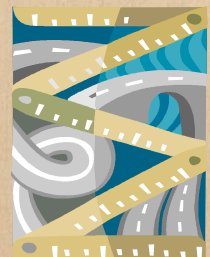


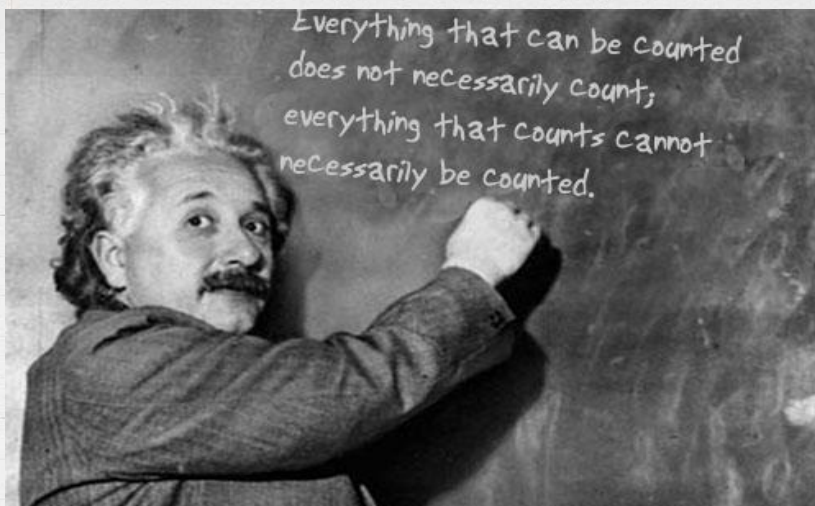
Universidad Tecnológica Nacional - Facultad Regional Córdoba
Cátedra de Ingeniería de Software
Docentes: Judith Meles - Laura Covaro

