



Modelos de Calidad para el Software

Preparación para CENEVAL



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



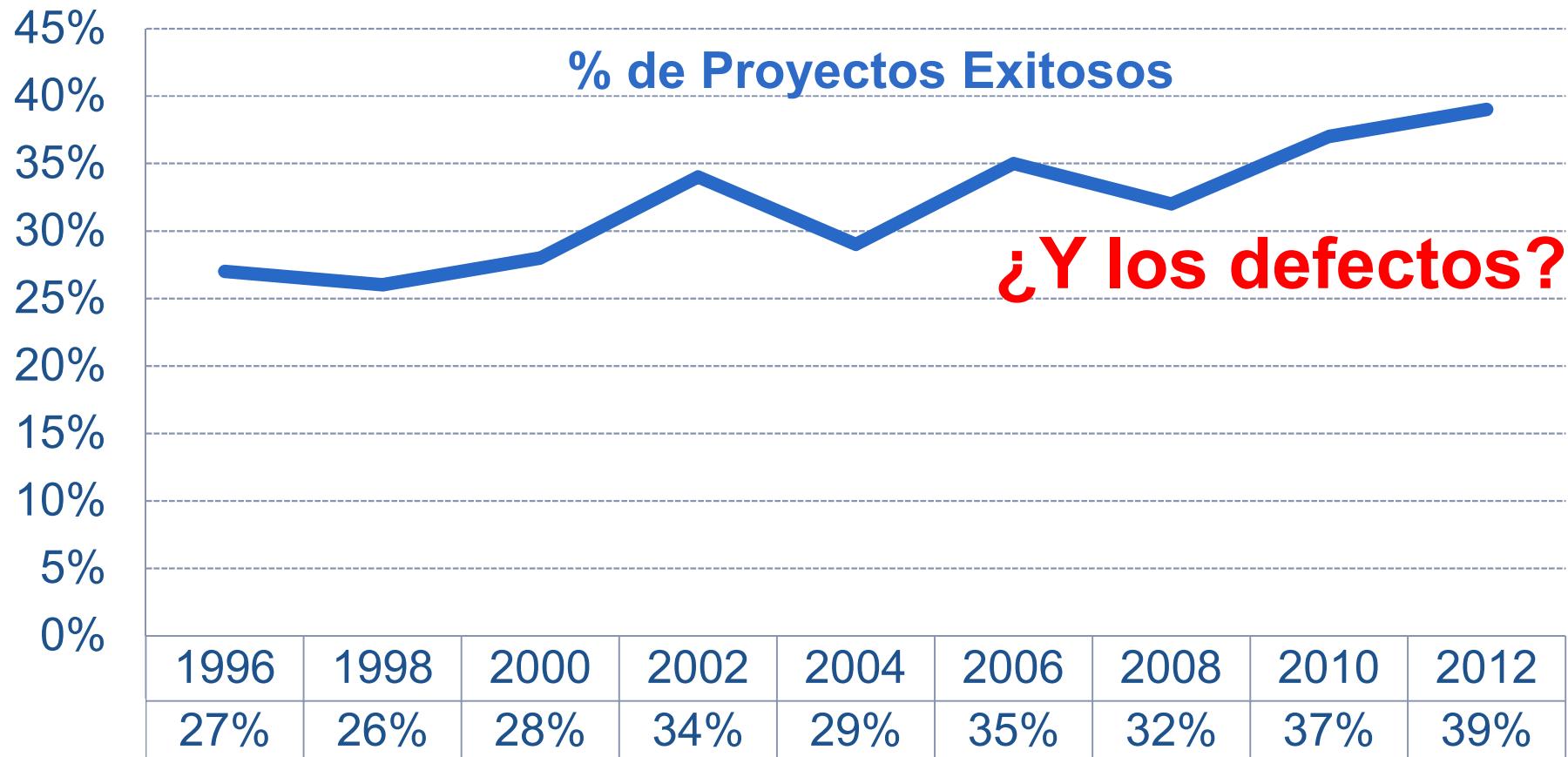
Varias décadas después...

- **La industria mundial del desarrollo del software continúa en problemas.**
- **Por ejemplo, en el 2004:**
 - 24% de los proyectos se cancelaron por problemas
 - Al mundo le costó \$81,000 millones USD
 - 44% de los proyectos se entregaron tarde o con menos funcionalidad
 - Costaron 189% del estimado inicial
 - Este exceso nos costó otros \$59,000 millones USD
 - Sólo 32% de los proyectos fueron exitosos (a tiempo y con funcionalidad prometida)

Fuente: Chaos Report 2005, Standish Group



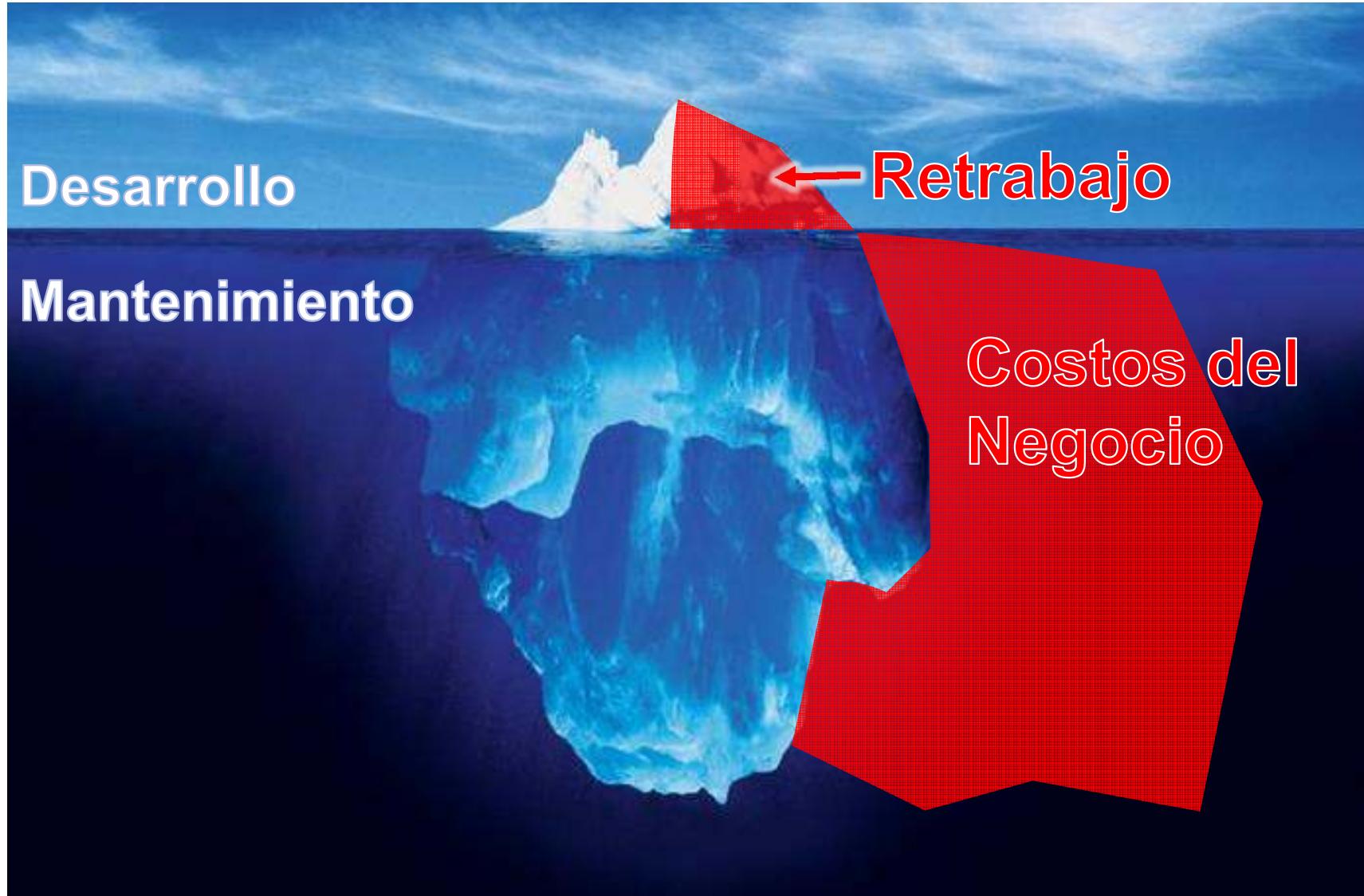
No existe una mejora sustantiva



Fuente: Chaos Manifesto 2013, Standish Group



¿Y los costos escondidos?





¿Y los costos escondidos?





Algo debemos estar haciendo mal

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



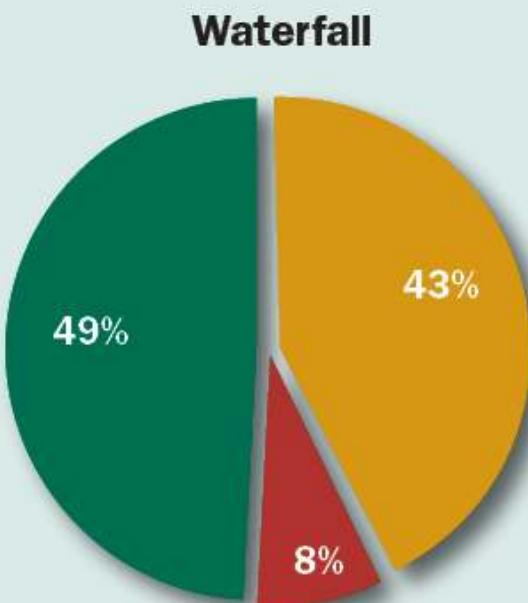
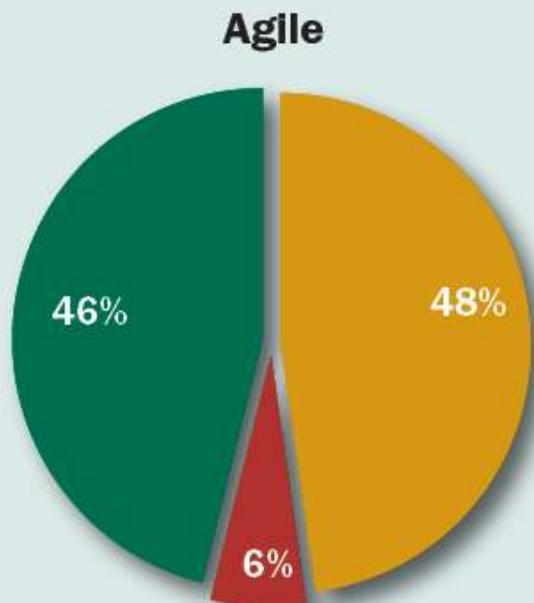
Hablemos primero de datos...





¿Qué metodología es mejor: ágil o cascada?

AGILE V. WATERFALL SMALL PROJECTS



- Successful
- Failed
- Challenged

The charts show success rates for small software development projects using modern languages, methods, and tools, from 2003 to 2012.

Fuente: Chaos Manifesto 2013, Standish Group



¿Qué tan efectivo es SCRUM?

Métrica	Unidad	SCRUM	TSP	% Dif.
Tamaño	Puntos Función	1,000	1,000	-
Tamaño	LDC de C++	75,000	75,000	-
Esfuerzo	Meses Hombre	84	86	2%
Defectos Entregados	Cantidad	370	87	-76%
Def. Críticos	Cantidad	68	16	-76%
Costo Desarrollo	USD	\$633,043	\$644,070	2%
Costo Calidad	USD	\$774,142	\$298,699	-61%
Costo Total	USD	\$1,467,957	\$1,026,660	-30%

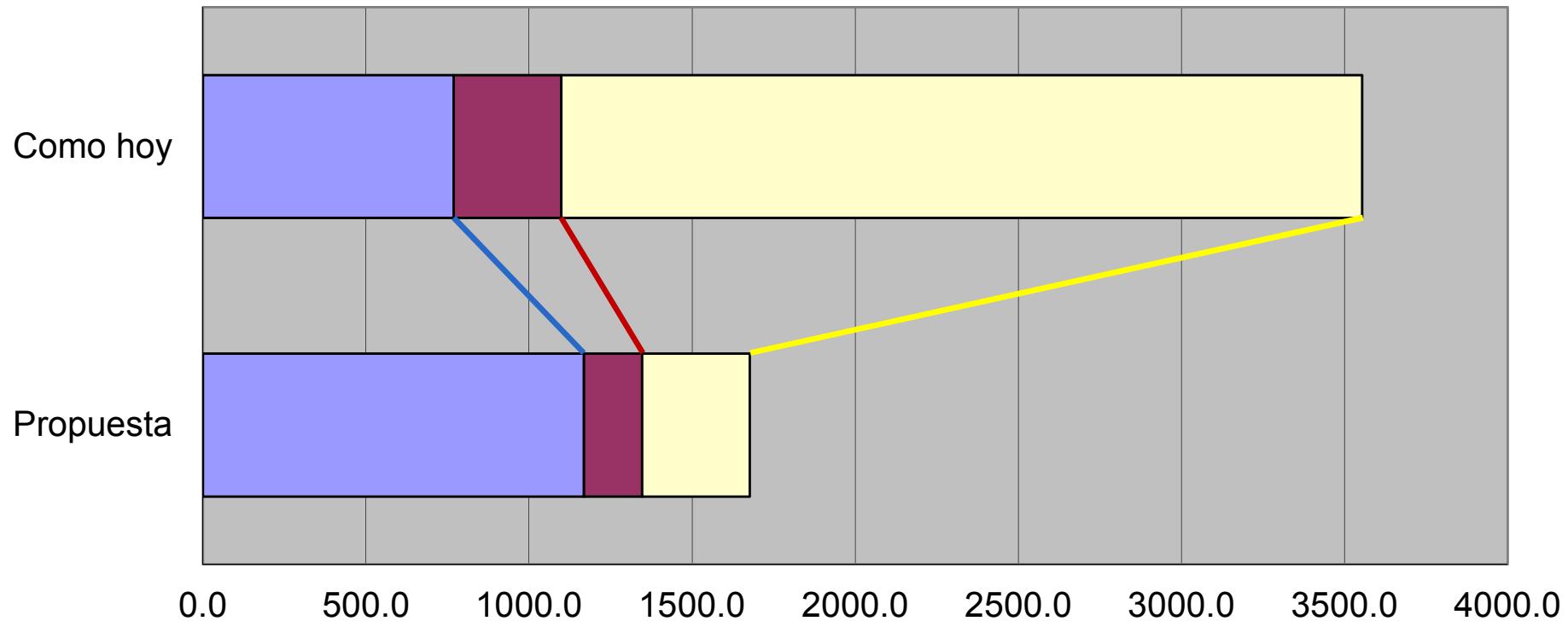
Fuente: Capers Jones, Evaluating Agile and Scrum with Other Software Methodologies, 20-Mar-2013

