Tecnológico de Monterrey

Módulo 1: Calidad y Enfoque a Procesos

Introducción a la Calidad en el Software (Parte 2)

Contenido del curso

Clases:

- **M1**: Calidad y enfoque a procesos
- **M2**: Modelos de calidad organizacionales
- **M3**: Modelos de calidad para equipos y personas
- **M4**: Proceso y herramien-tas de pruebas
- **M5**: Diseño de Casos de Prueba

Laboratorio:

- **PSP 0**: Métricas de tiempo y defectos
- **PSP 1**: Estimación y métricas de tamaño
- **PSP 2**: Calidad a través de revisiones
- **PSP 2.1**: Calidad a través del diseño

Recordando (del video)

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**

Enemigo a eliminar…

Tecnológico de Monterrey

Necesitamos remover y prevenir los defectos desde el inicio

Continuemos...

Tecnológico de Monterrey

Limitantes de las PRUEBAS

- Es difícil estimar cuanto nos vamos a tardar
  - ¿Cuanto me voy a tardar en corregir un bug?
  - ¿Cómo se que ya terminé de arreglar todo?
- No son muy efectivas
  - Datos muestran que las pruebas detectan entre un 35% a un 50% de los defectos
- “Desbaratan” la “elegancia” del código
- Se realizan con mucho estrés
  - Muy cerca de entregar (con el cliente y el jefe encima)
- Es un tiempo “desperdiciado” (no genera valor)
- Son muy costosas

Proceso para remover defectos

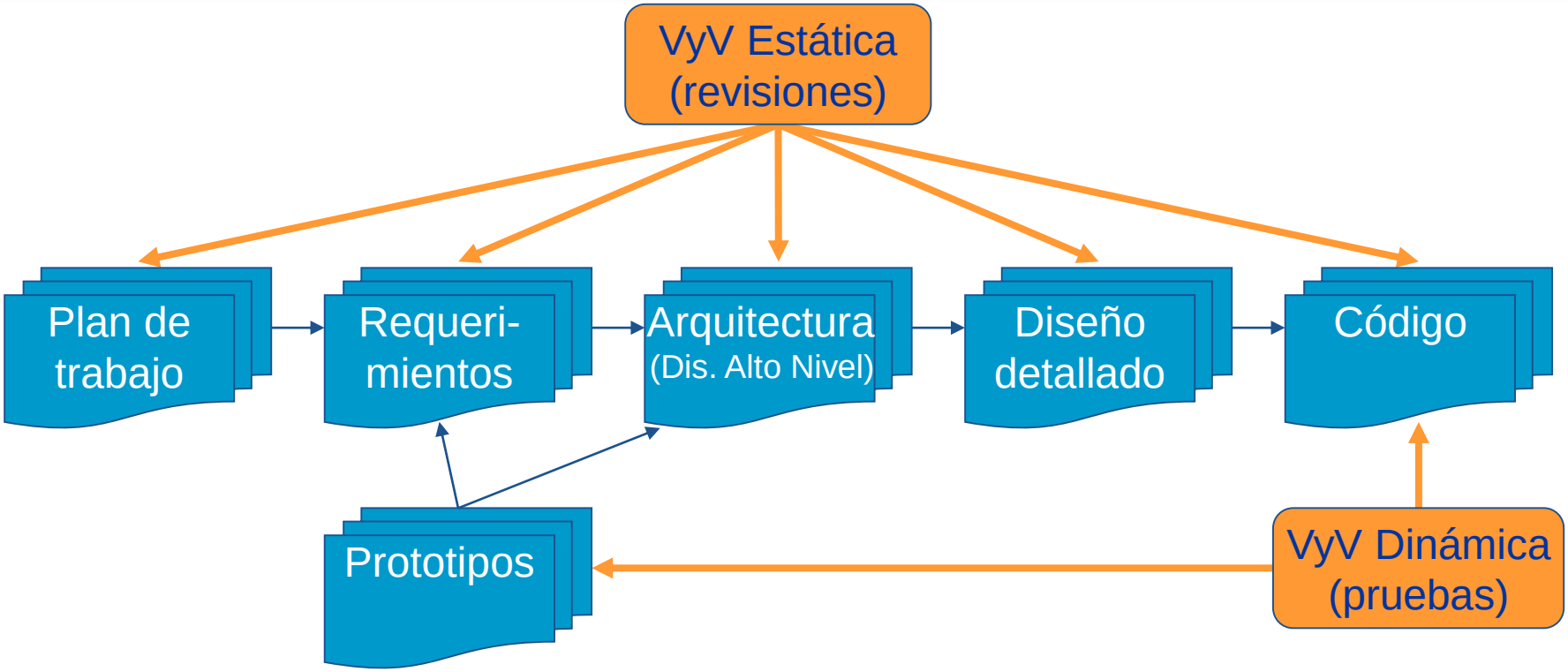
- Verificación
  - ¿Estamos construyendo el producto de forma correcta?
  - Orientado a Proceso
- Validación
  - ¿Estamos construyendo el producto correcto?
  - Orientado a Producto