Métricas de Software



1

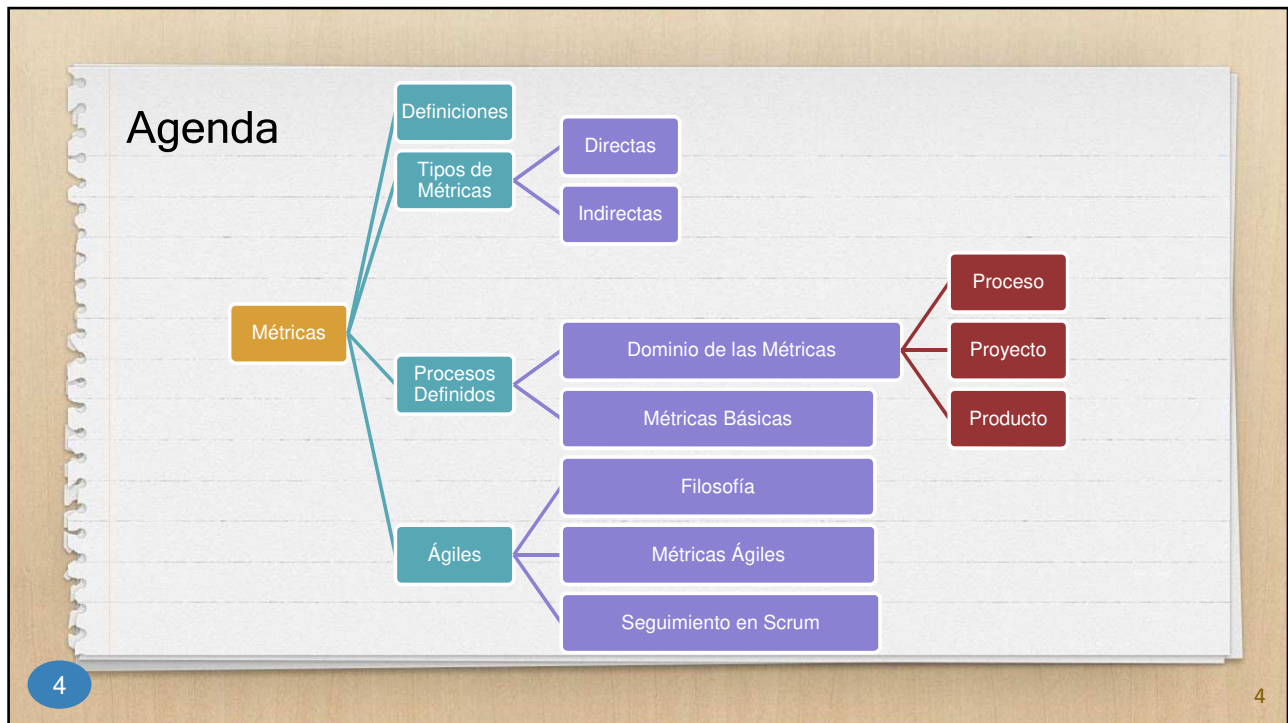
1



Sabiduría
de
Einstein

2

2



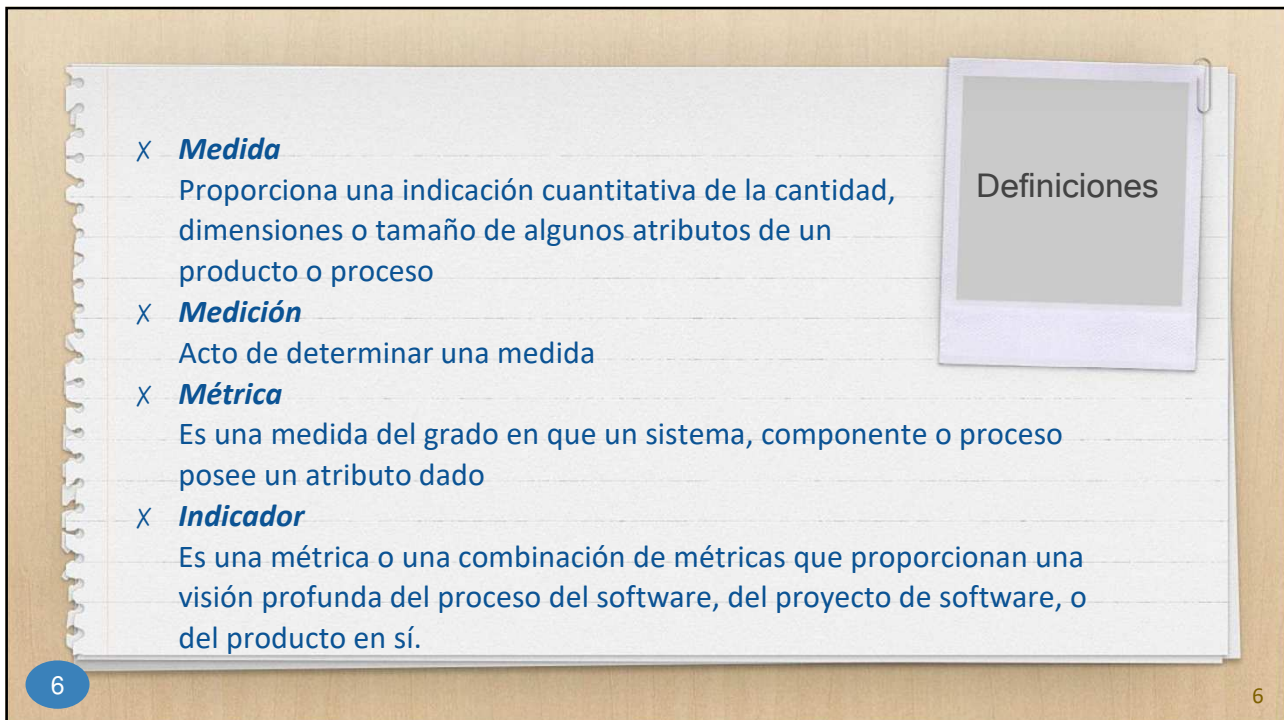
4

MEDIMOS POR UNA NECESIDAD...

- X De informar
- X De motivar
- X De comparar
- X De entender
- X De evaluar
- X De predecir
- X De mejorar

5

5



X **Medida**
Proporciona una indicación cuantitativa de la cantidad, dimensiones o tamaño de algunos atributos de un producto o proceso

X **Medición**
Acto de determinar una medida

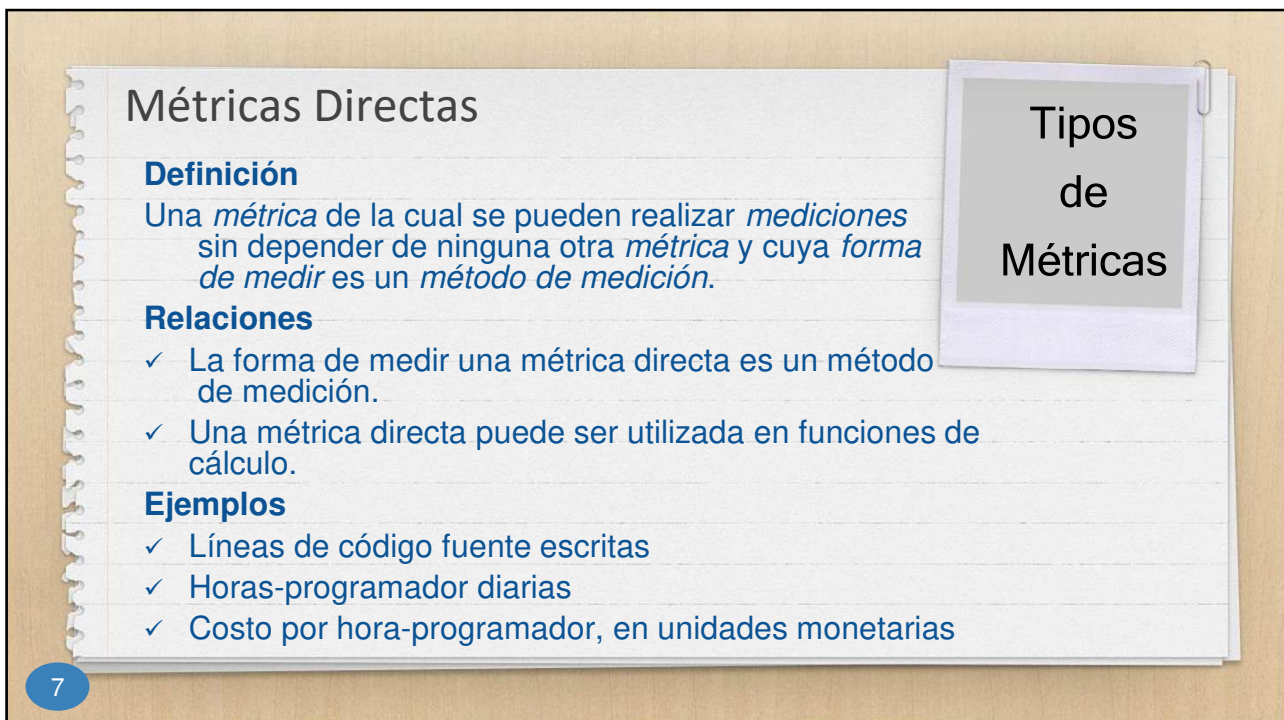
X **Métrica**
Es una medida del grado en que un sistema, componente o proceso posee un atributo dado

X **Indicador**
Es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software, o del producto en sí.

6

6

6



Métricas Directas

Definición
Una *métrica* de la cual se pueden realizar *mediciones* sin depender de ninguna otra *métrica* y cuya *forma de medir* es un *método de medición*.

Relaciones

- ✓ La forma de medir una métrica directa es un método de medición.
- ✓ Una métrica directa puede ser utilizada en funciones de cálculo.

Ejemplos

- ✓ Líneas de código fuente escritas
- ✓ Horas-programador diarias
- ✓ Costo por hora-programador, en unidades monetarias

7

7

7

Métricas Indirectas

Definición

Una *métrica* cuya *forma de medir* es una *función de cálculo*, es decir, las *mediciones* de dicha *métrica* utilizan las *medidas* obtenidas en *mediciones* de otras *métricas* *directas* o *indirectas*.

Relaciones

- ✓ La forma de medir una métrica indirecta es una función de cálculo.
- ✓ Una métrica indirecta puede usarse en una función de cálculo.