<http://www.infoq.com/articles/evaluating-agile-software-methodologies>



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

- **¡¡¡CUIDADO!!!**
con las conclusiones sin datos
 - Nuestra “intuición” nos puede traicionar
- **La solución parece que no solo está en cambiar de metodología**
 - La metodología se escoge de acuerdo al tipo de proyecto y al tipo de cliente
- **La solución está en... *¡¡¡la persona!!!***
 - Necesitamos... Calidad → Trabajo Personal



Hablemos ahora de Enfocarse al Proceso



Enfoque a Procesos

- **Es mejor y más barato construir el producto “bien desde el principio”**
 - Revisarlo al final es costoso, impredecible y defectuoso
- **Para lograrlo**
 - Los procesos de producción tienen que ser “procesos maduros”
 - Esto significa:



- **Para hacerlo con alta calidad**
 - Se requiere “disciplina de procesos”



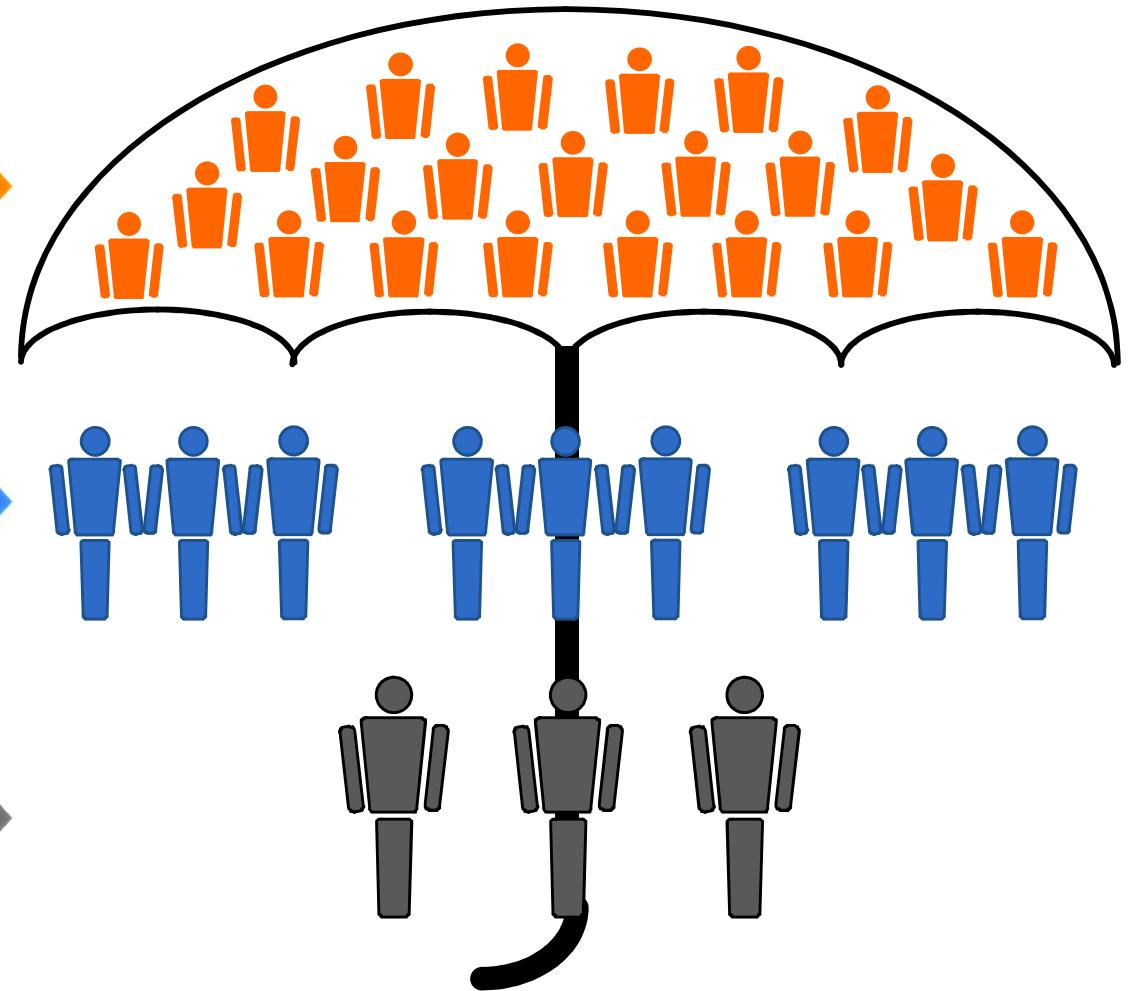
Clave del éxito: disciplina de procesos en todos los niveles

Disciplina de procesos en:

Organización

Equipos

Individuos





Clave del éxito: disciplina de procesos en todos los niveles

Disciplina de procesos en:

CMMI

ISO-15504

Moprossoft

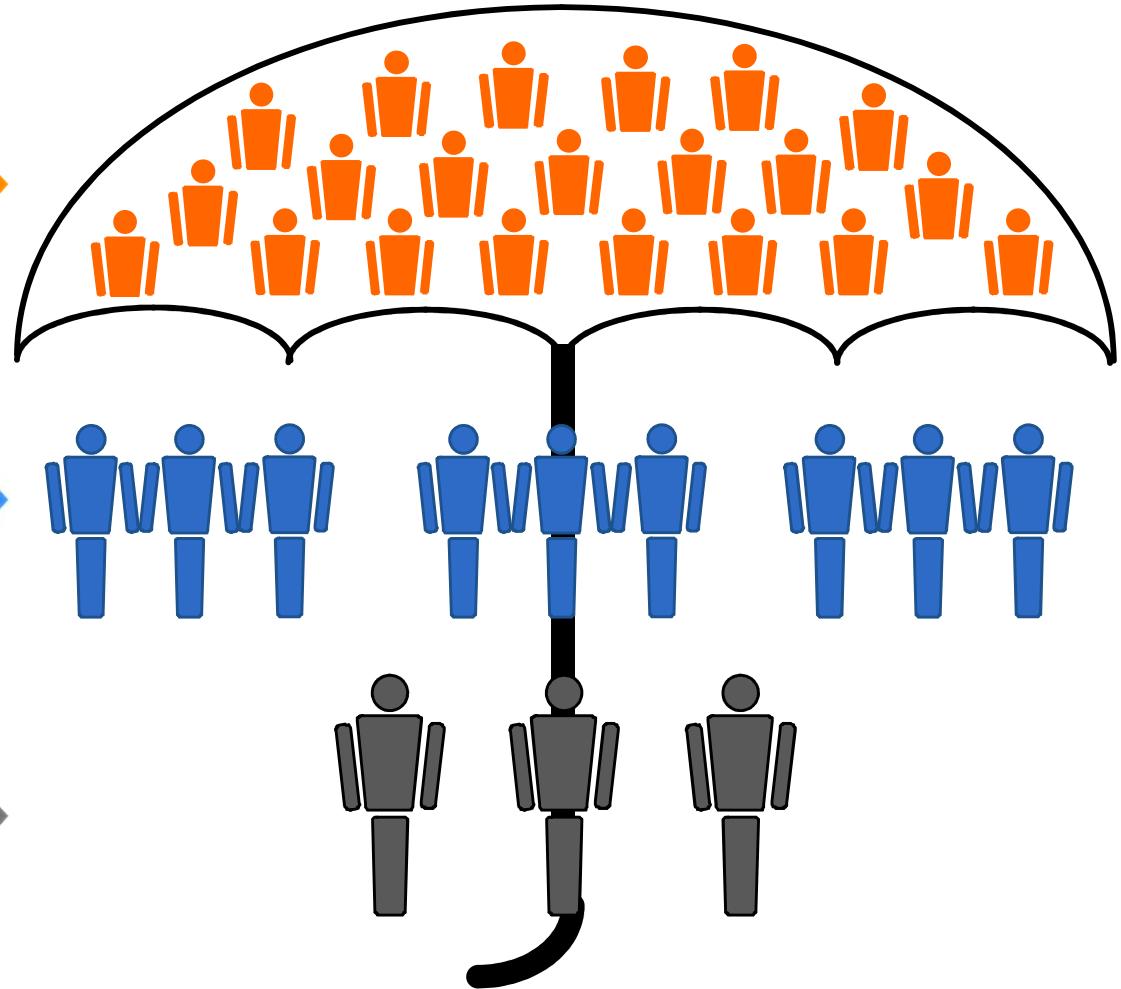
Organización

TSP

Equipos

PSP

Individuos



CMMI es una marca registrada por Carnegie Mellon University y Team Software Process, TSP, Personal Software Process y PSP son marcas de servicio registradas por Carnegie Mellon University



Clave del éxito: disciplina de procesos en todos los niveles

Disciplina de procesos en:

CMMI
ISO-15504
Moprossoft

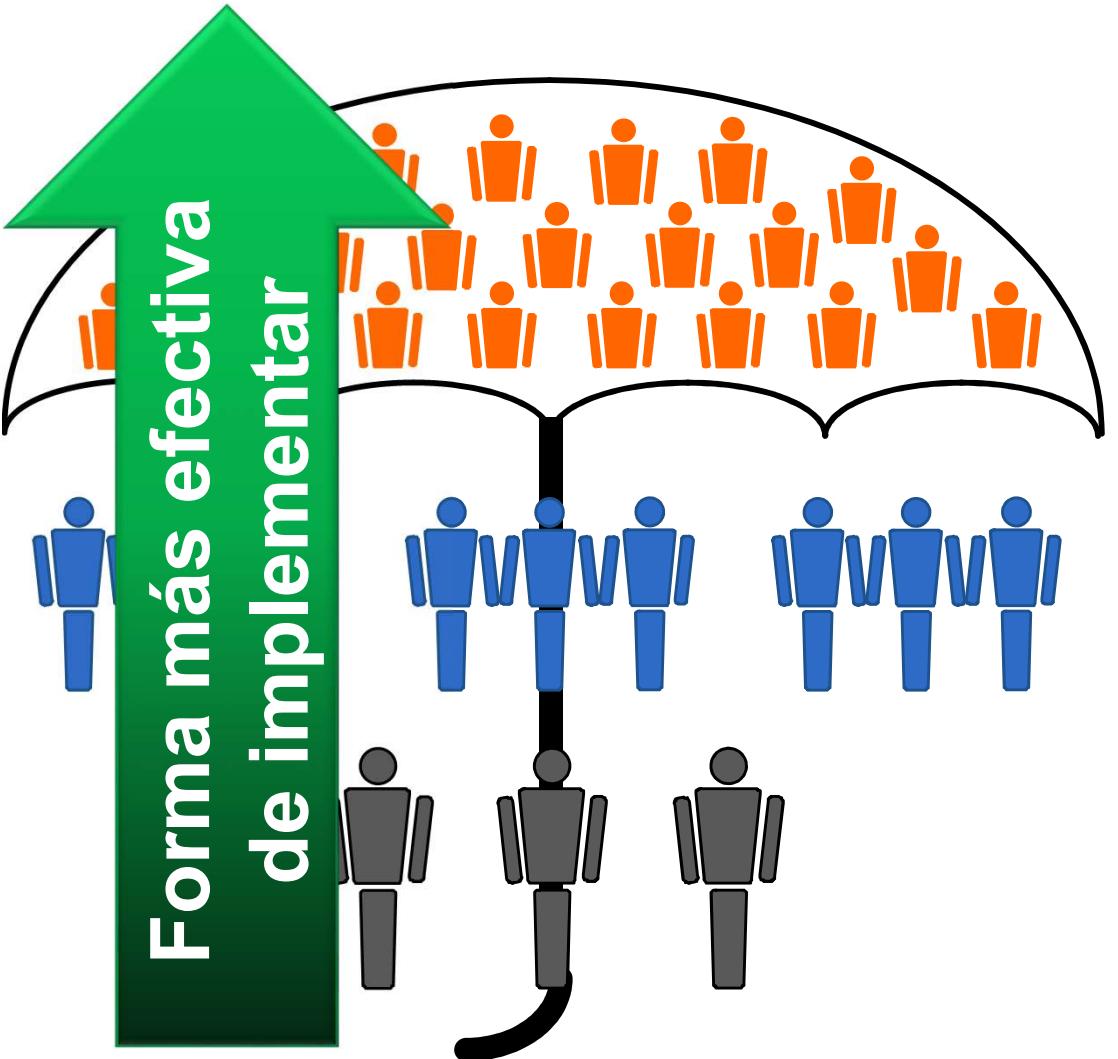
Organización

TSP

Equipos

PSP

Individuos



CMMI es una marca registrada por Carnegie Mellon University y Team Software Process, TSP, Personal Software Process y PSP son marcas de servicio registradas por Carnegie Mellon University



Modelos Operacionales: TSP y PSP



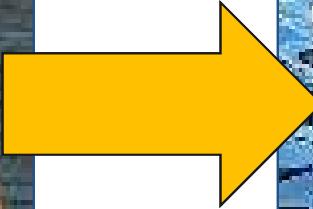
El SW se construye por personas en equipo

- Por eso necesitamos personas y equipos de alto desempeño





Paso #1 → la persona





Personal Software Process



Personal Software Process

SEI Certified PSP Developer
=

Desarrollador certificado en nivel 5 de CMMI

- Utiliza un proceso personal definido y medido
- Recolecta datos de tamaño, tiempo y defectos
- Utiliza estos datos para:
 - Planear y dar seguimiento a su trabajo personal
 - Administrar la calidad de los productos que produce
 - Mejorar cuantitativamente su propio rendimiento



Los elementos del PSP

Phase:	Purpose:	To guide you in developing module-level programs
	Inputs Required	Problem description PSP project plan summary form Time and defect recording logs Defect type log Stop watch (optional)
1	Planning	<ul style="list-style-type: none">• Produce or obtain a requirements statement.• Estimate the required development time.• Estimate the required project plan summary form.• Complete the time log.
2	Development	<ul style="list-style-type: none">• Design the program.• Implement the design.• Complete the code and fix and log all defects found.• Test the programs and fix and log all defects found.• Complete the time recording log.
3	Postmortem	<ul style="list-style-type: none">• Complete the project plan summary form with actual data.• Complete the time log.
	Exit Criteria	<ul style="list-style-type: none">• A thoroughly tested program.• Completed project plan summary with estimated and actual data.• Completed defect and time logs.