Siempre debemos hacer las 2

Tipos de VyV

- VyV Dinámica
  - Ejercitar y observar comportamiento del producto
  - Normalmente → pruebas del software
- VyV Estática
  - Analizar representación estática del sistema
  - Normalmente → revisiones de “productos de trabajo” (también llamados “artefactos”)

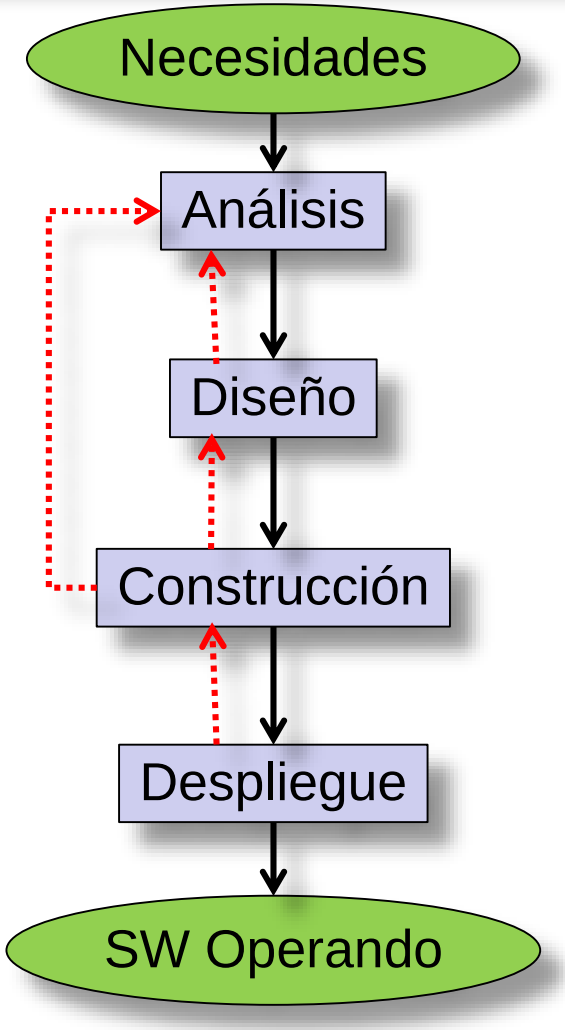
Dónde se aplica la VyV



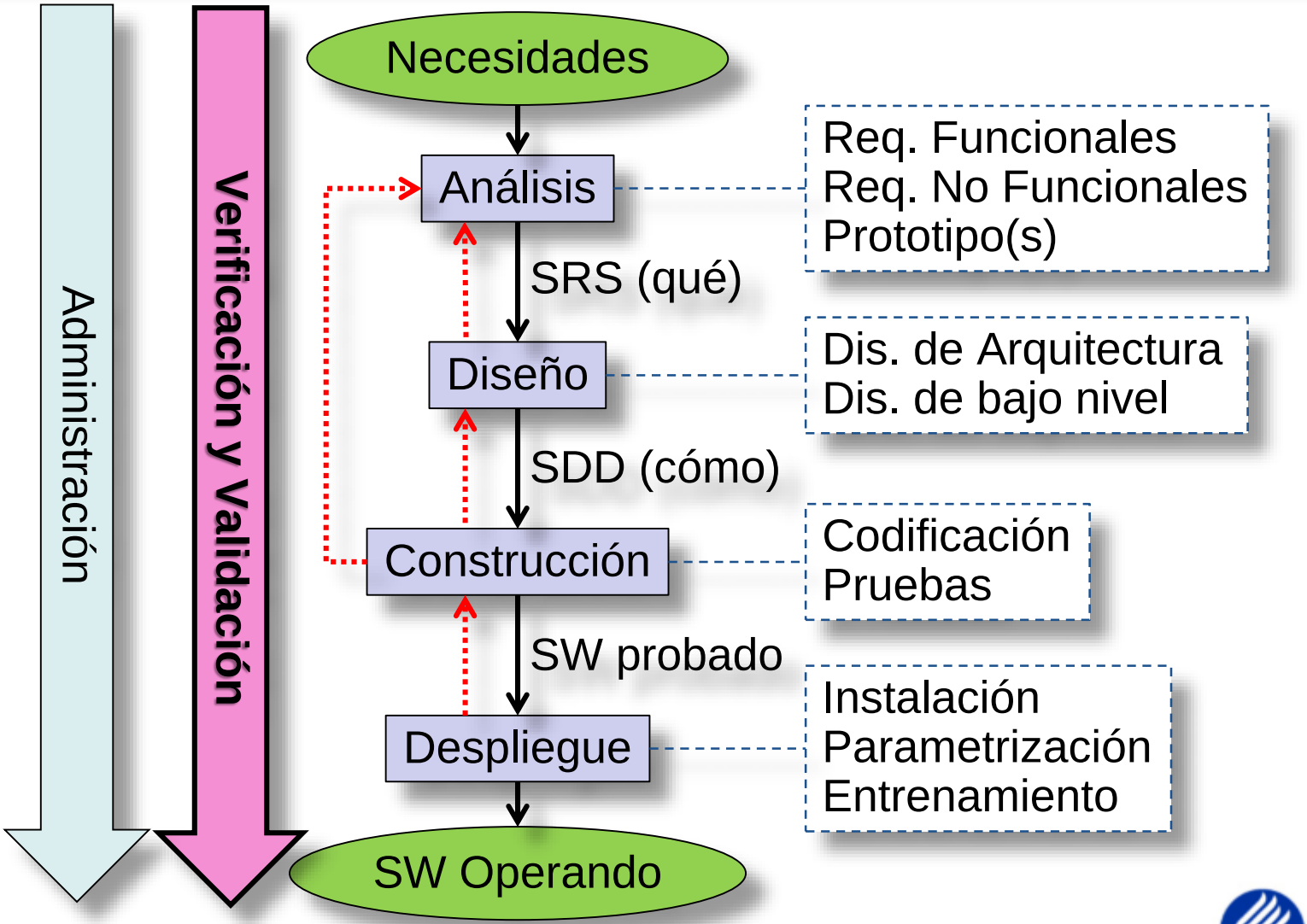
= Producto de Trabajo

Actividad

- En parejas:
  - Modifiquen la siguiente gráfica, que representa el proceso de desarrollo de software, para incluir el proceso de VyV
- Discusión en plenario



Proceso de Desarrollo de SW



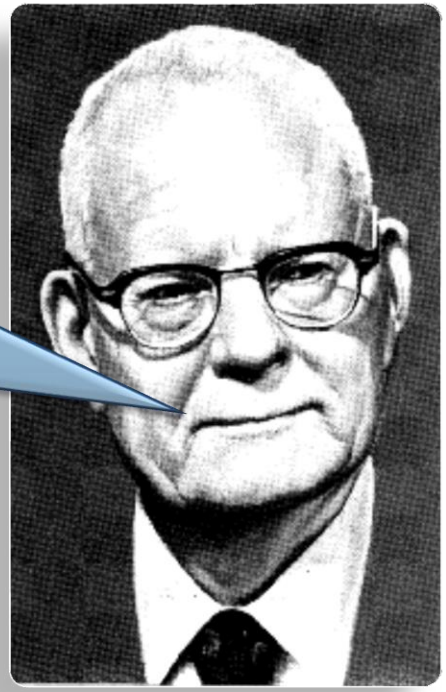
Hablemos ahora de datos…