Ejemplos

- ✓ Líneas de código fuente por hora de programador
- ✓ Costo total actual del proyecto, en unidades monetarias
- ✓ Costo por línea de código fuente

Tipos de Métricas

8

8

Escala

Definición

Un conjunto de valores con propiedades definidas [ISO 14598-1]

Relaciones

Toda escala es de un cierto "*Tipo de Escala*".

Ejemplos

Los valores que puede tomar la métrica "lenguaje de Programación usado en un proyecto": Pascal, C, Java (Nominal).

El nivel de madurez CMM: 1, 2, 3, 4, 5 (Ordinal).

El tamaño de un código software expresado en líneas de código: Conjunto de los números naturales (Ratio).

La temperatura expresada en grados centígrados o grados Fahrenheit (Intervalo).

Ejemplos de tipo de Escala

Nominal, Ordinal, Intervalo, Ratio y Absoluta

9

9

Confiabilidad (Reliability) y Validez (Validity)

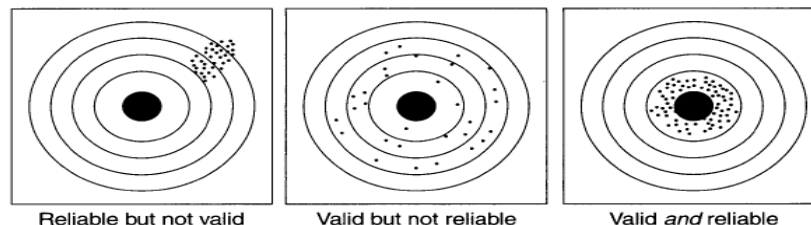
- x Muestran la calidad de la métrica
 - x **Validez** (*Grado en el cual la métrica refleja el significado real del concepto que se observa*)
 - x Contraparte de **exactitud**
 - x **Confiabilidad** (*métricas sin errores (error-free)*)
 - x Contraparte de **precisión**

10

10

Confiabilidad vs. Validez

- x Métricas confiables no son necesariamente válidas, y viceversa!
- x Una analogía:



11

11

¿Qué son las Métricas de Software?

“La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo de software y sus productos para suministrar información relevante a tiempo, así el líder de proyecto junto con el empleo de estas técnicas mejorará el proceso y sus productos” Michael [‘99]

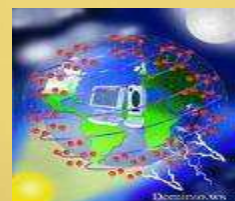


12

12

Métricas de software: DOMINIO

- **Métricas de proceso.**
- **Métricas de proyecto.**
- **Métricas de producto.**



13

13

13

Ejemplos:

Métricas de Proyecto

- X Costos estimados vs reales.
- X Esfuerzo por etapa.
- X Eficiencia de estimaciones del proyecto.
- X Desviación de calendario

Métricas de Producto

- X Líneas de Código
- X Defectos x severidad
- X Cantidad de Casos de uso x complejidad

Métricas de Proceso

- X Eficiencia en la remoción de defectos
- X Desviación Organizacional de Estimaciones
- X Defectos x severidad en los productos de la organización

14

14

Indicadores

Los indicadores de proceso son **ESTRATÉGICOS** y permiten:

- Tener una visión profunda de la eficacia de un proceso existente.
- Que los gestores evalúen lo que funciona y lo que no.

Los indicadores de proyecto son **TÁCTICOS** y permiten:

- Evaluar el estado del proyecto en curso
- Seguir la pista de los riesgos potenciales
- Detectar las áreas de problemas antes de que se conviertan en «críticas»
- Ajustar el flujo y las tareas del trabajo
- evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo de la ingeniería del software.



15

15

15

Métricas adecuadas al nivel adecuado....

X Desarrollador	X Organización	X Equipo de Desarrollo
1. Esfuerzo	1. Tiempo Calendario	1. Tamaño del producto
2. Esfuerzo y duración estimada y actual de una tarea.	2. Performance actual y planificada de esfuerzo.	2. Duración estimada vs. Actual
3. % de cobertura por el unit test	3. Performance actual y planificada de presupuesto	3. Niveles de staffing actuales y estimados.
4. Numero y tipo de defectos encontrados en el unit test.	4. Precisión de estimaciones en Schedule y esfuerzo	4. Tareas planificadas vs completadas.
5. Numero y tipo de defectos encontrados en revisión por pares.	5. Defectos en Release	5. Distribución del esfuerzo
		6. Status de requerimientos.
		7. Volatilidad de requerimientos.
		8. Defectos encontrados
		9. Status de distribución de defectos.
		10. % de test ejecutados

16

16

- X Las métricas no resuelven problemas... las personas los resuelven
- X Las métricas proveen información para que las personas puedan tomar mejores decisiones
- X Las métricas no son el problema... las métricas dan visibilidad a los problemas



17

17

Manténgalo Simple...

Si estás a millas de distancia de tu destino... no tiene sentido medir en milímetros.



Preguntas:

¿Nos da más información que la que tenemos ahora?

¿Es esta información de beneficio práctico?

¿Nos dice lo que queremos saber?

18

18

Métricas Básicas

📏 **Tamaño**

📏 **Esfuerzo**

📏 **Tiempo (Calendario)**

📏 **Defectos**



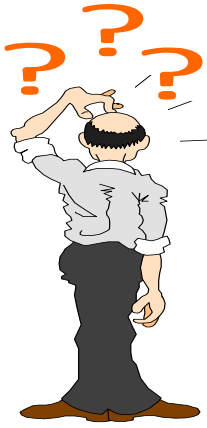
19

19

19

Escenario 1
Schedule: 100%
Effort: 75%
Features: 80%

Escenario 2
Schedule: 50%
Effort: 145%
Features: 50%



Análisis de Escenarios

Escenario 3
Schedule: 75%
Effort: 75%
Features: 75%

Escenario 4
Schedule: 100%
Effort: 100%
Features: 50%

20

Si...