Scripts

Documentan los criterios de entrada, las fases/pasos y los criterios de salida de cada proceso. Su propósito es guiar en el uso del proceso.



Métricas

Miden el proceso y el producto. Proveen visión sobre como funciona el proceso y sobre el estado actual del trabajo.

Student Program Identifier	Program #	Date	Program #	Date	Language
Summary					
LOC/Bloc					
Actual Time					
Planned Time					
(Time Difference Index)					
% Done					
% New Code					
Total Defects/LLOC					
Yield %					
% Approval COQ					
% Defects COQ					
COQ/A/F Ratio					
Program Size (LOC's)					
Deleted (D)					
Modified (M)					
Added (A)					
Reused (R)					
Total Lines of Code (LOC)					
Total LOC's (%)					
Total Lines of Code (LOC)					
Total New Reuse					
Total Object LOC (OLC)					
Upper Production Interval (70%)					
Lower Production Interval (70%)					
Time in Phase (min.)					
Planning					
Design					
Design review					
Code review					
Compile					
Test					
Postmortem					
Total Time (IP) (70%)					
Total Time (AP) (70%)					

Formatos

Proveen una forma conveniente y consistente de capturar y retener los datos medidos.



Estándares

Proveen definiciones consistentes que guían el trabajo y la recolección de datos.



Ejemplo de un script

Propósito		Guia el proceso de desarrollo de un programa computacional, utilizando el "Process Dashboard" como herramienta de captura de métricas personales.
Criterios de Entrada		-Descripción del Problema - "Process Dashboard" con las fases del PSP0 asignadas al programa a desarrollar
#	Paso	Descripción
1	Planeación <i>(Planning)</i>	-Iniciar el reloj (pausarlo cada vez que haya una interrupción) -Obtener la descripción de los requerimientos -Estudiar y entender los requerimientos -Estimar el tiempo que tomará el desarrollo del programa y capturarlo en la forma "Resumen del Plan" <i>(Project Plan Summary)</i> -Detener el reloj y marcar la fase como terminada
2	Diseño <i>(Design)</i>	-Iniciar el reloj (pausarlo cada vez que haya una interrupción) -Utilizando como base los requerimientos: <ul style="list-style-type: none">▪ Diseñar en forma completa el programa▪ Diseñar todos los casos de pruebas unitarias -Cada vez que se corrige un defecto, registrarlo en la "bitácora de defectos" <i>(Defect Log)</i> -Detener el reloj y marcar la fase como terminada
3	Codificación <i>(Code)</i>	-Iniciar el reloj (pausarlo cada vez que haya una interrupción) -Escribir el código de acuerdo al diseño realizado -Cada vez que se corrige un defecto, registrarlo en la "bitácora de defectos" <i>(Defect Log)</i> -Detener el reloj y marcar la fase como terminada
4	Compilación <i>(Compile)</i>	-Iniciar el reloj (pausarlo cada vez que haya una interrupción) -Compilar el código escrito (repetir hasta que el compilador no marque errores) -Cada vez que se corrige un defecto, registrarlo en la "bitácora de defectos" <i>(Defect Log)</i> -Detener el reloj y marcar la fase como terminada
5	Pruebas Unitarias <i>(Test)</i>	-Iniciar el reloj (pausarlo cada vez que haya una interrupción) -Ejecutar todos los casos de prueba diseñados (repetir hasta que ya no se encuentren errores) -Cada vez que se corrige un defecto, registrarlo en la "bitácora de defectos" <i>(Defect Log)</i> -Detener el reloj y marcar la fase como terminada
6	Postmortem <i>(Postmortem)</i>	-Iniciar el reloj (pausarlo cada vez que haya una interrupción) -Revisar que se hayan registrado en forma completa y correcta todos los tiempos y los defectos -Detener el reloj y marcar la fase y el programa como terminados
Criterios de Salida		-Programa que cumple con los requerimientos y pasa exitosamente todas las pruebas unitarias - "Resumen del Plan" <i>(Project Plan Summary)</i> lleno con los datos estimados y reales - Bitácora de tiempo <i>(Time Log)</i> y de defectos <i>(Defect Log)</i> completas



¿Para que sirven las métricas personales?

- **Cada persona es diferente**

- La variación del rendimiento de los desarrolladores es de 1 a 20 (Sackman, Erikson y Grant; 1968)
- Es un gran error en los planes que hacemos el asumir que todos somos iguales

- **Con datos personales cada persona puede predecir**

- El tamaño de lo que va a producir
- El tiempo que se va a tardar
- La cantidad de defectos que va a injectar



Métricas del PSP y TSP



Tamaño



Esfuerzo (tiempo)



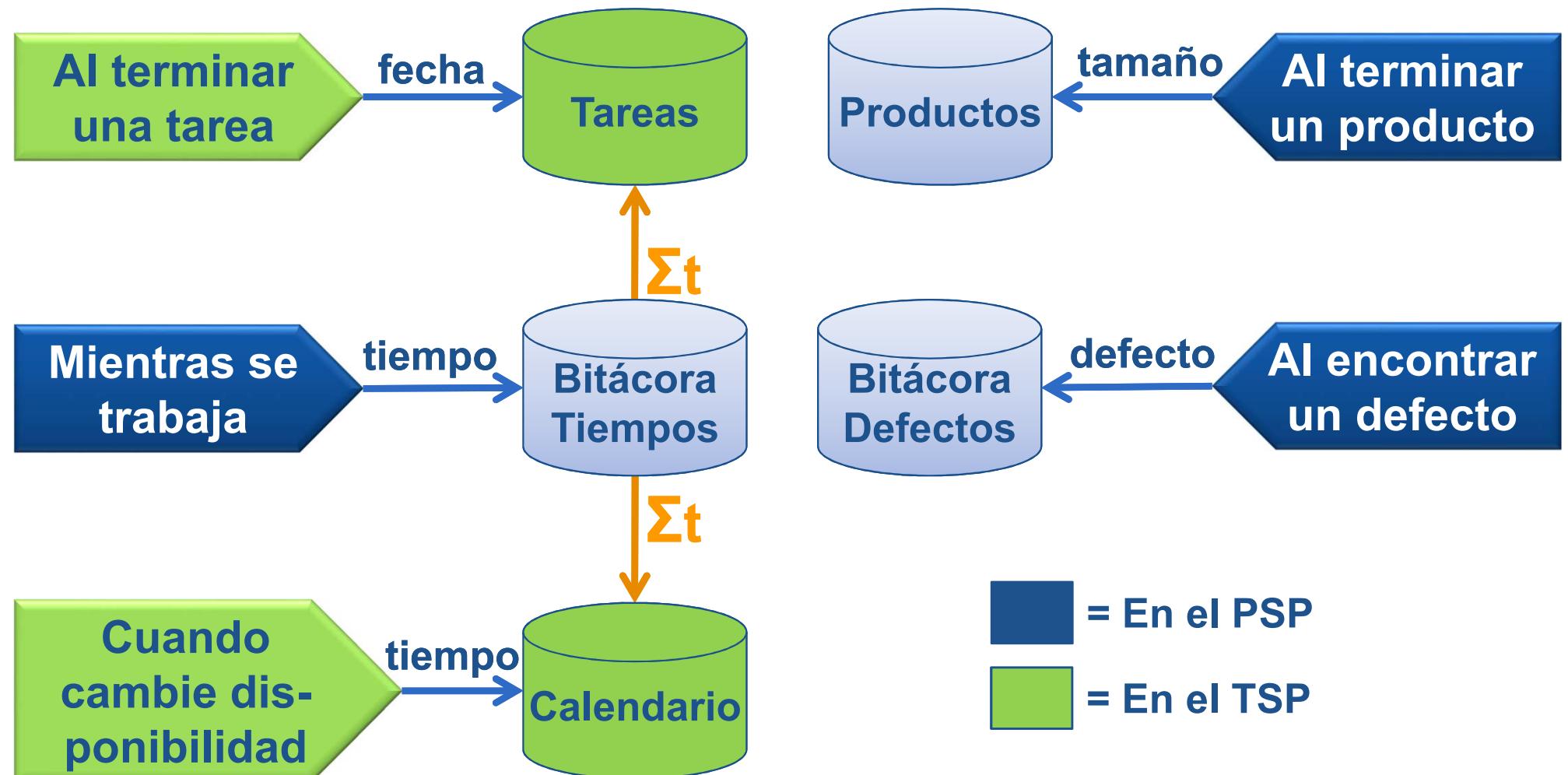
Calidad (defectos)



Calendario



Registro de datos reales





El problema de la estimación

Fecha y Costo



Requerimientos



El problema de la estimación

Requerimientos



**Esfuerzo y
Calendario**

Fecha y Costo



El problema de la estimación

Características de la métrica de tamaño:

1. Se correlaciona con el esfuerzo
2. Es precisa
3. Es fácil de contar

Requerimientos



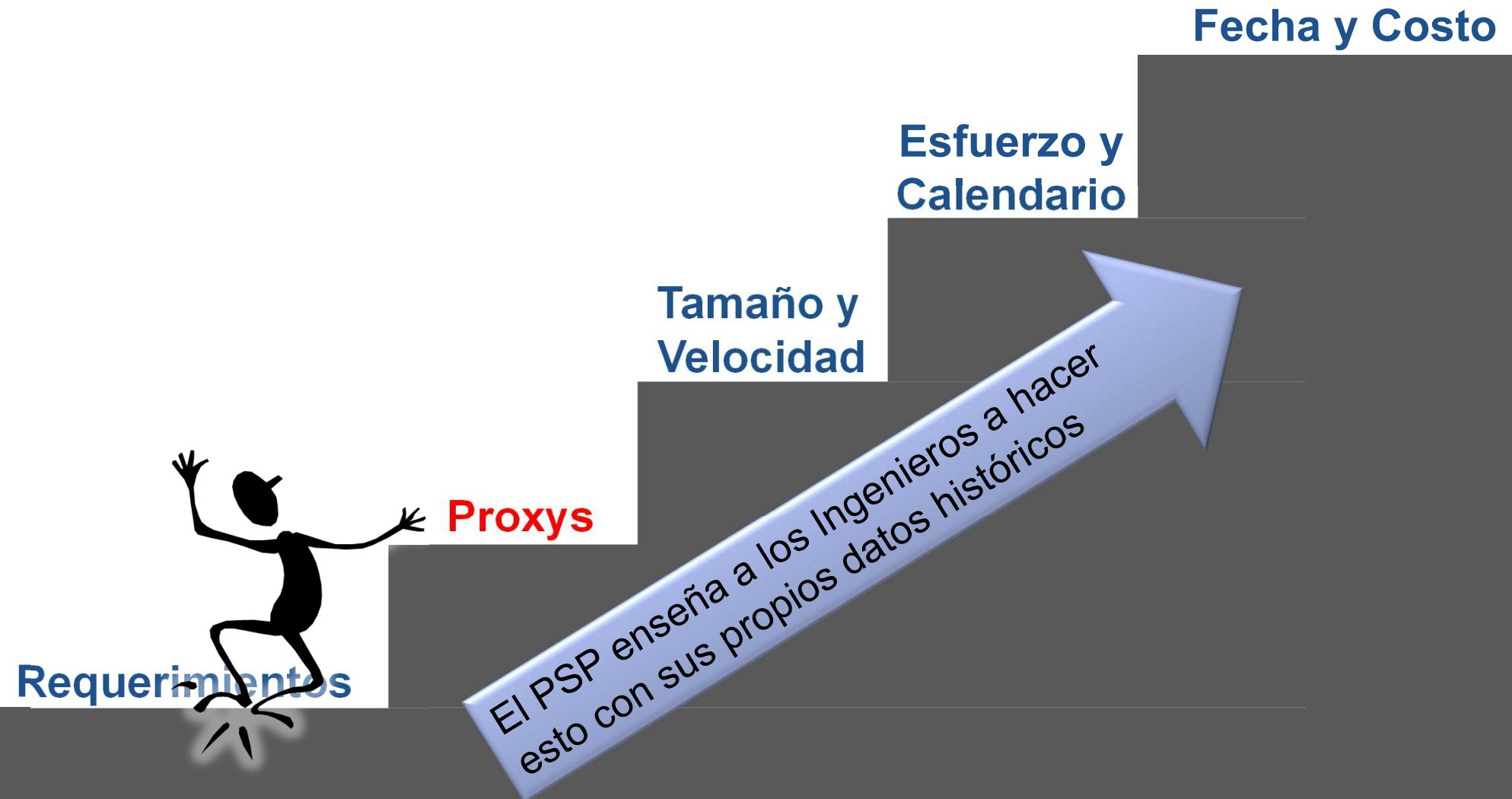
Tamaño y
Velocidad

Esfuerzo y
Calendario

Fecha y Costo



El problema de la estimación





Administración de la Calidad

- **Lo que no se administra:**
 - Muy probablemente no se logra
- **Administrar significa:**
 - Estimar → Planear → Dar seguimiento → Ajustar
 - Sin un buen plan no se puede dar seguimiento
 - Sin seguimiento no se puede cumplir
- **Nos han enseñado como hacerlo con la fecha de entrega:**
 - Para cumplir con la promesa de la fecha de entrega: administramos tareas (o sea, tiempo)
- **Pero... ¿Y la calidad?**



Administración de la Calidad

- **Objetivo:**

- Entregar cero defectos al cliente

- **Para entregar cero defectos:**

- Administraremos la cantidad de defectos
 - Eso significa:
 - Estimar defectos a inyectar (basados en datos históricos)
 - Hacer plan de cómo y cuando removerlos
 - Darle seguimiento al plan de calidad
 - Realizar los ajustes necesarios para cumplir



Hacer solo pruebas no es efectivo

- **Las pruebas encuentran ≈50% de los defectos**
 - Si entran a pruebas muchos defectos → entregamos muchos defectos
- **La mejor forma de entregar cero defectos al cliente**
 - Es encontrar cero defectos en pruebas
- **Por lo tanto → debemos remover todos los defectos antes de pruebas**



Queremos utilizar los métodos más efectivos

	Eficiencia	Eficacia	Predictibilidad
	Taza de Remoción Promedio (defectos / hr.)	Yield de la Fase (% defectos removidos)	Esfuerzo Estimado Depende del tamaño del producto
Revisión Diseño	1.5	50% a 70%	
Inspección Diseño	0.5	50% a 70%	
Revisión Código	4	50% a 70%	
Inspección Código	1	50% a 70%	Depende del tiempo de encontrar la solución
Pruebas Unitarias	0.2	35% a 50%	
Pruebas Integración	0.1	35% a 50%	
Pruebas Sistema	0.05	35% a 50%	



Pero... ¡ El PSP no es suficiente !