¿Se puede hablar de una "Ingeniería" sin datos?

Los datos fueron el pilar de la calidad en la manufactura

- Lamentablemente estamos muy acostumbrados a proponer mejoras sin datos

Only "in God we trust"… all others must bring data



W. Edwards Deming (1900-1993)

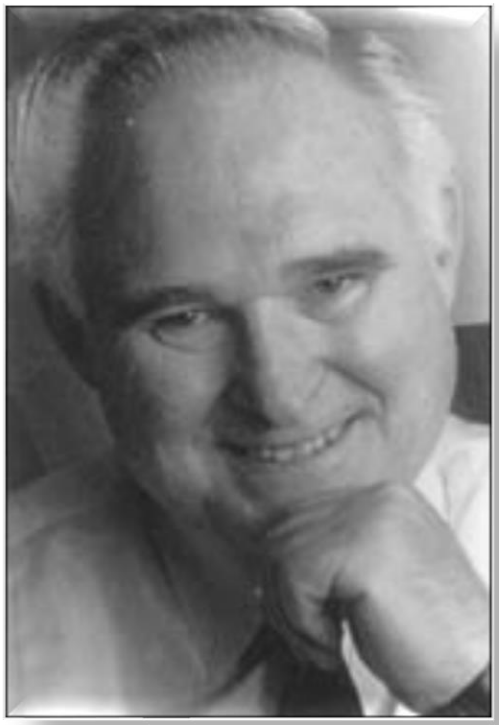
Tecnológico de Monterrey

¿Por qué necesitamos medir nuestro trabajo?

James Harrington dijo:

La medición es el primer paso que nos guía al control para después llevarnos a la mejora

- Si no mides algo, no lo podrás entender.
- Si no lo entiendes, no lo podrás controlar.
- Si no lo tienes bajo control, no lo podrás mejorar.



Medición → **Entendi-miento** → **Control** → **Mejora**

¿Qué métricas necesitamos?