Mediciones Basadas en objetivos *Basili et al.*
The GQM Approach
<http://www.cs.toronto.edu/~sme/CSC444F/handouts/GQM-paper.pdf>

Provea feedback



Tenga e identifique stakeholders





Foco en procesos, productos & servicios



Procesos, Productos & Servicios

21

21

21

No...

Mida individuos

Use métricas como un "castigo"

Use sólo una métrica

Ignore los datos

22

22

Métricas Ágiles

23

23

Regla de Oro Ágil sobre Métricas

x La medición es una salida, no una actividad

Una filosofía minimalista sobre las Métricas:

Medir la que sea necesario
y nada más.

24

24

Dos principios ágiles que guían la elección de las Métricas

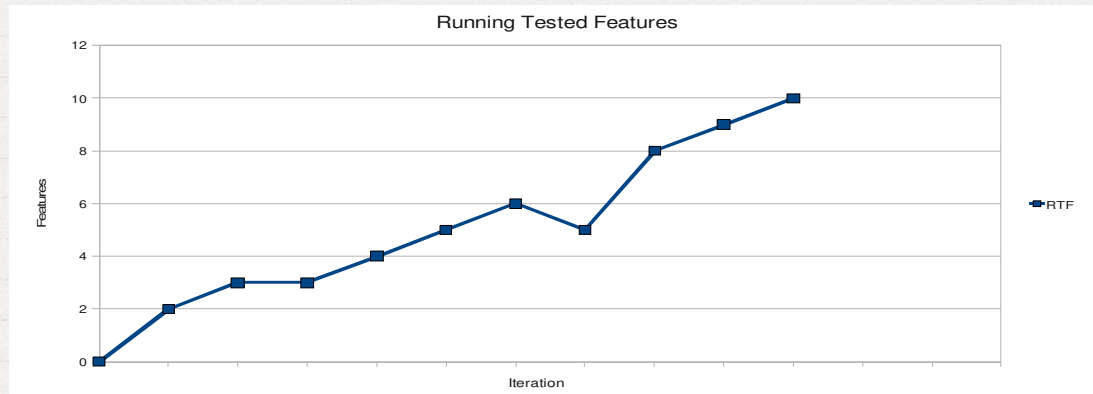
"Nuestra mayor prioridad es satisfacer al cliente
por medio de entregas tempranas y continuas de
software valioso."

"El Software trabajando es la principal medida de
progreso."

25

25

Running Tested Features (RTF)



26

26

Capacidad

Capacidad es...

Una estimación de cuánto trabajo puede completarse en un período de tiempo dado.

Basado en la cantidad de tiempo ideal disponible del equipo.

...Se puede medir en...

- Esfuerzo (horas)
- Puntos de Historia

27

27

Capacidad

× **Horas de Trabajo Disponibles por día (WH) X Días Disponibles Iteración (DA) = Capacidad**

$$WH \times DA = \text{Capacidad}$$

× **Ejemplo:**

- × Equipo de 8 miembros
- × 4 miembros disponibles los 2 primeros sprints.
- × 1 miembro se casa en sprints 5 y 6
- × 6 horas de trabajo

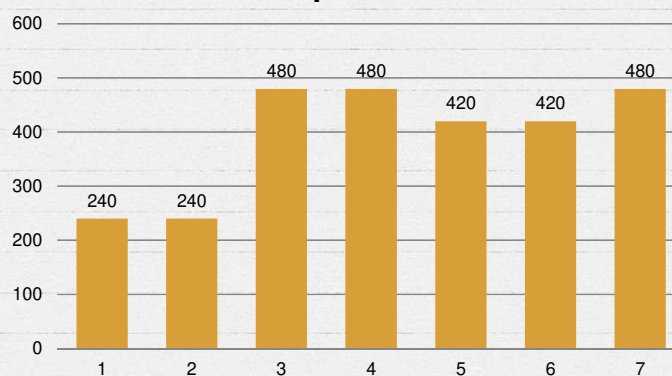
28

28

Capacidad

Sprint	1	2	3	4	5	6	7	Total
Horas	240	240	480	480	420	420	480	2760
Puntos de Historia	30	30	45	60	58	52	60	335

Capacidad



29

29

Velocidad

Velocidad es...

Una observación empírica de la capacidad del equipo para completar el trabajo por iteración.

...y no es...

- una estimación
- un objetivo a alcanzar

30

30

Velocidad

Velocidad está...

Basada en el tamaño de los ítems de trabajo propio de cada equipo.

...y no está...

- basada en estimaciones o tiempo real
- determinado o impuesto por nadie más que por los miembros del equipo.

31

31

Velocidad

Velocidad es...

Comparable entre iteraciones para un equipo dado y un proyecto dado.

...y no es...

- comparable entre equipos
- comparable entre proyectos

32

32

Unidad de medida de la Velocidad

Cómo planea el equipo	Unidad de Medida
Compromiso con las historias	Historia
Tamaño relativo (puntos)	Puntos
Estimación (horas ideales)	Horas ideales

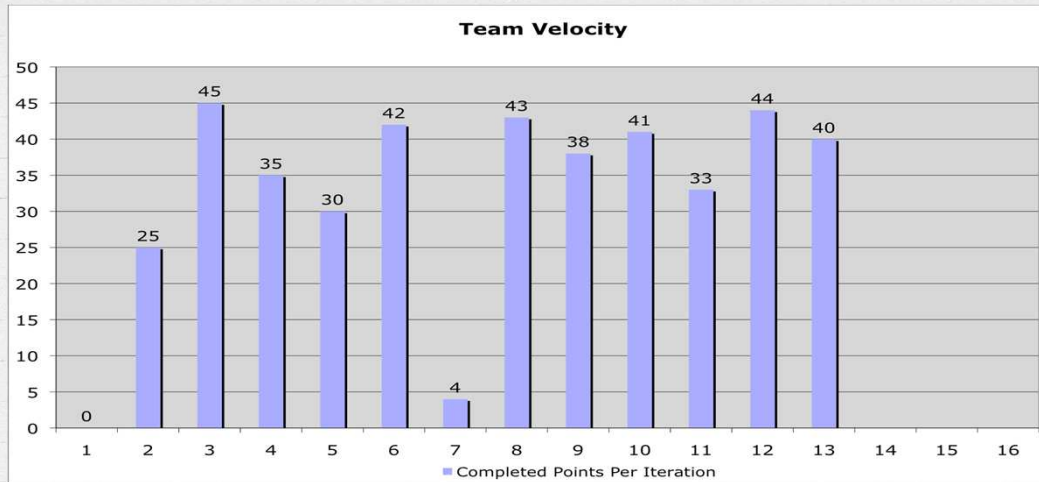
¿Qué cuenta para la velocidad?