- **El software normalmente se construye en equipo**
 - Los Ingenieros de Software no saben trabajar en equipo
 - Muchos gerentes no saben propiciar el trabajo en equipo
- **El PSP por sí sólo **no** funciona**
 - Ya se ha intentado muchas veces





Team Software Process



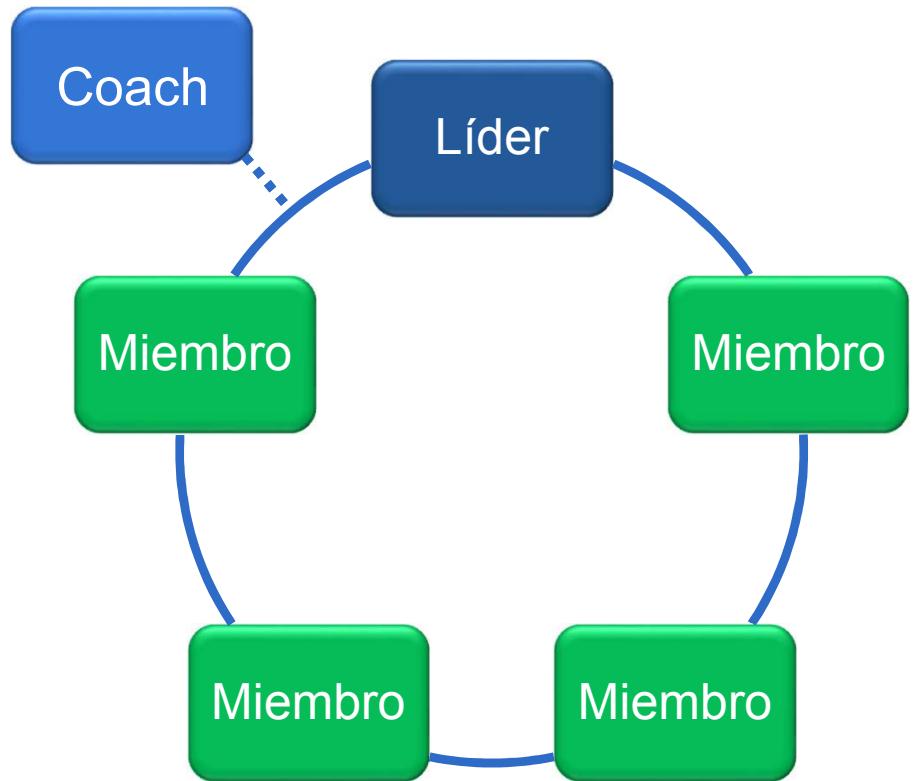
Team Software Process

- **Es una disciplina de procesos para equipos de ingenieros de SW basada en los 5 niveles del CMMI**
 - No es una "metodología" (funciona con cualquier metodología)
- **Se sustenta en los siguientes principios**
 - Los desarrolladores conocen mejor lo que se debe hacer.
 - Cuando los desarrolladores planean su trabajo se comprometen.
 - Para un seguimiento preciso del proyecto se necesitan planes detallados y datos exactos.
 - Solo la gente que hace el trabajo puede recolectar datos precisos y exactos.
 - Para minimizar el tiempo del proyecto el equipo debe balancear su carga de trabajo.
 - Para maximizar su productividad el equipo debe enfocarse primero a la calidad.

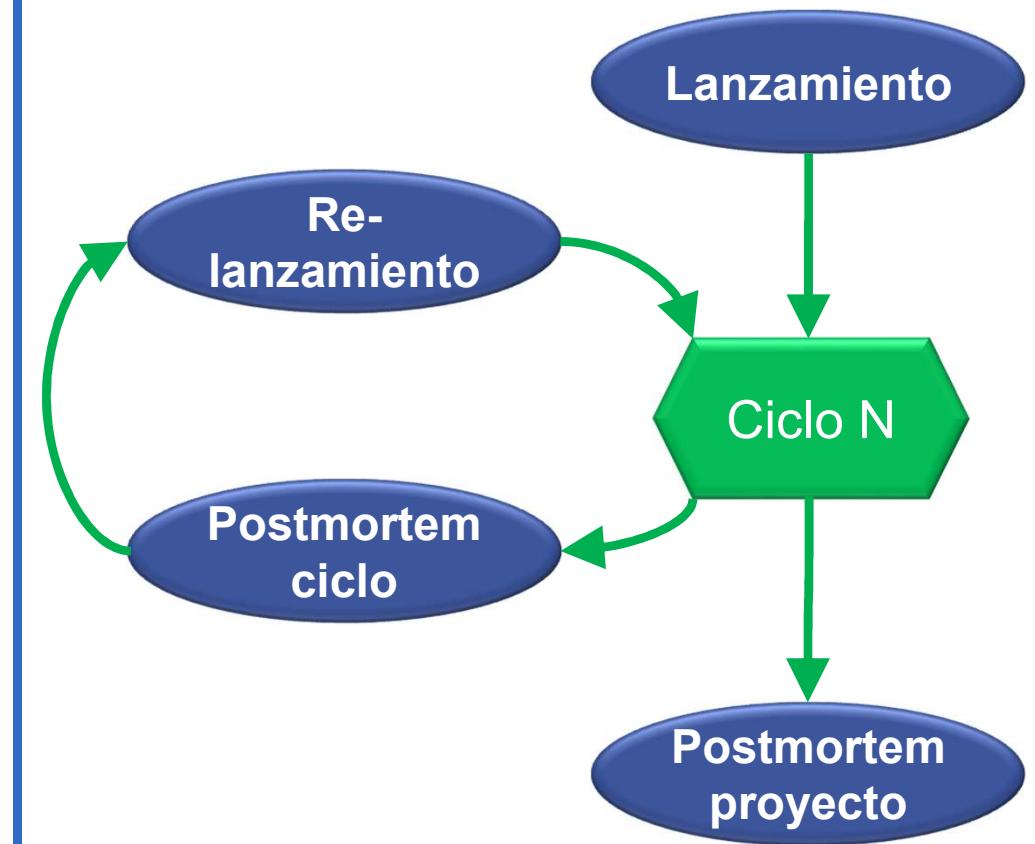


Organización en el TSP

Equipos: auto-administrados
y disciplinados



Proyectos: cíclicos, planeados,
seguidos y en mejora continua





El TSP consta de 2 procesos

Proceso para construir un equipo:

- Durante un “lanzamiento” o “re-lanzamiento”
- El equipo:
 - Define metas y se asigna roles
 - Estima el tamaño del producto que va a desarrollar
 - Define la estrategia, los procesos y los planes detallados para realizar el trabajo
 - Negocia los compromisos finales

Proceso para trabajar en equipo:

- El equipo se auto-administra
 - Sigue los procesos y planes que definió
 - Recolecta datos precisos y exactos de su trabajo
 - Analiza los datos para identificar problemas con anticipación
 - Encuentra soluciones en equipo y las ejecuta
- Siempre se enfoca primero a la calidad y después a la fecha de entrega





Objetivos del lanzamiento

- **Contar con un equipo "aglutinado" (jelled team, Tom de Marco)**

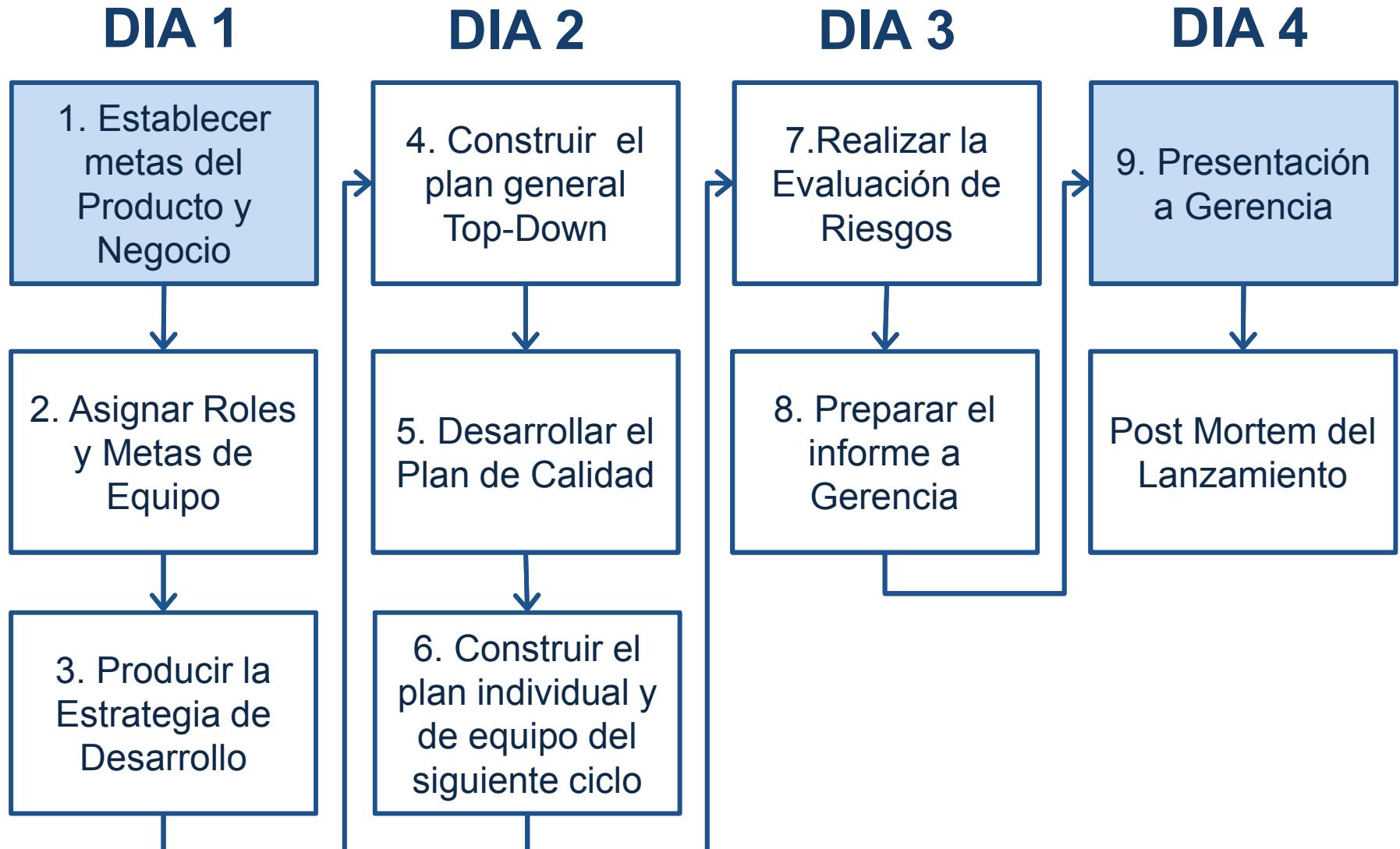
- Con objetivos compartidos
- Con un plan de acción conjunto en el que creen
- Con roles asignados (cada quien sabe lo que tiene que hacer)
- Con el control sobre su trabajo
- Con energía y entusiasmo

- **Contar con un plan agresivo pero realista**

- Individual, pero sincronizado con el equipo
- Con suficiente detalle para el trabajo diario de los siguientes meses
- Con procesos comunes y consensados
- Balanceado



Juntas del Lanzamiento





Productos resultantes

Necesidades de Negocio

Metas de Dirección de Proyectos

Requerimientos del Producto

¿QUÉ?

¿CÓMO?

¿CUÁNDO?

¿QUIÉN?

¿QUÉ TAN BIEN?

¿QUÉ PASA SI...?