- Sin datos reales y comparables no se puede administrar apropiadamente
  - Adm. de fecha y costo → tiempo y calendario (fechas)
  - Adm. de calidad → defectos
  - Para estimar y comparar → tamaño
- El único que puede recolectar los datos reales es el desarrollador
  - La única forma de que sean reales es recolectarlos al mismo tiempo que se trabaja
  - La única forma de que sean comparables es seguir un proceso estándar

Actividad

- En laptop o celular:
  - Entrar a menti.com
  - Escribir número mostrado en pantalla por el profesor
- Seguir las instrucciones del profesor
- Escribir respuesta en menti.com

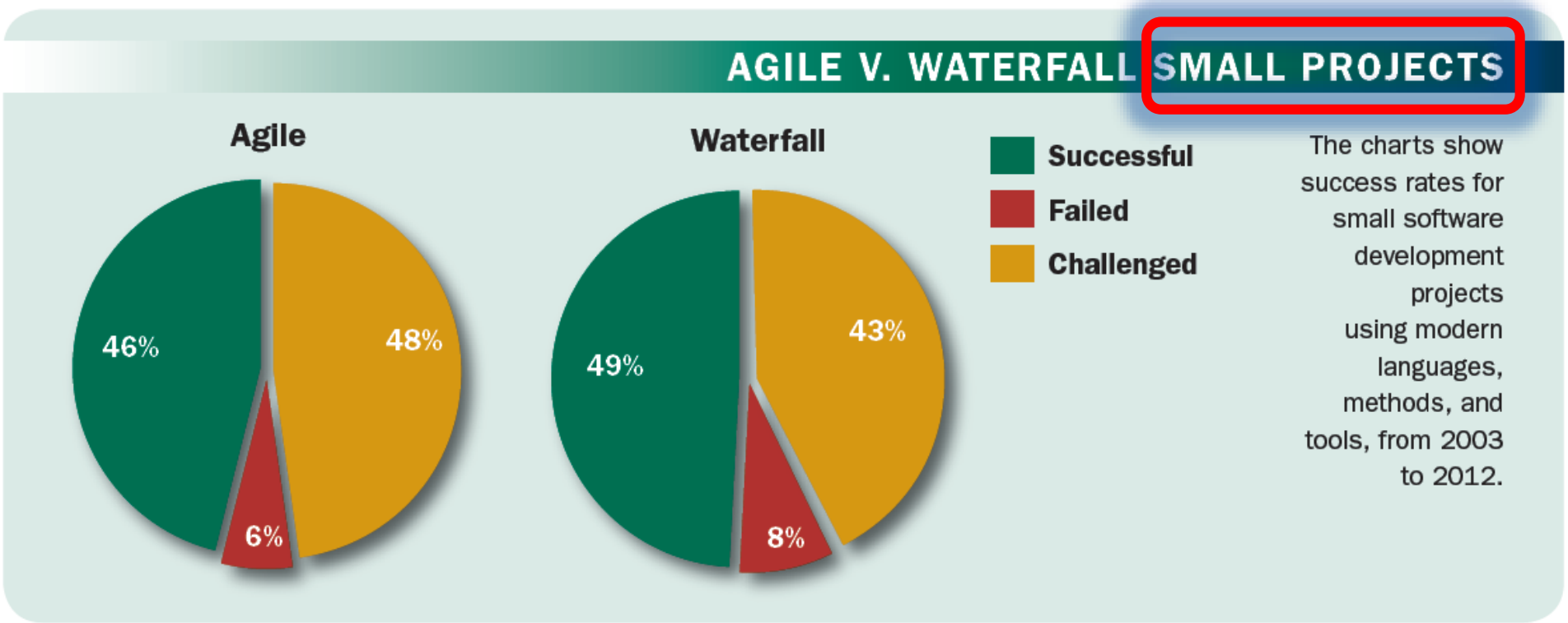
Buenos datos = buenas decisiones

Sondeo

- Instrucciones:
  - Entra a la pagina **www.socrative.com**
  - Haz login como **estudiante**
  - Escribe **CPS** en el nombre del cuarto
  - Selecciona la respuesta correcta a la siguiente pregunta

- **¿Cuál metodología es mejor?**
  a) Ágil
  b) Cascada
  c) Ambas son iguales

Ágil vs Cascada



Fuente: Chaos Manifesto 2013, Standish Group

Comparativo de Metodologías

RAW DATA:	SPEED*			QUALITY			ECONOMICS	
Methodologies	Schedule Months	Effort Months	Development Cost	Defect Potential	Defects Delivered	Hi Sev. Defects	TCO	COQ
Agile - Extreme Prog (XP)	11.78	84	$630,860	4,500	299	55	$1,318,539	$627,106
Agile - SCRUM	11.82	84	$633,043	4,800	370	68	$1,467,957	$774,142
CMMI 1 + waterfall	15.85	158	$1,188,870	6,000	1,274	236	$3,944,159	$2,804,224
CMMI 3 + iterative	13.34	107	$800,113	4,500	397	73	$1,748,043	$925,929
CMMI 5 + spiral	12.45	83	$622,257	3,000	122	22	$1,034,300	$377,880
Object Oriented Prog.	12.78	107	$805,156	4,950	310	57	$1,617,839	$735,388
RUP	13.11	101	$756,157	3,900	192	36	$1,360,857	$506,199
TSP	12.02	86	$644,070	2,700	87	16	$1,026,660	$298,699
Average	**13.00**	**112.6**	**$844,852**	**4,370**	**374**	**69**	**$1,784,238**	**$866,996**

