


---

**Universidad Tecnológica Nacional**  
**Facultad Regional Córdoba**  
**Cátedra de Ingeniería y Calidad de Software**  
**Docentes: Judith Meles – Laura Covaro**

# MÉTRICAS DE SOFTWARE EN LOS DIFERENTES ENFOQUES DE GESTIÓN



# MÉTRICAS DE SOFTWARE EN EL ENFOQUE TRADICIONAL

BASADO EN PROCESOS  
DEFINIDOS

# Métricas de software

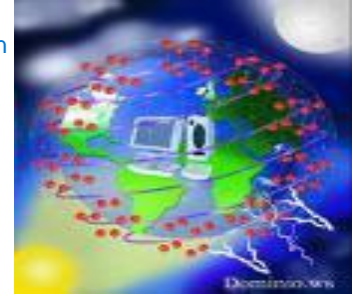
**El dominio de las métricas del software se divide en:**

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

¿Qué es una métrica? ¿Para qué sirve?

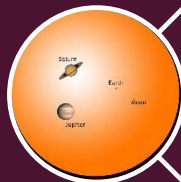
Una métrica es un valor cuantitativo que nos permite dar información sobre un elemento de interés que nos interese medir, con un resultado objetivo

Es la cuantificación del grado de un atributo que quiero medir.



**Las métricas del proyecto se consolidan para crear métricas de proceso que sean públicas para toda la organización del software.**

Son buscadas las métricas de las 3 dimensiones



## Tamaño del producto



## Esfuerzo

Mide el trabajo que hay que hacer (talento humano) en horas personas lineales ("bolsa de horas")



## Tiempo (Calendario)



## Defectos

# Métricas básicas para un proyecto de software

Como para empezar a medir algo en el proyecto. Hay que armarlas a las métricas una vez definido qué hay que medir.

Métricas de proceso: son a nivel organizacional, métricas públicas. El foco es la mejora, su impacto es grande. Son despersonalizadas, no hablan de ningún proyecto ni producto en particular, sino del comportamiento organizacional.

Métricas de proyecto: son privadas, miden el desempeño de un sólo proyecto.

Métricas de producto: son privadas, miden el desempeño de un sólo producto. Ej: User stories a realizar, líneas de código hechas, etc.

# EL SUEÑO DEL PIBE...

- **Desarrollador**

1. Esfuerzo
2. Esfuerzo y duración estimada y actual de una tarea.
3. % de cobertura por el unit test
4. Numero y tipo de defectos encontrados en el unit test.
5. Numero y tipo de defectos encontrados en revisión por pares.

- **Organización**

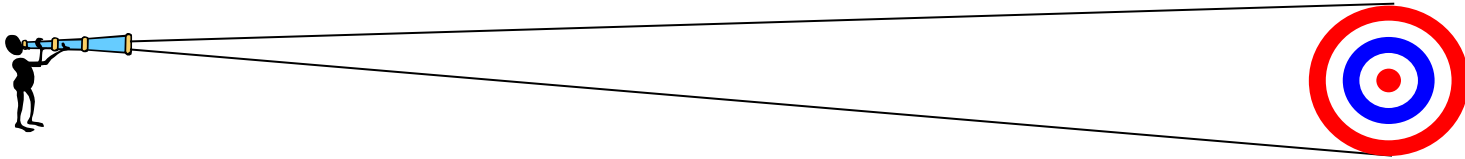
1. Tiempo Calendario
2. Performance actual y planificada de esfuerzo.
3. Performance actual y planificada de presupuesto
4. Precisión de estimaciones en Schedule y esfuerzo
5. Defectos en Release

- **Equipo de Desarrollo**

1. Tamaño del producto
2. Duración estimada y actual entre los hitos más importantes.
3. Niveles de staffing actuales y estimados.
4. Nro. de tareas planificadas y completadas.
5. Distribución del esfuerzo
6. Status de requerimientos.
7. Volatilidad de requerimientos.
8. Nro. de defectos encontrados en la integración y prueba de sistemas.
9. Nro. de defectos encontrados en peer reviews.
10. Status de distribución de defectos.
11. % de test ejecutados