Metas de Equipo

Estrategia de Equipo

Plan de Tarea/Hora

Roles de Equipo

Plan de Calidad

Evaluación de Riesgos

Diseño Conceptual

Procesos de Equipo

Plan Calendario

Plan de Tareas

Planes Alternativos

Productos Planeados

Plan de Valor Ganado

Plan Detallado

Estimación de Tamaño



Un equipo TSP utiliza varios planes

- Plan de tareas (Task Plan)**
- Plan de calendario (Schedule Plan)**
- Plan de valor ganado (Earned Value Plan)**
- Plan de calidad (Quality Plan)**
- Plan de mitigación de riesgos (Risk Plan)**

Explicación de símbolos:

- = Normalmente utilizado fuera del TSP
- = Casi nunca utilizado fuera del TSP



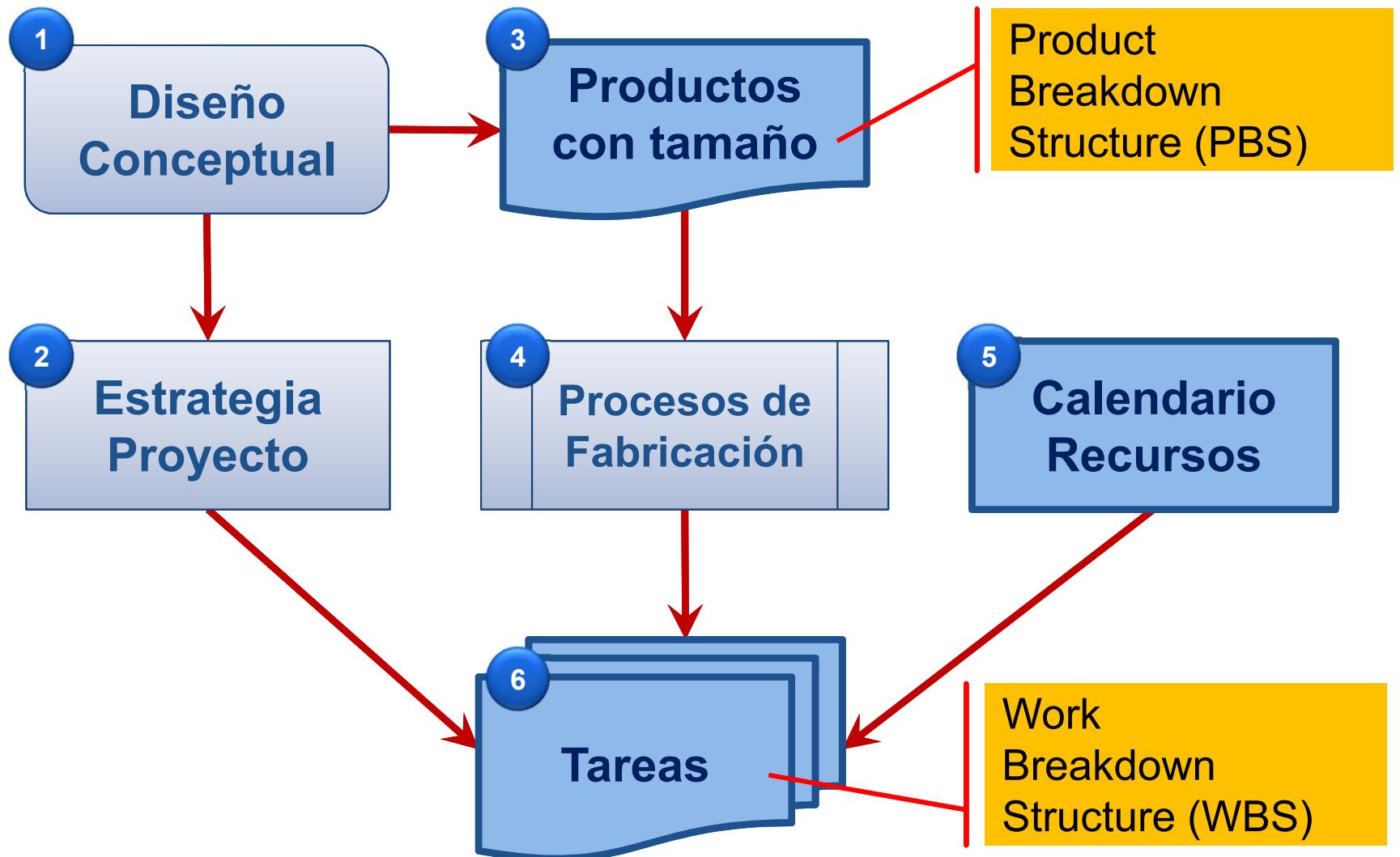
Plan de Tareas*

Assembly	Phase	Task	Resources	Est. Size	Size Measure	Engrs	Plan Hours	Plan Date	Plan Week
PF C001 Dev M2	PLAN	PF C001 Dev M2 - COD - Plan	ES	92	LOC	1.0	0.8	27/07/2009	3
PF C001 Dev M2	DLD	PF C001 Dev M2 - COD - DLD	ES	92	LOC	1.0	1.7	03/08/2009	4
PF C001 Dev M2	DLLDR	PF C001 Dev M2 - COD - DLD Rev	ES	92	LOC	1.0	0.7	03/08/2009	4
PF C001 Dev M2	TD	PF C001 Dev M2 - COD - Dis. Pruebas	ES	92	LOC	1.0	0.7	03/08/2009	4
PF C001 Dev M2	DLDINSP	PF C001 Dev M2 - COD - DLD Insp	ES	92	LOC	1.0	0.7	03/08/2009	4
PF C001 Dev M2	CODE	PF C001 Dev M2 - COD - Code	ES	92	LOC	1.0	1.7	03/08/2009	4
PF C001 Dev M2	CR	PF C001 Dev M2 - COD - Code Rev	ES	92	LOC	1.0	0.7	03/08/2009	4
PF C001 Dev M2	COMPILE	PF C001 Dev M2 - COD - Compila	ES	92	LOC	1.0	0.2	03/08/2009	4
PF C001 Dev M2	CODEINSP	PF C001 Dev M2 - COD - Code Insp	ES	92	LOC	1.0	0.7	03/08/2009	4
PF C001 Dev M2	UT	PF C001 Dev M2 - COD - UT	ES	92	LOC	1.0	0.6	03/08/2009	4
PF C001 Dev M2	PM	PF C001 Dev M2 - COD - PM	ES	92	LOC	1.0	0.2	03/08/2009	4
PF C001 Dev M1	DLDINSP	PF C001 Dev M1 - COD - DLD Insp	ES	191	LOC	1.0	1.4	03/08/2009	4
PF C001 Dev M1	CODEINSP	PF C001 Dev M1 - COD - Code Insp	ES	191	LOC	1.0	1.4	03/08/2009	4
PC C001 Dev M1	DLDINSP	PC C001 Dev M1 - COD - DLD Insp	ES	305	LOC	1.0	2.2	03/08/2009	4
PC C001 Dev M1	CODEINSP	PC C001 Dev M1 - COD - Code Insp	ES	305	LOC	1.0	2.2	03/08/2009	4
PF C001 Dev M1	PLAN	PF C001 Dev M1 - COD - Plan	MC	191	LOC	1.0	1.7	10/08/2009	5
PF C001 Dev M1	DLD	PF C001 Dev M1 - COD - DLD	MC	191	LOC	1.0	3.4	10/08/2009	5

* También conocido como WBS (Work Breakdown Structure)



Como se construye el plan de tareas (juntas 3 y 4)





Plan de Valor Ganado

Assembly	Phase	Task	Resources	Plan Hours	Plan Value	Cumulative Plan Value
PF C001 Dev M2	PLAN	PF C001 Dev M2 - COD - Plan	ES	0.8	0.14	0.14
PF C001 Dev M2	DLD	PF C001 Dev M2 - COD - DLD	ES	1.7	0.28	0.42
PF C001 Dev M2	DLLDR	PF C001 Dev M2 - COD - DLD Rev	ES	0.7	0.11	0.53
PF C001 Dev M2	TD	PF C001 Dev M2 - COD - Dis. Pruebas	ES	0.7	0.12	0.65
PF C001 Dev M2	DLDINSP	PF C001 Dev M2 - COD - DLD Insp	ES	0.7	0.11	0.76
PF C001 Dev M2	CODE	PF C001 Dev M2 - COD - Code	ES	1.7	0.28	1.04
PF C001 Dev M2	CR	PF C001 Dev M2 - COD - Code Rev	ES	0.7	0.11	1.16
PF C001 Dev M2	COMPILE	PF C001 Dev M2 - COD - Compila	ES	0.2	0.03	1.18
PF C001 Dev M2	CODEINSP	PF C001 Dev M2 - COD - Code Insp	ES	0.7	0.11	1.30
PF C001 Dev M2	UT	PF C001 Dev M2 - COD - UT	ES	0.6	0.11	1.41
PF C001 Dev M2	PM	PF C001 Dev M2 - COD - PM	ES	0.2	0.03	1.43
PF C001 Dev M1	DLDINSP	PF C001 Dev M1 - COD - DLD Insp	ES	1.4	0.24	1.67
PF C001 Dev M1	CODEINSP	PF C001 Dev M1 - COD - Code Insp	ES	1.4	0.24	1.90
PC C001 Dev M1	DLDINSP	PC C001 Dev M1 - COD - DLD Insp	ES	2.2	0.38	2.28
PC C001 Dev M1	CODEINSP	PC C001 Dev M1 - COD - Code Insp	ES	2.2	0.38	2.65
PF C001 Dev M1	PLAN	PF C001 Dev M1 - COD - Plan	MC	1.7	0.28	2.94
PF C001 Dev M1	DLD	PF C001 Dev M1 - COD - DLD	MC	3.4	0.59	3.52
PF C001 Dev M1	DLLDR	PF C001 Dev M1 - COD - DLD Rev	MC	1.4	0.24	3.76
PF C001 Dev M1	TD	PF C001 Dev M1 - COD - Dis. Pruebas	MC	1.4	0.24	4.00





Plan de Calendario

Date	Week	Plan Hours	Cumulative Plan Hours	Actual Hours	Cumulative Actual Hours
13/07/2009	1	25.4	25.4	22.7	22.7
20/07/2009	2	52.5	77.9	28.8	51.5
27/07/2009	3	85.2	163.1	46.6	98.1
03/08/2009	4	54.7	217.8	32.3	130.4
10/08/2009	5	92.6	310.4	62.0	192.4
17/08/2009	6	86.0	396.4		
24/08/2009	7	95.0	491.4		
31/08/2009	8	107.0	598.4		
07/09/2009	9	110.0	708.4		
14/09/2009	10	101.0	809.4		
21/09/2009	11	92.0	901.4		
28/09/2009	12	122.0	1023.4		
05/10/2009	13	107.0	1130.4		
12/10/2009	14	107.0	1237.4		
19/10/2009	15	98.0	1335.4		
26/10/2009	16	85.0	1420.4		
02/11/2009	17	87.0	1507.4		
09/11/2009	18	84.0	1591.4		
16/11/2009	19	81.6	1673.0		



Plan de calidad

- **Es de humanos equivocarse**
 - Por lo tanto **no** se debe castigar a alguien por equivocarse, sino por **no registrar ni corregir** la equivocación
 - Cada persona se equivoca de formas diferentes y en lugares diferentes
 - Pero cada persona normalmente se equivoca en lo mismo... ¡esa es una gran ventaja!
- **En el TSP las personas tienen datos históricos de que tan frecuentemente se equivocan y en dónde**
 - Utilizan esos datos para estimar cuantos defectos van a "inyectar" en los productos que fabrican
 - Realizan por lo tanto un plan de cómo encontrarlos y removerlos lo más pronto posible



Plan de calidad

Ejemplo:

- **Los datos históricos de Raúl muestran que en los requerimientos:**
 - Inyecta 0.6 defectos por hora al escribirlos
 - Encuentra y corrige 2.1 defectos por hora al revisarlos
 - Ha encontrado solo el 70% de los defectos
- **Según el plan de tareas de Raúl se va a tardar 26.7 horas haciendo unos requerimientos**
- **Raúl quiere ahora encontrar el 80% de los defectos**
- **¿Cuánto debe tardarse revisando sus requerimientos?**



Plan de calidad

Ejemplo:

- **Los datos históricos de Raúl muestran que en los requerimientos:**
 - Inyecta 0.6 defectos por hora al escribirlos
 - Encuentra y corrige 2.1 defectos por hora al revisarlos
 - Ha encontrado solo el 70% de los defectos
- **Según el plan de tareas de Raúl se va a tardar 26.7 horas haciendo unos requerimientos**
- **Raúl quiere ahora encontrar el 80% de los defectos**
- **¿Cuánto debe tardarse revisando sus requerimientos?**
 - A menos de que Raúl haga algo diferente a lo que ha hecho en el pasado, es muy probable que inyecte 16 defectos ($26.7 \text{ horas} * 0.6 \text{ def. por hora}$)
 - Raúl quiere encontrar 13 de ellos (80% de 16)
 - Para encontrarlos Raúl debe tardarse en la revisión 6.2 horas ($13 \text{ defectos} / 2.1 \text{ def. por hora}$).



Plan de calidad

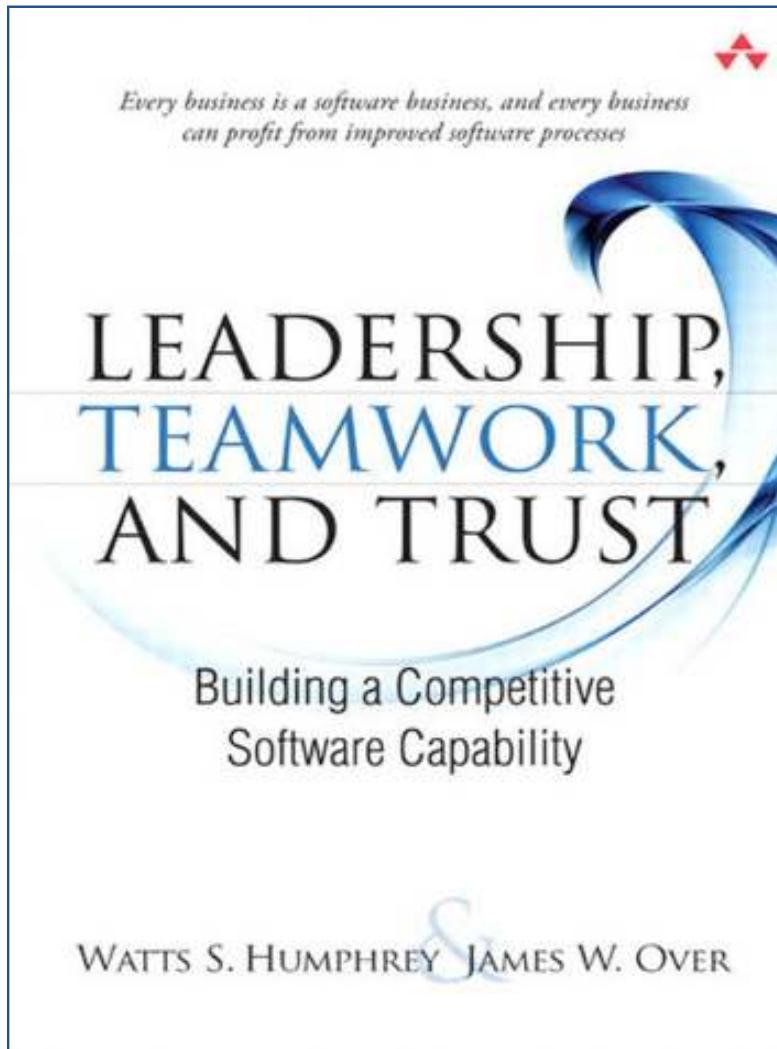
Phase	Defects Injected Per Hour	Phase Yields	Defects Injected	Defects Removed
Planning				
Requirements	0.55		52.48	
System Test Plan				
REQ Inspection		85%		44.60
High-Level Design	0.70		100.80	
Integration Test Plan	0.50		43.44	
HLD Inspection		50%		76.05
Detailed Design	0.91		91.38	
DLD Review		50%		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





Más información sobre TSP

Lo mejor es leer el último libro de Watts Humphrey:





Modelos Organizacionales: CMMI, ISO-15504 y MoProSoft



Hay muchos modelos de calidad organizacionales

■ Modelos Internacionales Generales

- ISO-9000 (2 niveles, sí o no)
- ISO-15504 (5+1 niveles)

■ Modelos Internacionales para el Sw

- CMMi SW (5 niveles)
- Otros
 - Trillium
 - Bootstrap
 - Etc.