* Tamaño = 75,000 LDC; Lenguaje = C++

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

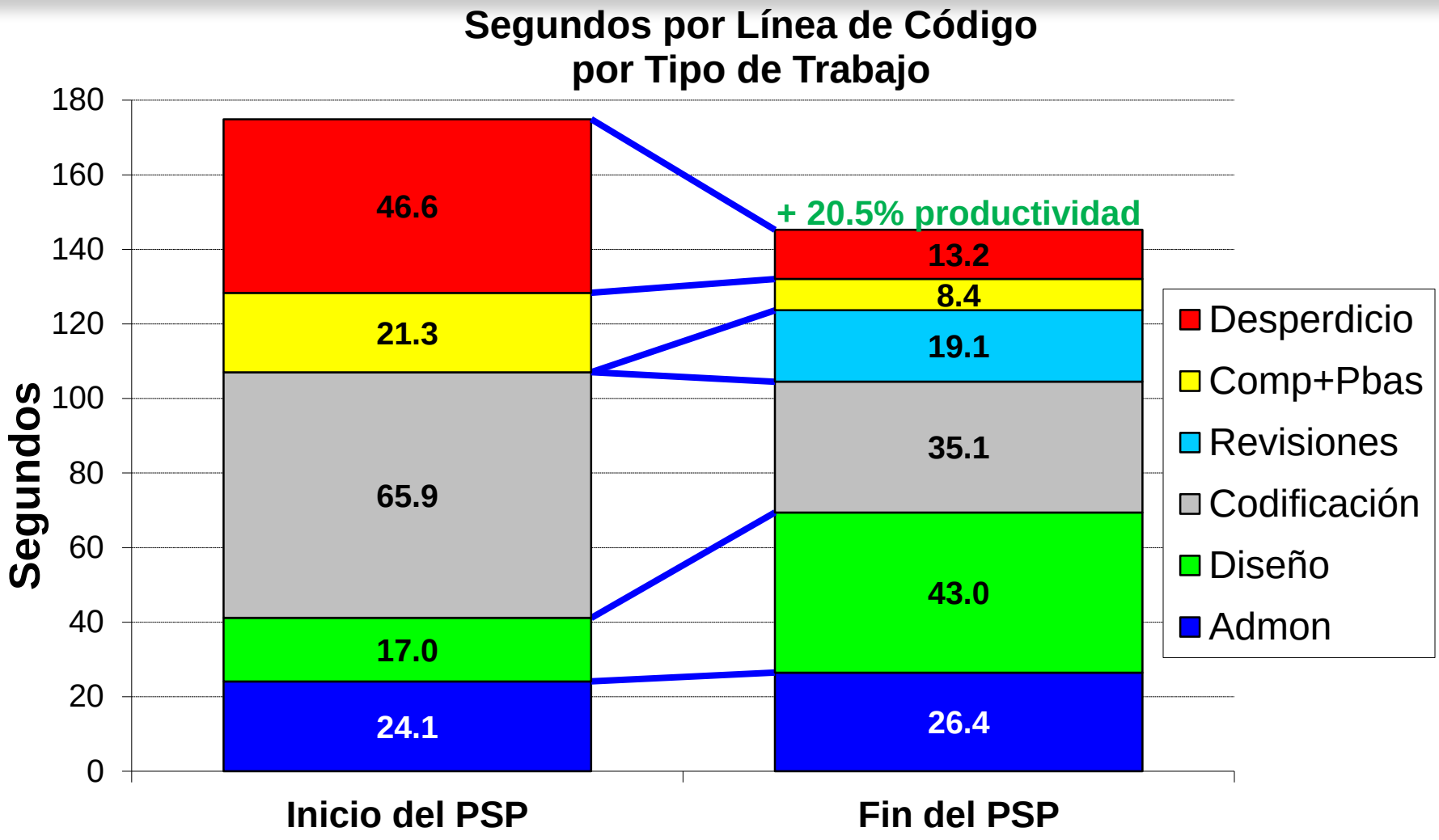
http://www.infoq.com/articles/evaluating-agile-software-methodologies

Sondeo

- Instrucciones:
  - Entra a la pagina **www.socrative.com**
  - Haz login como **estudiante**
  - Escribe **CPS** en el nombre del cuarto
  - Selecciona la respuesta correcta a la siguiente pregunta

- **Para ser más productivos ¿Cuánto tiempo debemos dedicar a diseñar nuestro programa antes de teclear la primer LDC?**
  a) Entre 0% y 15% del tiempo de codificación
  b) Entre 16% y 30% del tiempo de codificación
  c) Entre 31% y 45% del tiempo de codificación
  d) Entre 46% y 60% del tiempo de codificación
  e) Más de 60% del tiempo de codificación

Datos de developers de la industria



Fuente: 26 profesionistas **Mexicanos** utilizando lenguajes tipo C++

Tecnológico de Monterrey

Lo barato sale caro...