**Solo cuenta el trabajo completado
para la velocidad**

33

33

Velocidad



34

34

Mediciones Problemáticas

No relevantes con los métodos ágiles:

- ✓ Diagrama de Gantt
- ✓ Porcentaje Completado
- ✓ Tiempo de cada miembro del equipo por tarea
- ✓ Tiempo Real vs. Tiempo Estimado



35

35

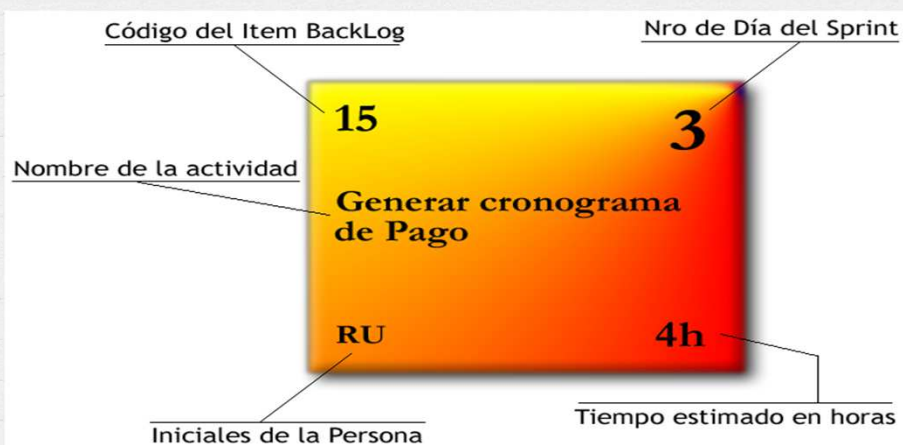
Herramientas de Scrum



36

36

Tarjeta de Tarea (Task)



37

37

Taskboard (Tablero)

Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... SC 8
	Code the... 2	Code the... 8	Test the... SC 8		Test the... SC 8
	Test the... 8	Test the... 4			Test the... SC 6
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... SC 8
	Code the... 4	Code the... 6			Test the... SC 6