■ Modelo Mexicano para PYMES

- Moprossoft (5+1 niveles, basado en ISO-15504)



CMMI

(Capability Maturity Model Integrated)



Historia del CMMI

1979	Modelo de madurez de Crosby
1985	Modelo de madurez en IBM (Radice)
1987→1989	Modelo de madurez de procesos de software del SEI (Watts Humphrey)
1990→1991	SW-CMM v1.0
1993	SW-CMM v1.1
1995→1997	SW-CMM v2.0 (sólo llegó al draft c)
1997→2000	CMMI v1.0
2001	CMMI v1.1
2005	SW-CMM se deja de soportar
2006	CMMI v 1.2
2010	CMMI v 1.3



Suite de Productos CMMI

- **Varias disciplinas**

- Software Engineering (SW)
- Systems Engineering (SE)
- Integrated Product and Process Development (IPPD)
- Supplier Sourcing (SS)

- **Métodos de evaluación**

- SCAMPI A, SCAMPI B y SCAMPI C

- **Entrenamiento**

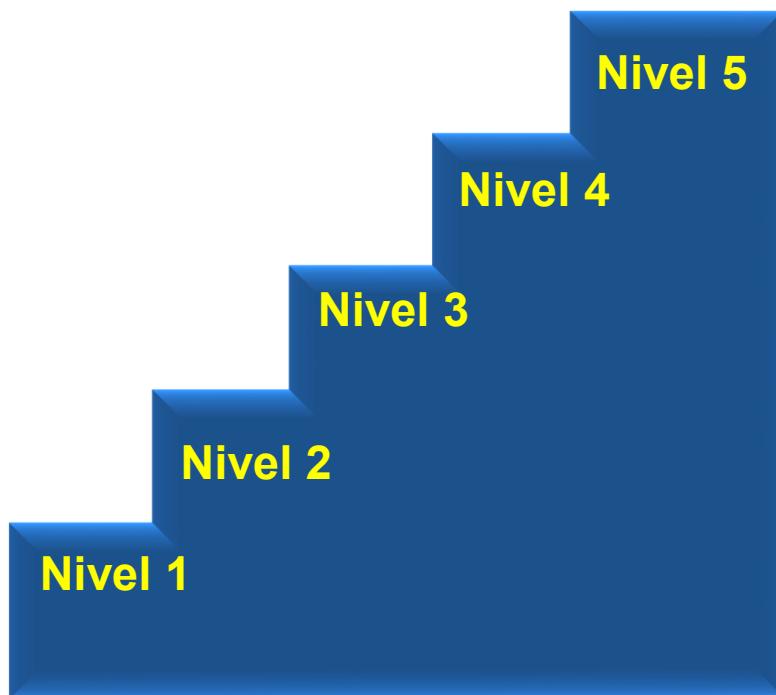
- **Dos representaciones**

- Escalonado (Staged)
- Continuo (Continuous)



Dos representaciones

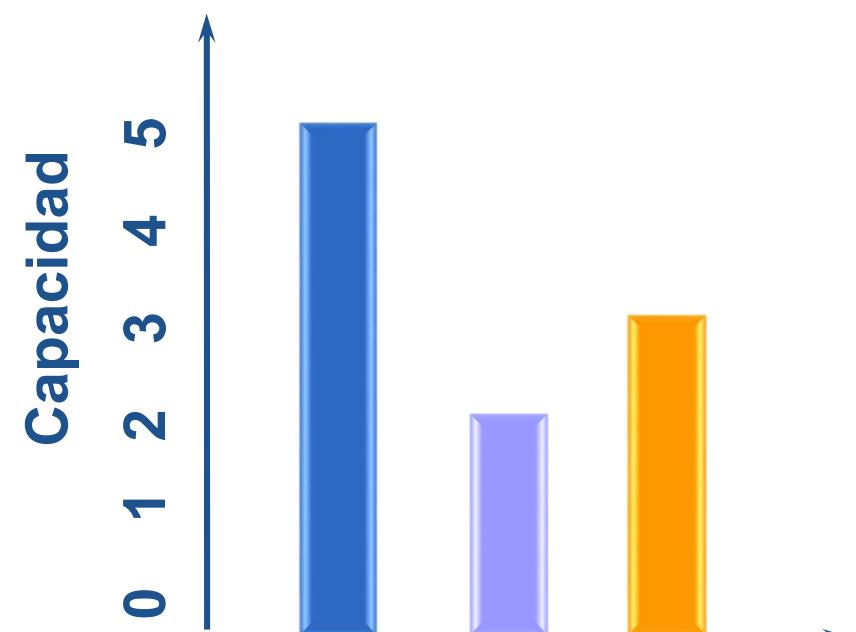
Escalonado



Organización



Continuo



Áreas de Procesos





CMMI-SW 1.2 Escalonado

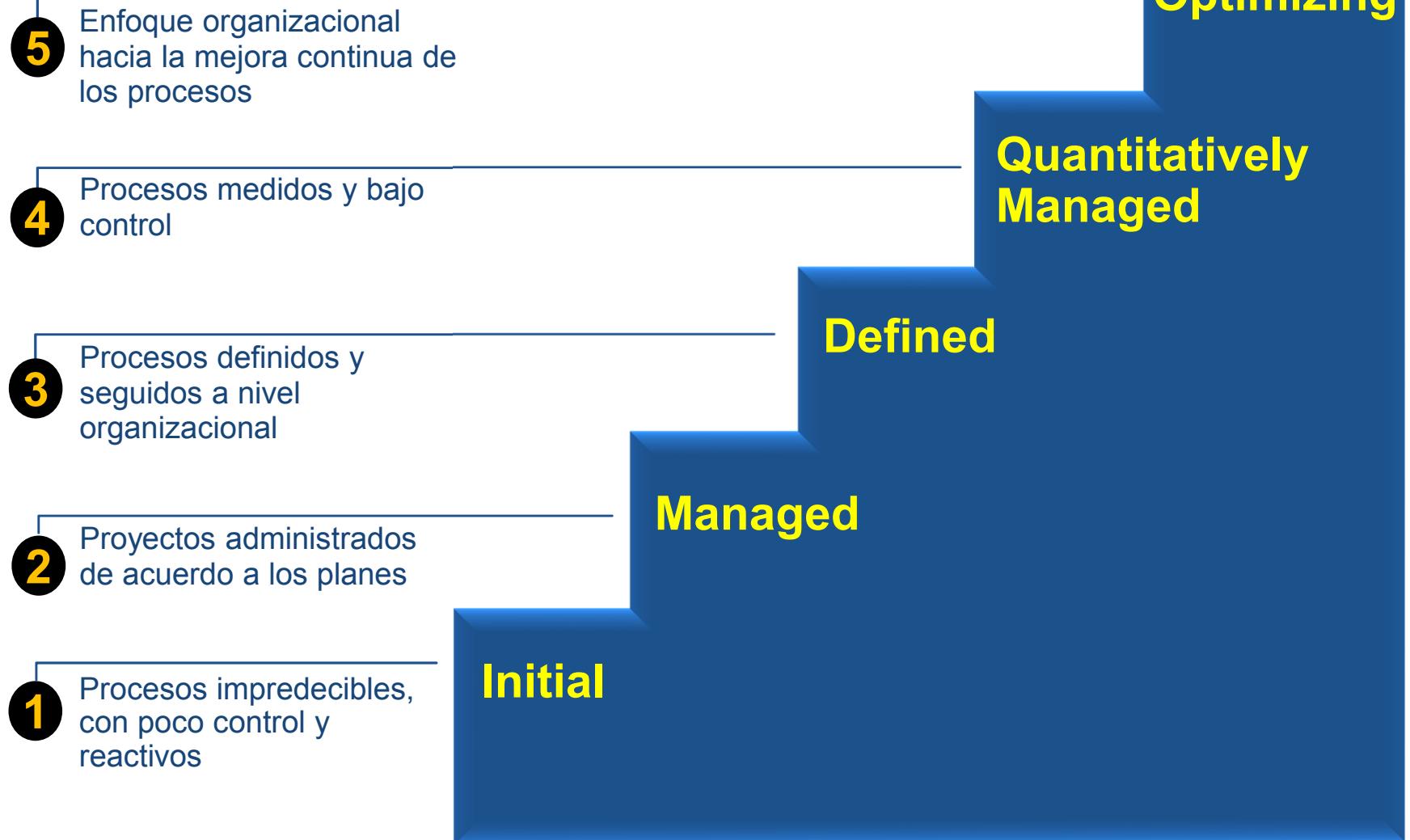


Características del modelo escalonado

- **Evalúa la madurez de una organización**
- **Define etapas por las cuales debe transitar la organización**
- **Permite comparar organizaciones fácilmente**

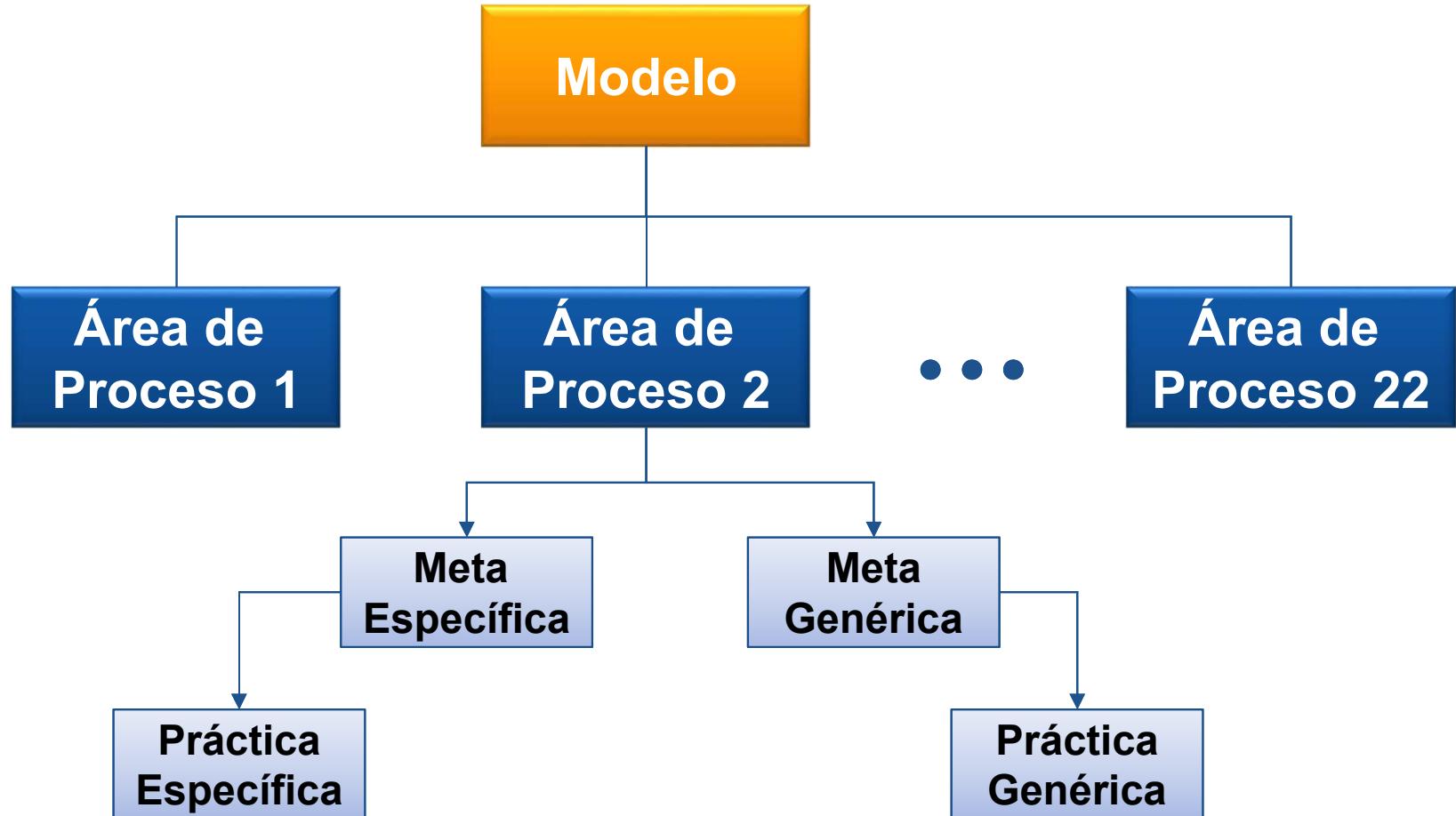


Niveles CMMI-SW 1.2





Estructura del CMMI





Process Areas (PAs)

Level	Process Area	Category
2	Requirements Management	Engineering
2	Project Planning	Project Management
2	Project Monitoring and Control	Project Management
2	Supplier Agreement Management	Project Management
2	Configuration Management	Support
2	Measurement and Analysis	Support
2	Process and Product Quality Assurance	Support
3	Product Integration	Engineering
3	Requirements Development	Engineering
3	Technical Solution	Engineering
3	Validation	Engineering
3	Verification	Engineering
3	Organizational Process Definition +IPPD	Process Management
3	Organizational Process Focus	Process Management
3	Organizational Training	Process Management
3	Integrated Project Management +IPPD	Project Management
3	Risk Management	Project Management
3	Decision Analysis and Resolution	Support