- Con las prácticas actuales:
  - Si dedicas 174.9 seg/LDC
    - Generas 20,583 LDC en un año (1 año ~ 1,000 horas)
  - Si 46.6 seg/LDC son para arreglar defectos
    - Desperdicias 266.4 horas en un año (1.3 días por semana)
- Con buena ingeniería:
  - Si dedicas 145.2 seg/LDC
    - Generarás 24,793 LDC en un año (20.5% más de código)
  - Si sólo 13.2 seg/LDC son para arreglar defectos
    - Reducirás el desperdicio a 90.9 horas en un año (0.46 d/s)
    - Recuperas 175.5 horas (0.88 días por semana)
- Y todavía faltan las horas para arreglar los defectos encontrados en las demás fases de pruebas!!!!

Estudio realizado en México

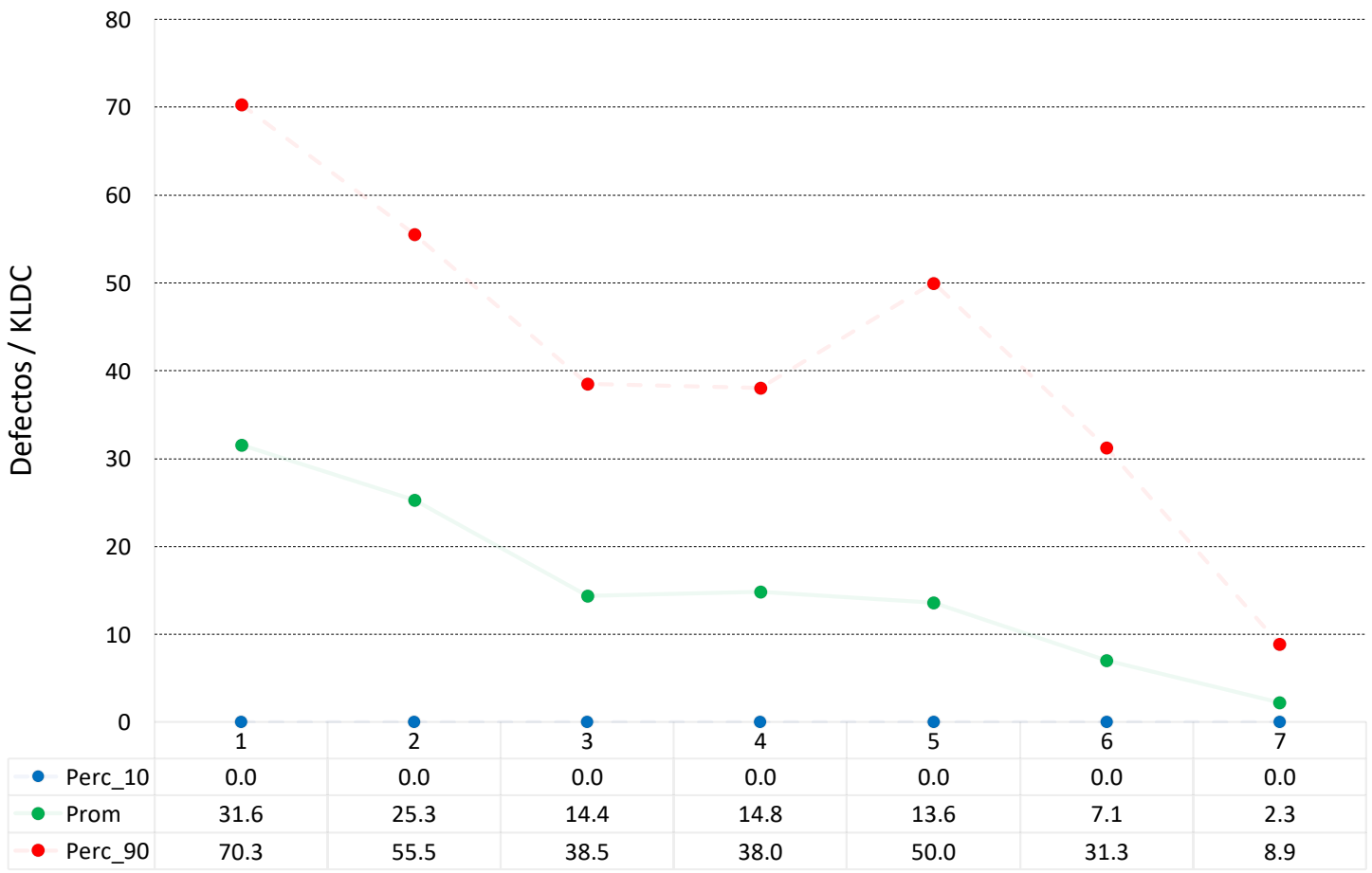
- Empresa grande, CMMI 5, proyecto grande
  - Tamaño producto: ≈400 KLDC
  - Tamaño equipo: ≈30 gente
  - Defectos entregados: 14 (0.035 defectos/KLDC)
  - Incremento en productividad (vs. no-TSP): 34.5%
- Empresa MUY pequeña, CMMI 1, proy. pequeño