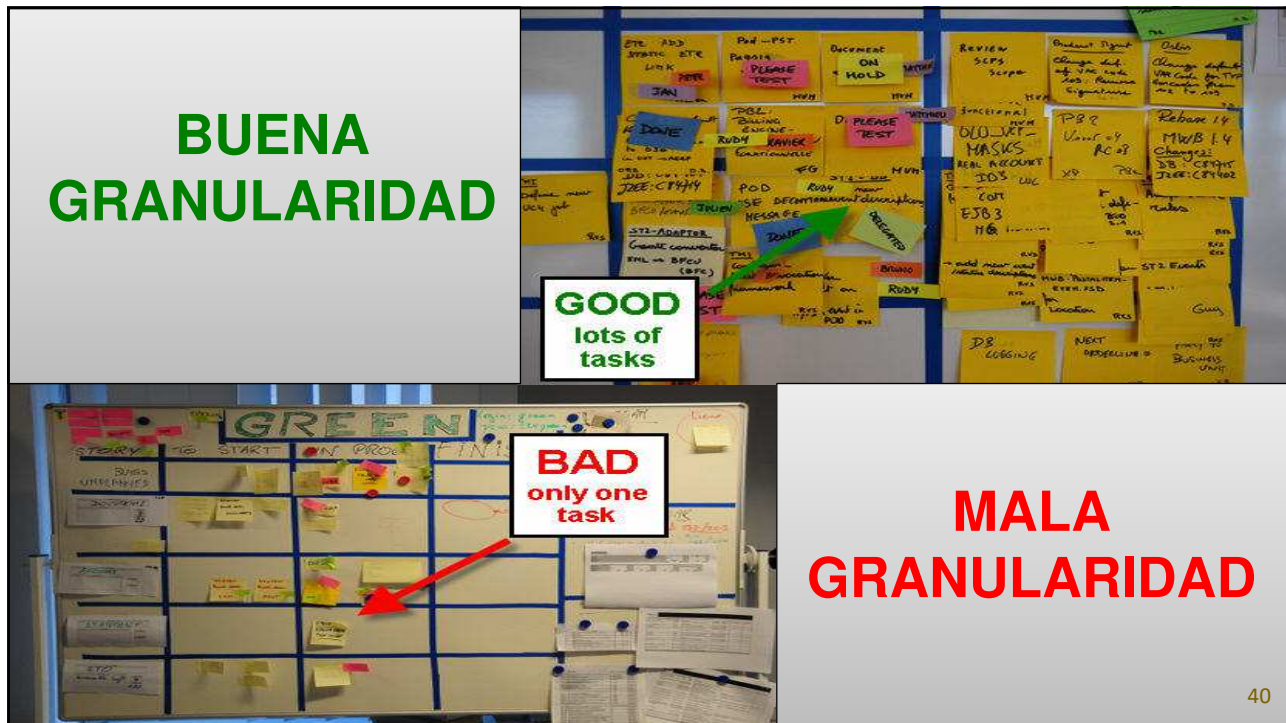
38

38

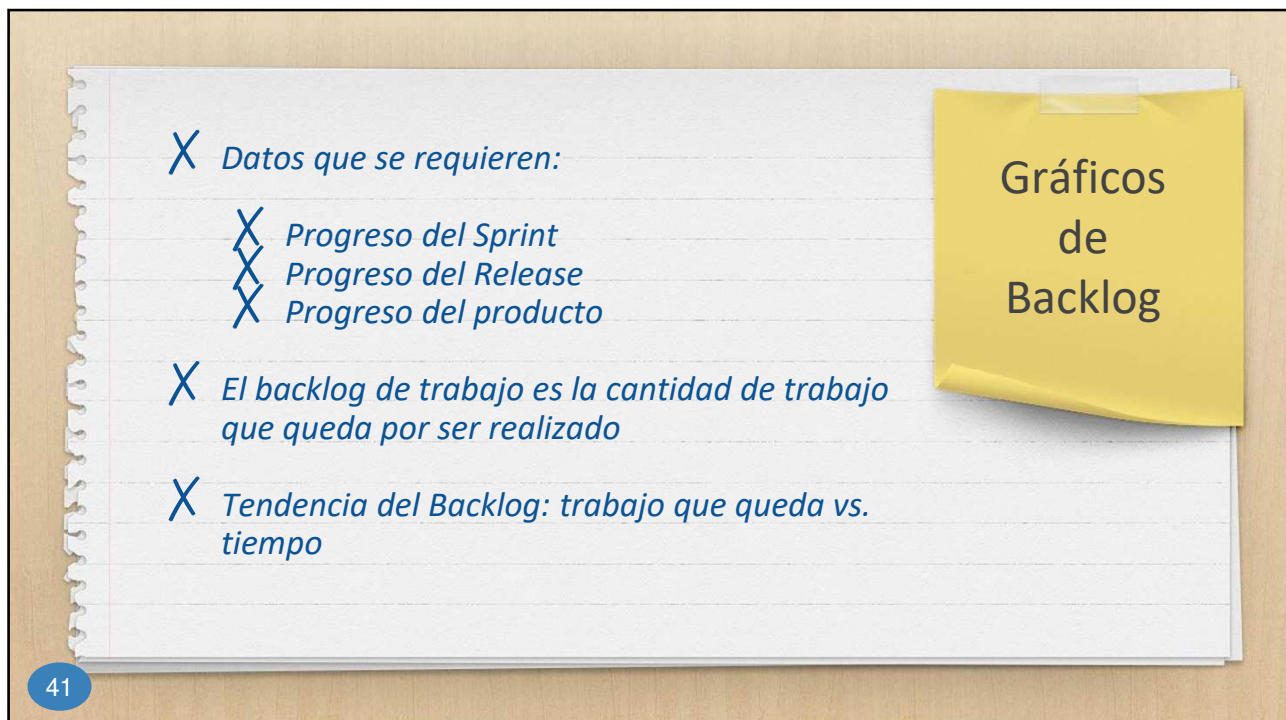


39

39



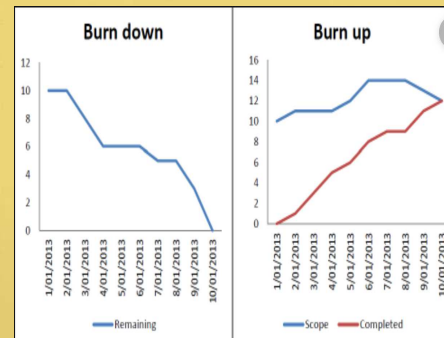
40



41

Poniendo la velocidad a trabajar...

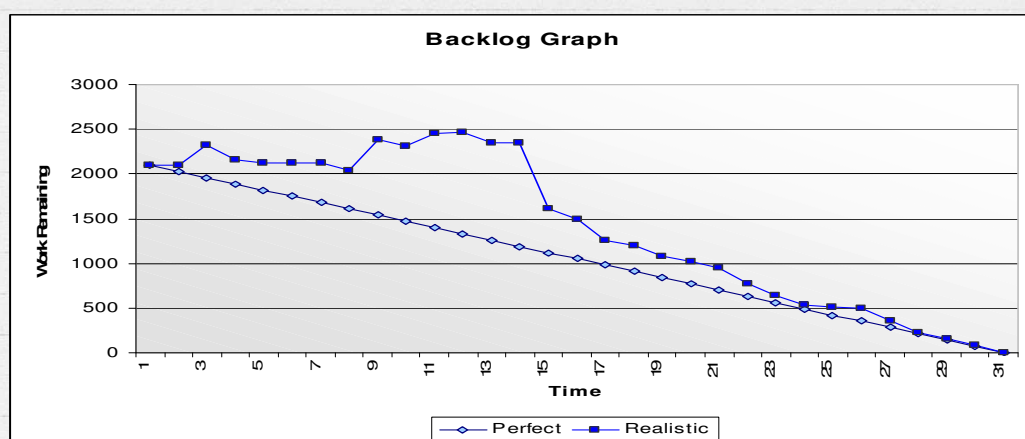
- **Burn-down chart**
¿Cuánto trabajo queda para terminar?
- **Burn-up chart**
¿Cuánto trabajo se ha completado?
- **Combined burn chart**
¿Cuánto trabajo se ha completado y cuánto trabajo queda?