Process Areas (PAs)

Level	Process Area	Category
4	Organizational Process Performance	Process Management
4	Quantitative Project Management	Project Management
5	Organizational Innovation and Deployment	Process Management
5	Causal Analysis and Resolution	Support



Ejemplo de un PA: Project Planning (PP)

- **Purpose:**

- To establish and maintain plans that define project activities

- **Specific Goals and Practices**

- SG 1 Establish Estimates
 - SP 1.1 Estimate the Scope of the Project
 - SP 1.2 Establish Estimates of Work Product and Task Attributes
 - SP 1.3 Define Project Lifecycle
 - SP 1.4 Determine Estimates of Effort and Cost

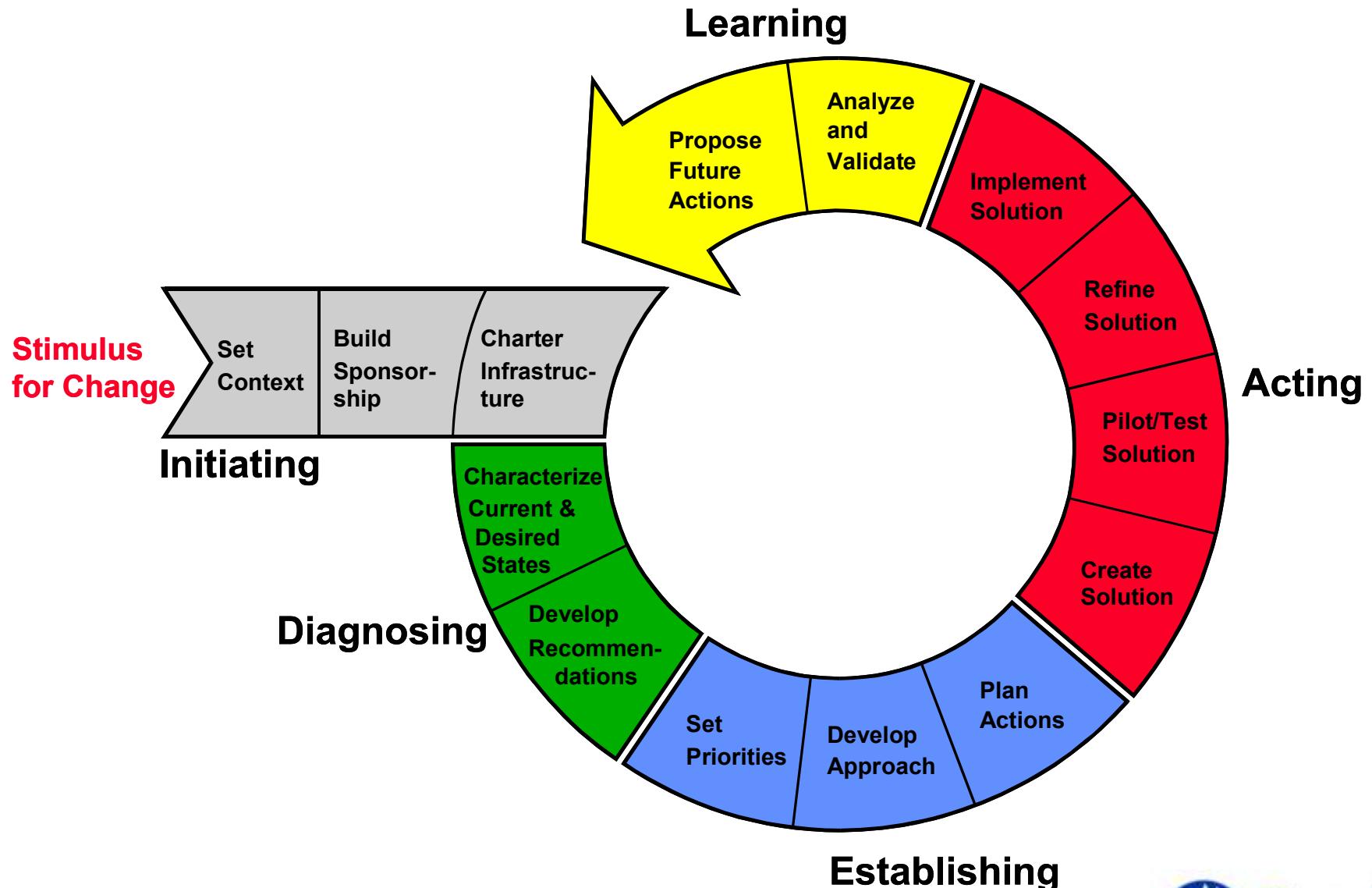


Ejemplo de un PA: Project Planning (PP)

- **Specific Goals and Practices**
 - **SG 2 Develop a Project Plan**
 - SP 2.1 Establish the Budget and Schedule
 - SP 2.2 Identify Project Risks
 - SP 2.3 Plan for Data Management
 - SP 2.4 Plan for Project Resources
 - SP 2.5 Plan for Needed Knowledge and Skills
 - SP 2.6 Plan Stakeholder Involvement
 - SP 2.7 Establish the Project Plan
 - **SG 3 Obtain Commitment to the Plan**
 - SP 3.1 Review Plans That Affect the Project
 - SP 3.2 Reconcile Work and Resource Levels
 - SP 3.3 Obtain Plan Commitment



Implementación: Modelo IDEAL del SEI



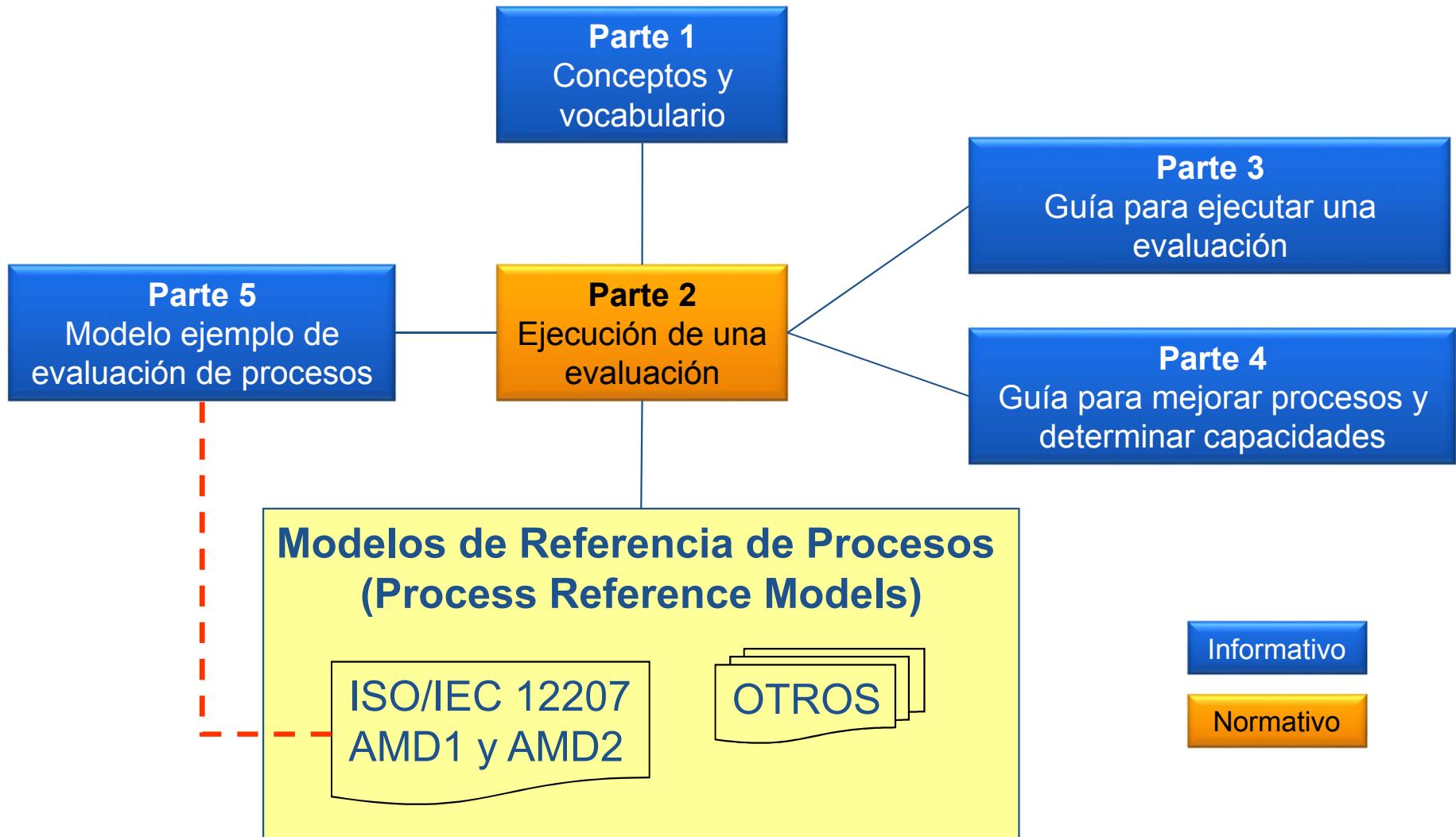


Norma Internacional ISO/IEC-15504

- **Establecido por ISO/IEC JTC1/SC7 WG10**
 - 1993: Inicio
 - 1998: Technical Review
 - 2005: Estándar
- **Evalúa la capacidad de cada proceso en vez de la capacidad organizacional**
 - Modelo Continuo
- **No sustituye a los actuales modelos de evaluación de capacidad**

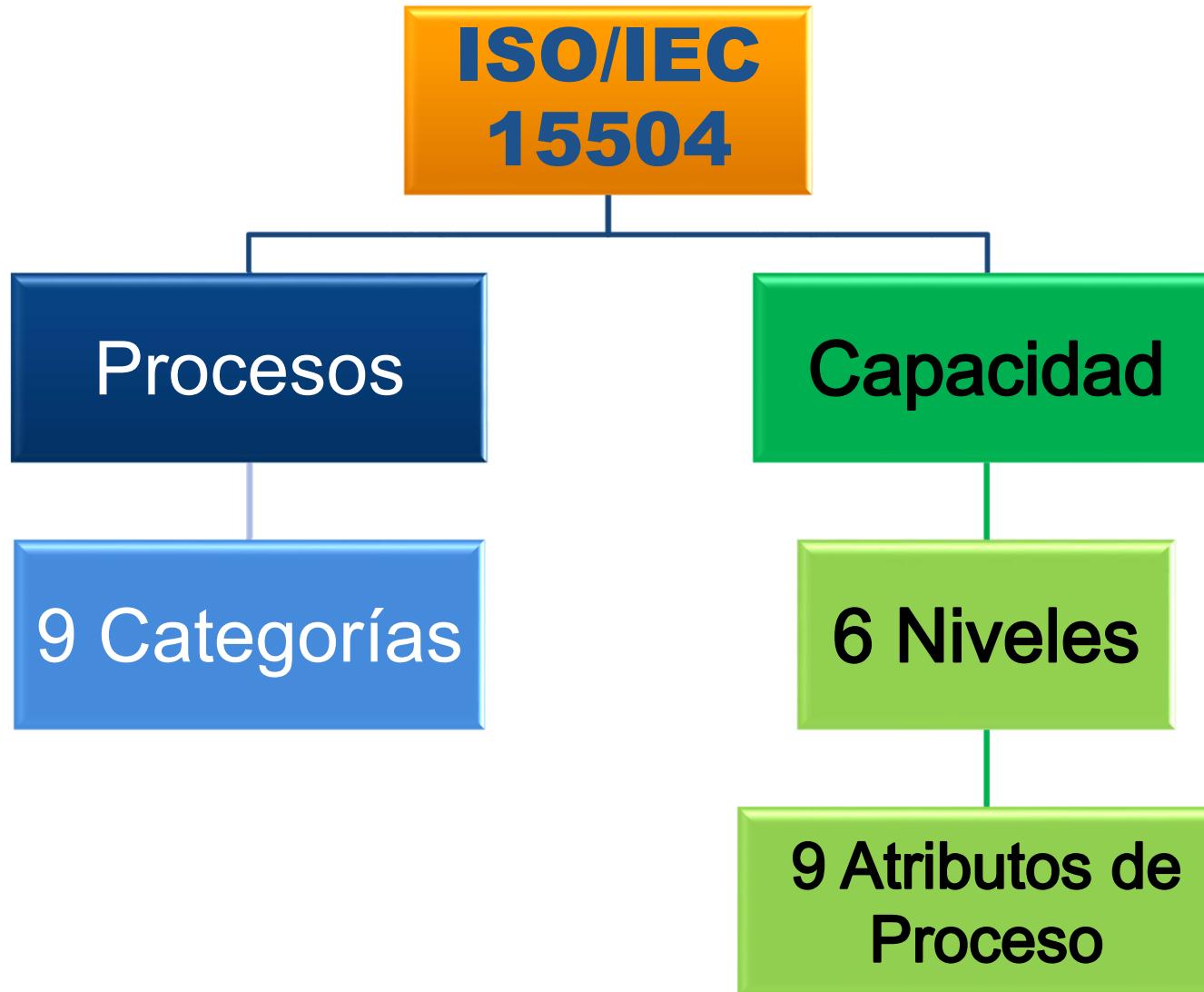


Componentes del ISO/IEC 15504





Organizado en dos dimensiones





Dimensión: Procesos

Category	Description
Acquisition	Processes performed by the customer, in order to acquire a product and/or a service.
Supply	Processes performed by the supplier in order to propose and deliver a product and/or a service.
Engineering	Processes that directly elicit and manage the customer's requirements, specify, implement, and/or maintain the software product and its relation to the system.
Operation	Processes performed in order to provide for the correct operation and use of the software product and/or service.
Supporting	Processes which may be employed, by any other processes at various points in the software life cycle.
Management	Processes that contain practices that may be used by anyone who manages any type of project or process within a software life cycle.
Process Improvement	Processes performed in order to define, deploy, assess and improve the processes performed in the organizational unit.
Resource and Infrastructure	Processes performed in order to provide adequate human resources and necessary infrastructure as required by any other process performed by the organizational unit.
Reuse	Processes performed in order to systematically exploit reuse opportunities in organization's reuse programmers.