	Proy. 1	Proy. 2	Proy. 3	Proy. 4	4 vs 1
Tamaño del equipo	4	4	4	3	-
Tamaño producto (KLDC)	10.7	7.5	21.6	13.5	-
Def. entregados / KLDC	1.03	0.93	0.18	0.15	**-85%**
Productividad en LDC/hr	7.0	8.1	11.2	12.9	**+84%**

Fuente: SEI, reporte técnico CMU/SEI-2009-TR-011

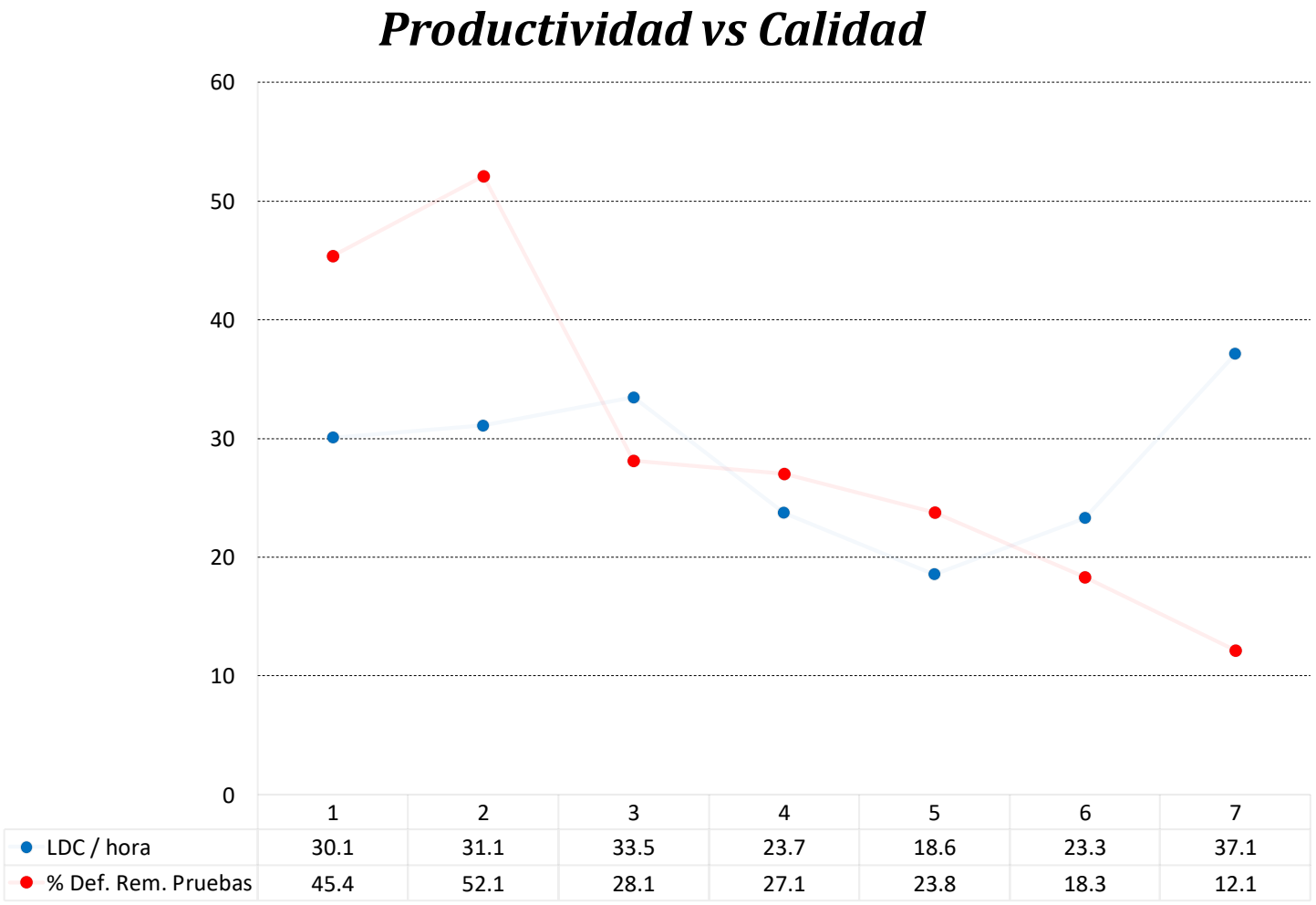
Datos de 175 alumnos del Tec

Defectos Removidos en Pruebas (def/KLDC)