42

42

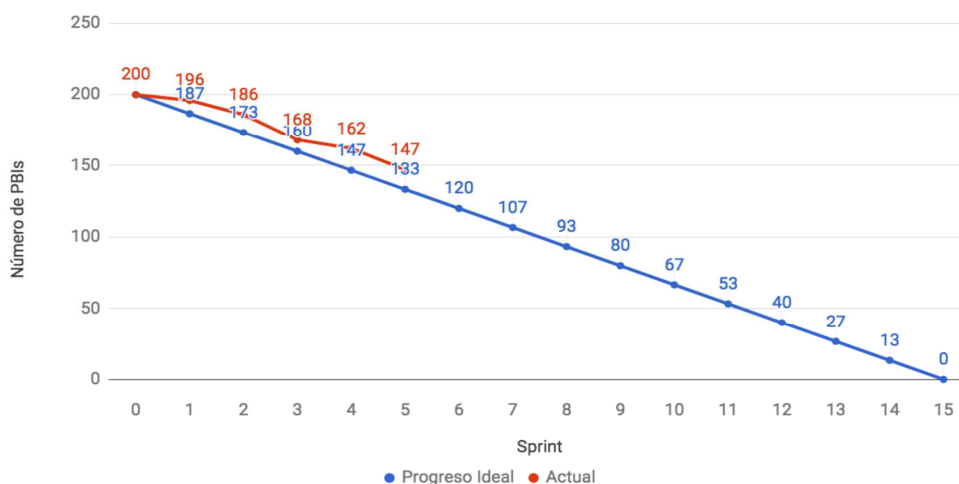
Burndown chart



43

43

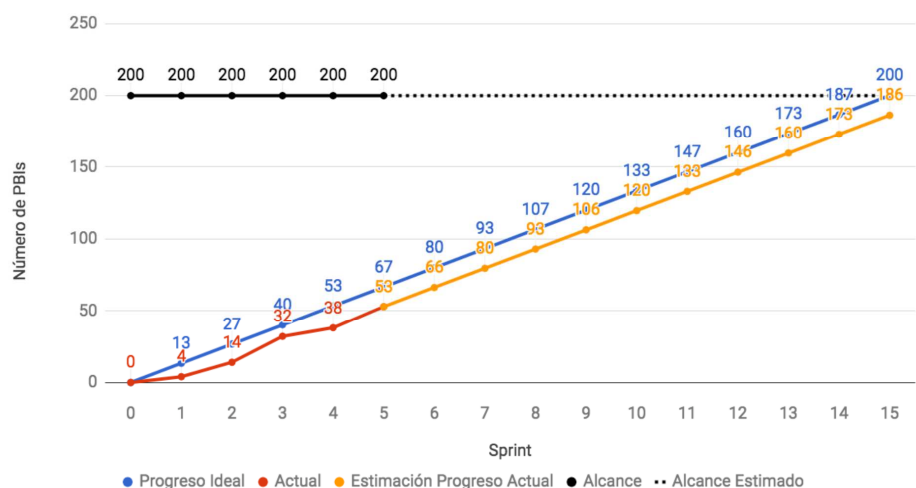
Burndown Chart – Line Style



44

44

Burnup Chart

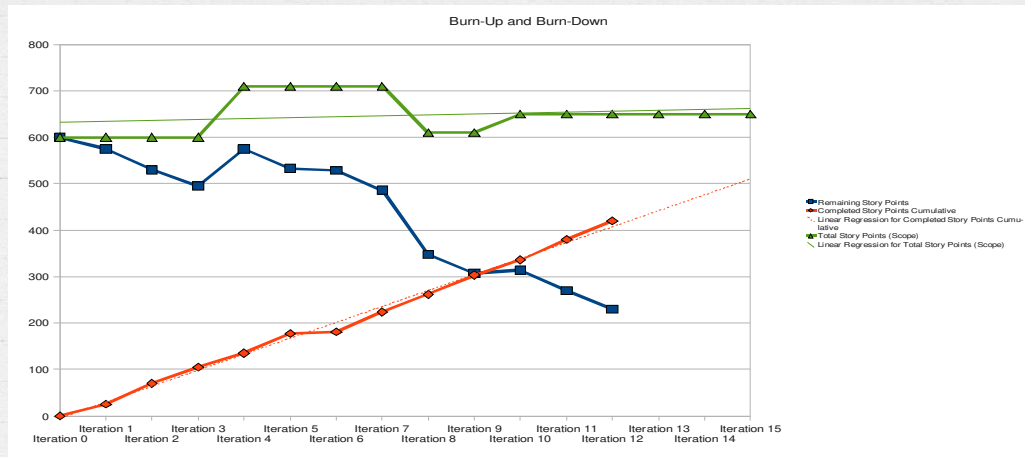


45

45

45

Burnup and Burndown Chart

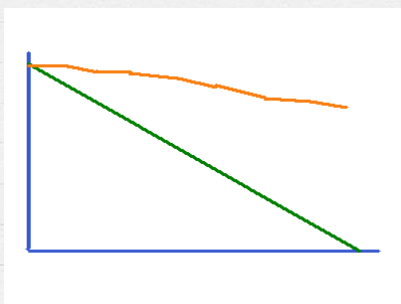


46

46

¿Diagnóstico?

BURNDOWN Chart

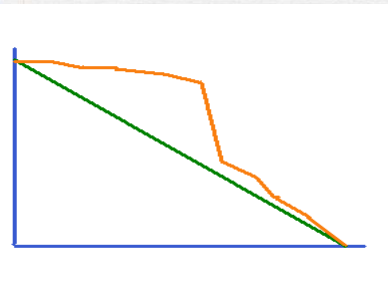


- No todo lo incluido en el sprint será entregado en la fecha propuesta. El backlog no fue modificado para alcanzar la fecha (remover requerimientos o modificar la fecha)
- Posibles causas:
 - Pobres técnicas de estimación
 - Mal manejo de riesgos
 - Rotación del equipo o renuncias
- Este gráfico muestra que el Scrum Master (SM) y el Product Owner (PO) no reaccionaron para evitar esta situación.
- Evite llegar a este estadio!

47

47

Burndown charts



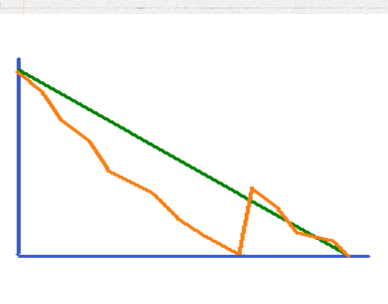
¿Diagnóstico?

- Este BC es mucho mejor que el anterior
- Al comienzo del sprint hubo problemas, pero en el medio el equipo de trabajo reaccionó y tomó medidas correctivas. Por la línea se nota que el PO quitó requerimientos para mantener la fecha de release.
- También pone en evidencia un mejor manejo de riesgos lo que facilitó aumentar la velocidad en la segunda parte del sprint.

48

48

Burndown charts



¿Diagnóstico?

- Este BC también tiene características negativas.
- Este no es un ejemplo de desarrolladores con las mejores habilidades técnicas. Demuestra pobres técnicas de estimación.
- El PO y el SM deberían haber agregado más requerimientos para ese release.

