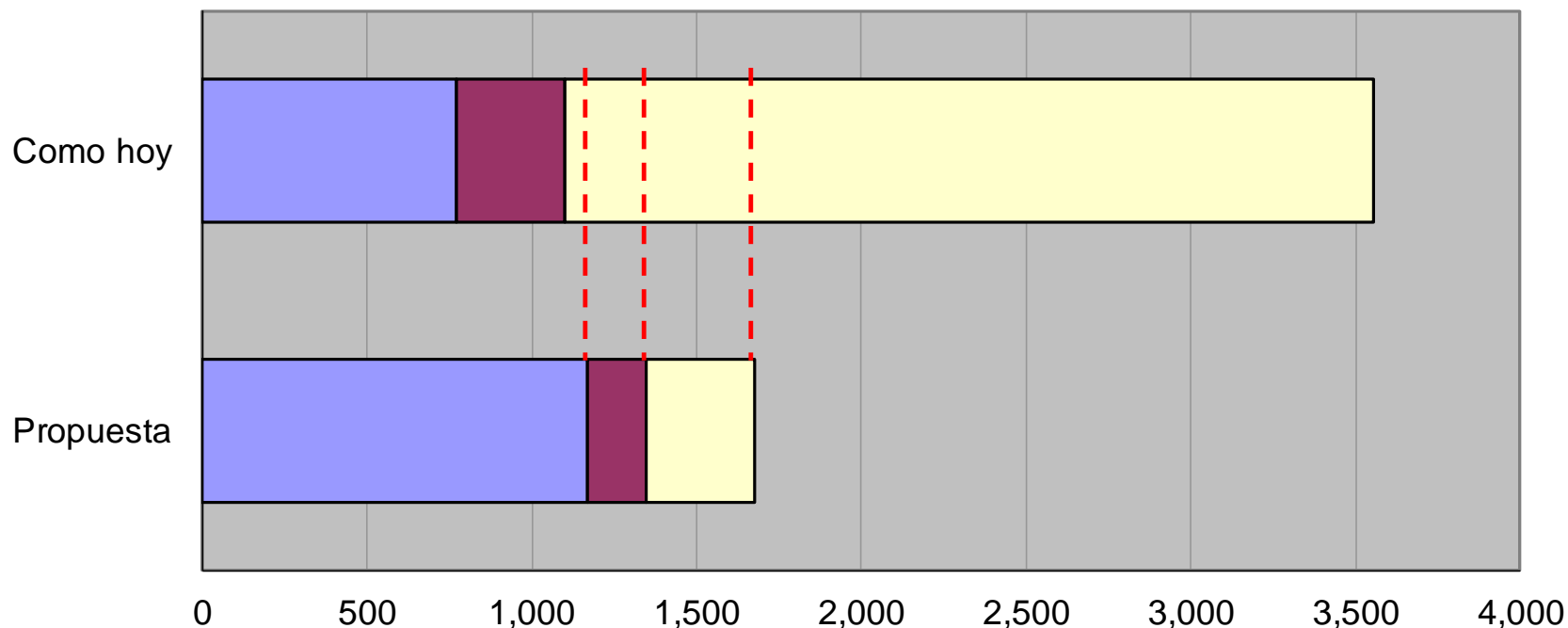


Ejercicio en parejas: ¿Las inspecciones son burocráticas?

- Supongan un software al que
 - No se le realizan inspecciones durante su desarrollo
 - Se le encuentran 150 defectos en las pruebas de sistema
- Supongan además que
 - Arreglar un requerimiento defectuoso en la fase de análisis toma 15 minutos
 - Realizar una inspección del documento de requerimientos toma 80 HP (5 personas, 2 días)
- **¿Cuántas HP se hubieran ahorrado si se hubiera inspeccionado el documento de requerimientos?**
- Tomar en cuenta que de acuerdo a estadísticas mundiales
 - 40% de los defectos ocurren en los requerimientos
 - Arreglar un requerimiento defectuoso durante las pruebas de sistema cuesta 40 veces más que arreglarlo en el análisis
 - Una inspección detecta en promedio el 70% de los defectos

Economía de la Calidad

Horas dedicadas al desarrollo vs pruebas



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