	1	2	3	4	5	6	7
Perc_10	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Prom	31.6	25.3	14.4	14.8	13.6	7.1	2.3
Perc_90	70.3	55.5	38.5	38.0	50.0	31.3	8.9

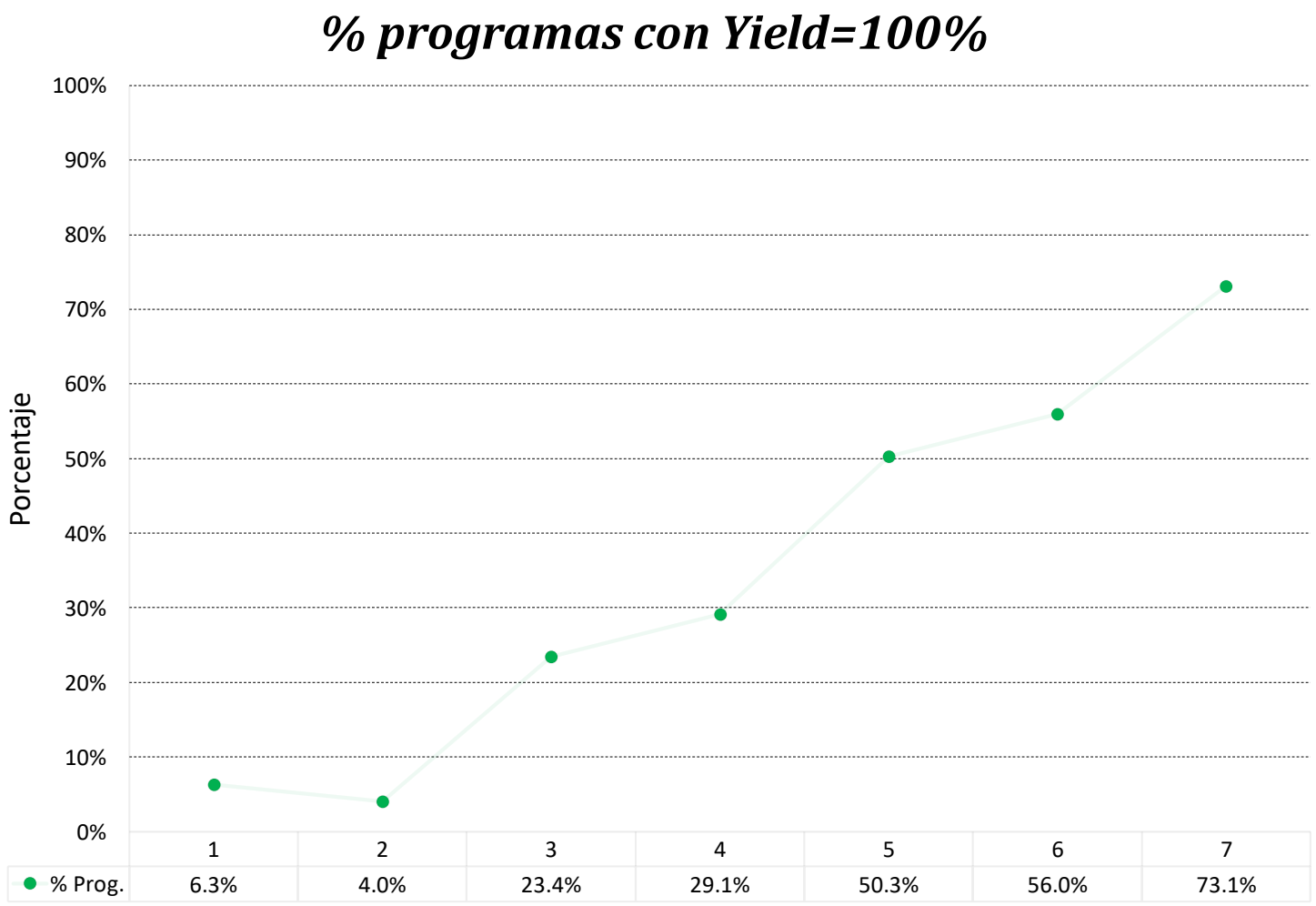
Productividad vs. Calidad



	1	2	3	4	5	6	7
LDC / hora	30.1	31.1	33.5	23.7	18.6	23.3	37.1
% Def. Rem. Pruebas	45.4	52.1	28.1	27.1	23.8	18.3	12.1

Tecnológico de Monterrey

% Programas con cero defectos en compilación y Pruebas



Tecnológico de Monterrey

Costo de arreglar defectos

Promedio de minutos dedicados a corregir cada defecto:

	Fase Removido				
Fase Inyectado	**Rev. Diseño**	**Rev. Código**	**Compilación**	**Pruebas**	
Diseño	2.62	3.06	8.56	10.21	**3.9x**
Codificación		1.71	1.93	8.39	**4.9x**

N = 272 personas