Dimensión: Capacidad

Level	Name	Description
0	Incomplete	The process is not implemented, or fails to achieve its process purpose.
1	Performed	The process is implemented, or achieves its process purpose.
2	Managed	The previously described Performed process is now implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained.
3	Established	The previously described Managed process is now implemented using a defined process that is capable of achieving its process outcomes.
4	Predictable	The previously described Established process now operates within defined limits to achieve its process outcomes.
5	Optimizing	The previously described Predictable process is continuously improved to meet relevant current and projected business goals.



La capacidad se mide con Atributos de Proceso

Process Attribute	Description
1.1 Process Perf.	A measure of the extent to which the process purpose is achieved
2.1 Performance Mgm	A measure of the extent to which the performance of the process is managed.
2.2 Work Product Mgm	A measure of the extent to which the work products produced by the process are appropriately managed.
3.1 Proc. Definition	A measure of the extent to which a standard process is maintained to support the deployment of the defined process.
3.2 Proc. Deployment	A measure of the extent to which the standard process is effectively deployed as a defined process to achieve its process outcomes.



La capacidad se mide con Atributos de Proceso

Process Attribute	Description
4.1 Proc. Measurement	A measure of the extent to which measurement results are used to ensure that performance of the process supports the achievement of relevant process performance objectives in support of defined business goals.
4.2 Proc. Control	A measure of the extent to which the process is quantitatively managed to produce a process that is stable, capable, and predictable within defined limits.
5.1 Proc. Innovation	A measure of the extent to which changes to the proc. are identified from analysis of common causes of variation in performance, and from inv. of innovative approaches to the definition and deployment of the proc.
5.2 Proc. Optimization	A measure of the extent to which changes to the definition, management and performance of the process result in effective impact that achieves the relevant process improvement objectives.

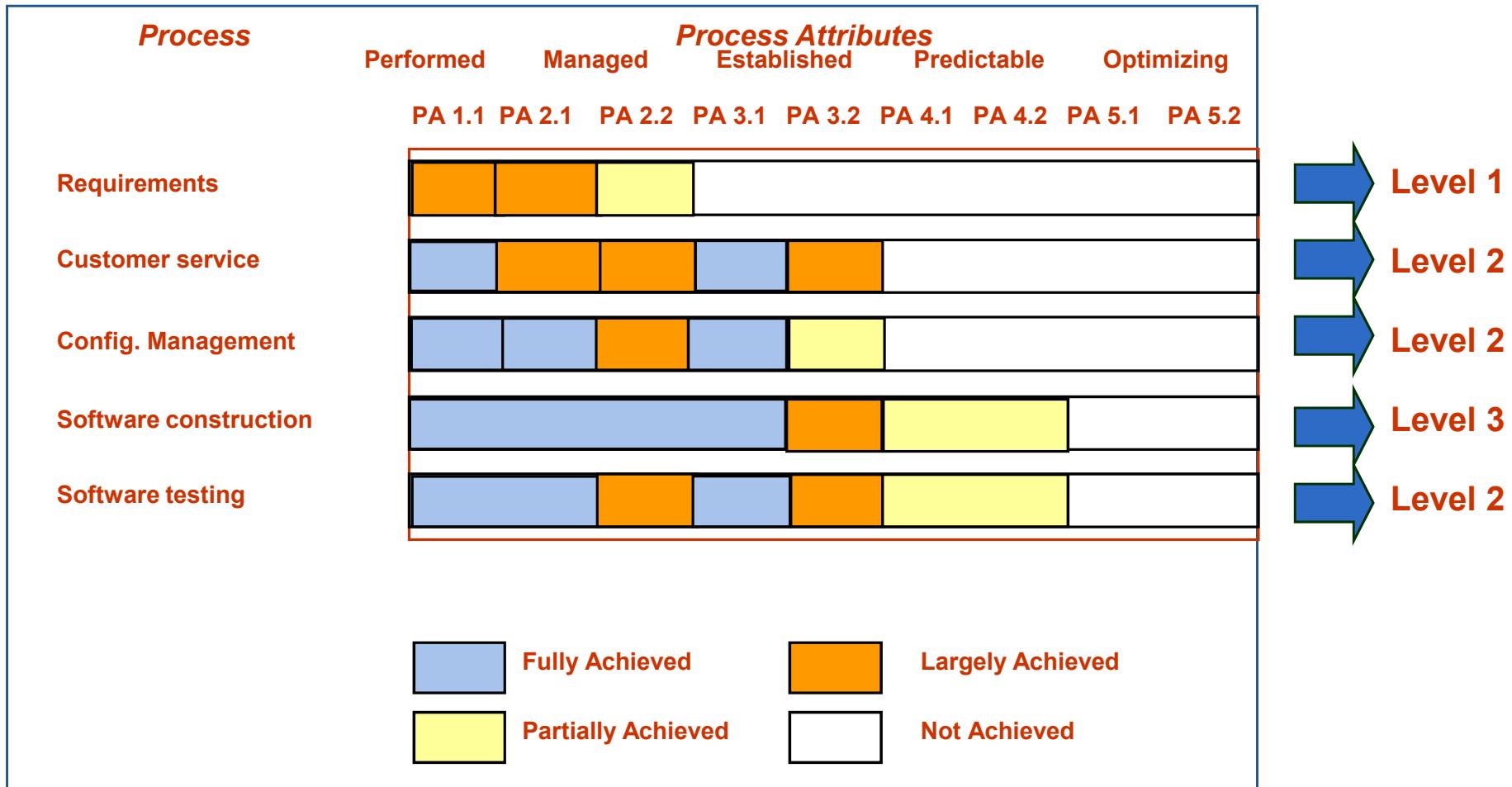


Como se evalúa cada proceso

Atributos del Proceso	Nivel de Capacidad				
	1	2	3	4	5
1.1 Process Performance	■				
2.1 Performance Management		■			
2.2 Work product management		■			
3.1 Process definition			■		
3.2 Process deployment			■		
4.1 Process measurement				■	
4.2 Process control				■	
5.1 Process innovation					■
5.2 Process Optimization					■

■ Largely Achieved (50-85%) ■ Fully Achieved (>85%)

Ejemplo





Norma Mexicana: MoProSoft

(Modelo de Procesos para la Industria de Software)



Estructura de MoProSoft

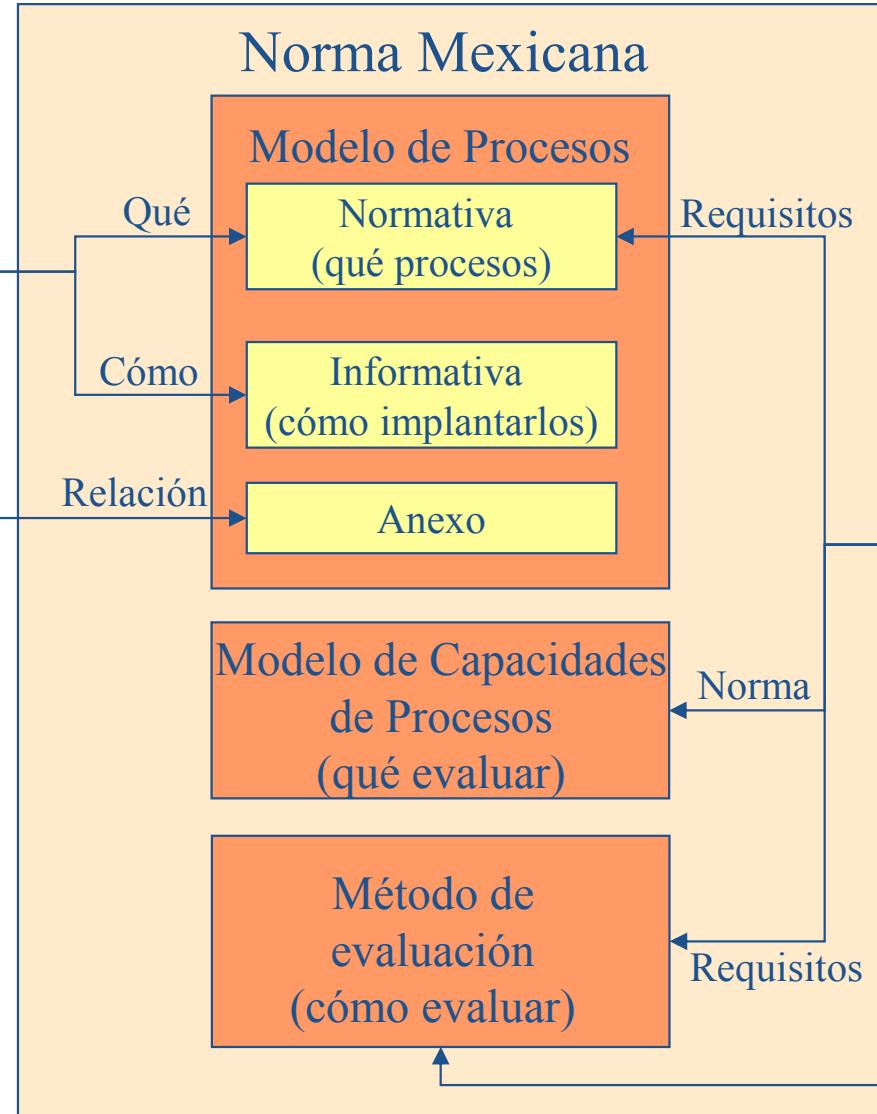




Basado en ISO 15504

MOPROSOFT

ISO 12207
Software life cycle processes



ISO/IEC FDIS 15504-2
Performing an assessment

ISO/IEC FDIS 15504-3
Guidance on performing
an assessment

Guías



Tecnológico
de Monterrey



En resumen

- **Nuestros clientes quieren:**

Calidad + Productividad
↓ ↓
Cero defectos + Oportuno y a tiempo

- **Esto no se da por “accidente”**

El software se desarrolla por gente, y... el ser humano no es de “alto rendimiento” sin disciplina.



Por ello se requiere
disciplina de procesos

Disciplina de
procesos en:

CMMI
ISO-15504
Moprossoft

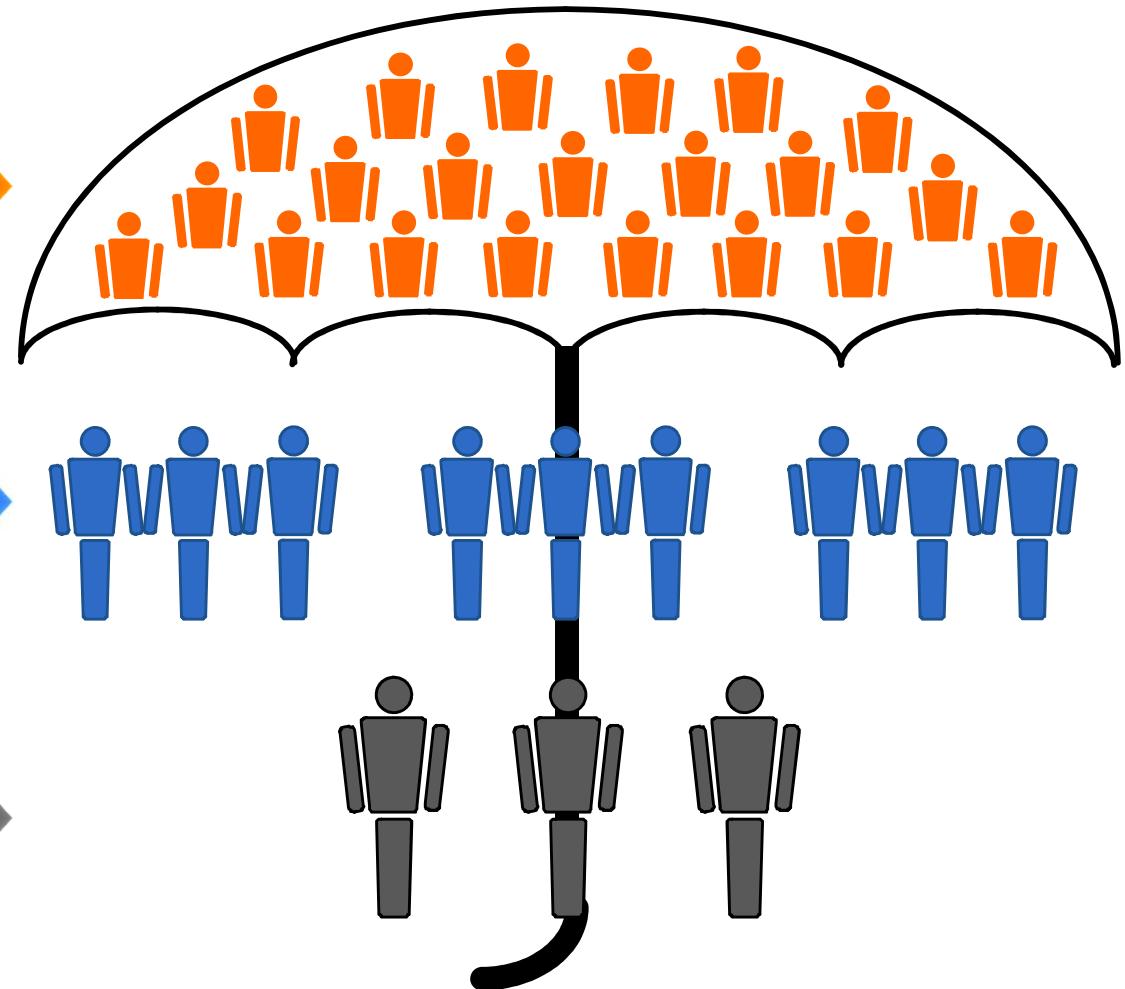
Organización

TSP

Equipos

PSP

Individuos



CMMI es una marca registrada por Carnegie Mellon University y Team Software Process, TSP, Personal Software Process y PSP son marcas de servicio registradas por Carnegie Mellon University



Tecnológico
de Monterrey

Muchas Gracias

Ing. Rafael Salazar Chávez
Director de la Iniciativa Mexicana TSP
ITESM, Campus Monterrey
rafael.salazar@itesm.mx