49

49

Burndown charts

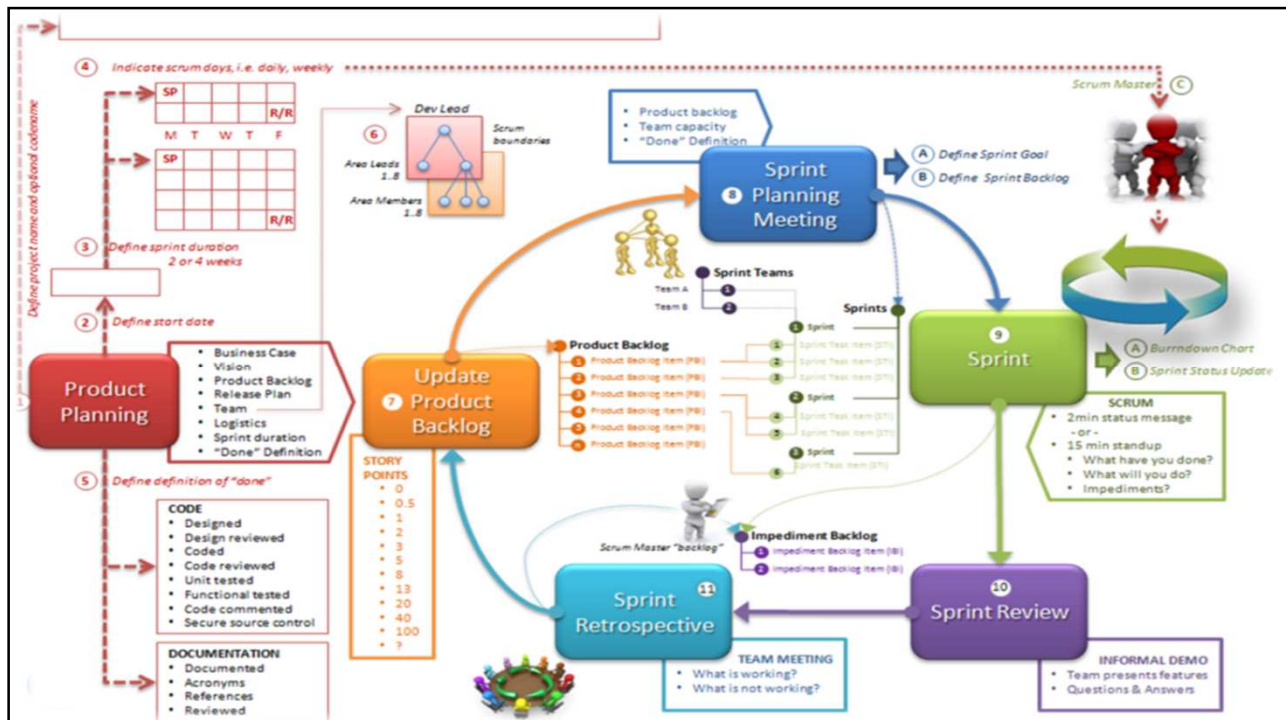


Diagnóstico?

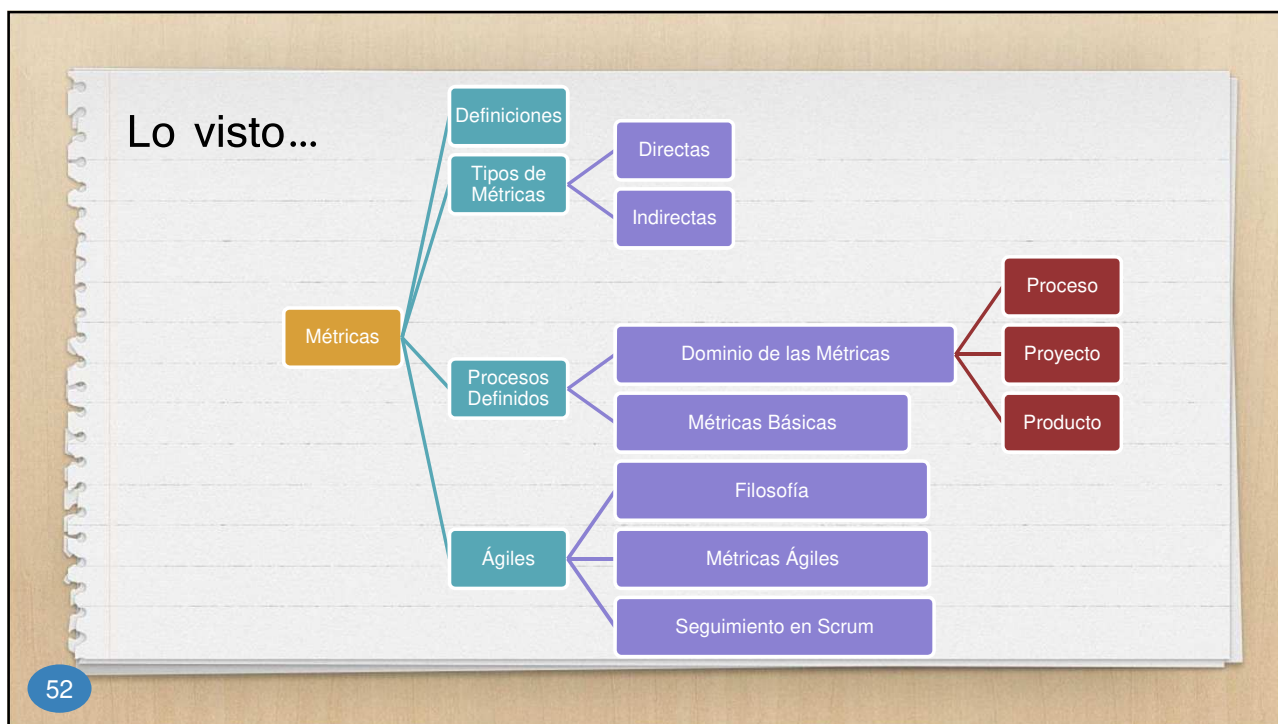
- Ideal
- Buenas técnicas de estimación.
- Buena velocidad de desarrollo
- Buen manejo de impedimentos
- Todos los requerimientos prometidos son los entregados.
- Buen manejo de la asignación de tareas (tareas cortas, reporte diario)
- Quizás.....demasiado bueno para ser verdad 😊

50

50



51



52