# MANTÉNGALO SIMPLE...

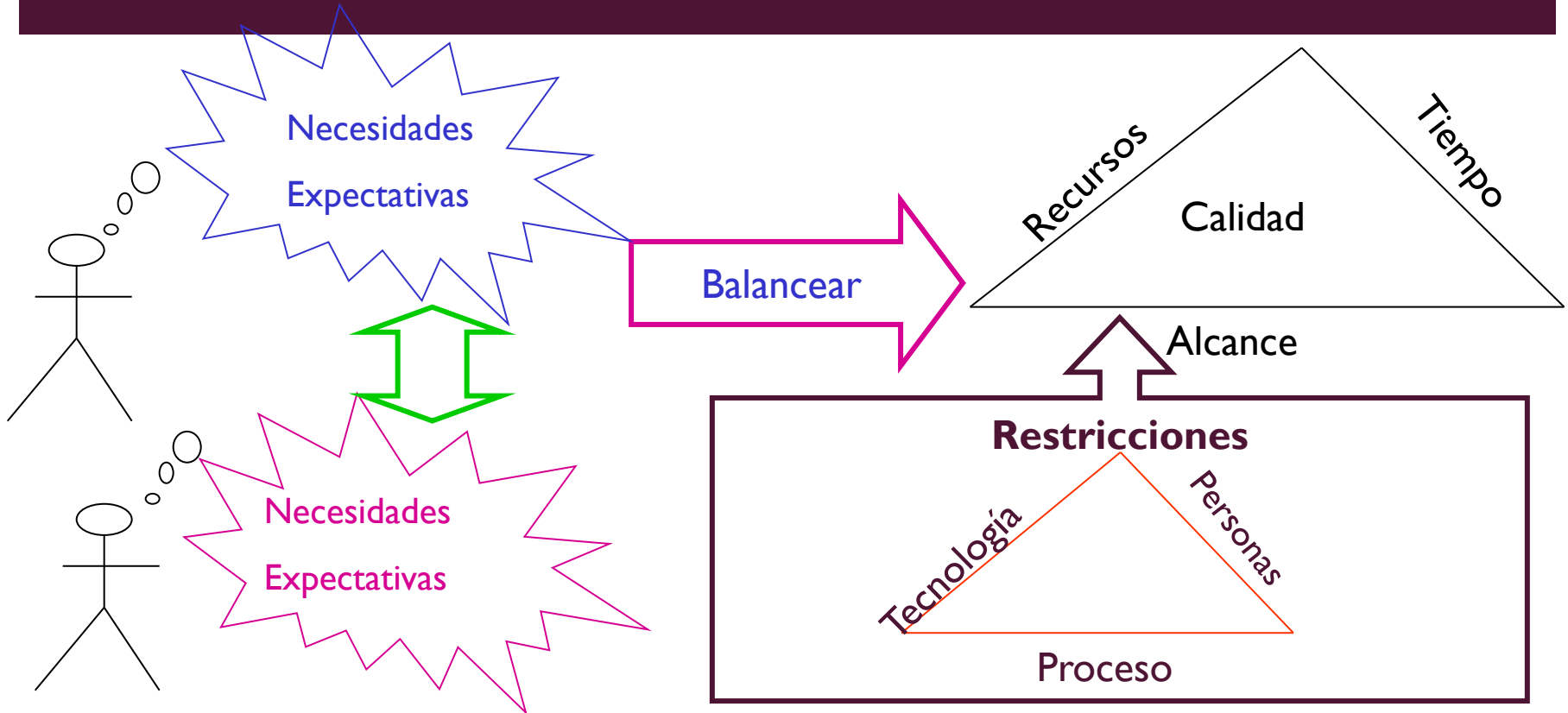
Si estás a millas de distancia de tu destino... no tiene sentido medir en milímetros.



## Preguntas:

- ¿Nos da más información que la que tenemos ahora?
- ¿Es esta información de beneficio práctico?
- ¿Nos dice lo que queremos saber?

# ALGO MÁS SOBRE LAS MÉTRICAS...



# MÉTRICAS DE SOFTWARE EN AMBIENTES ÁGILES

PARA PROCESOS EMPÍRICOS





# REGLA DE ORO ÁGIL SOBRE MÉTRICAS

- La medición es una salida, no una actividad

Una filosofía minimalista sobre las Métricas:

Medir la que sea necesario  
*y nada más.*

## DOS PRINCIPIOS ÁGILES QUE GUÍAN LA ELECCIÓN DE LAS MÉTRICAS

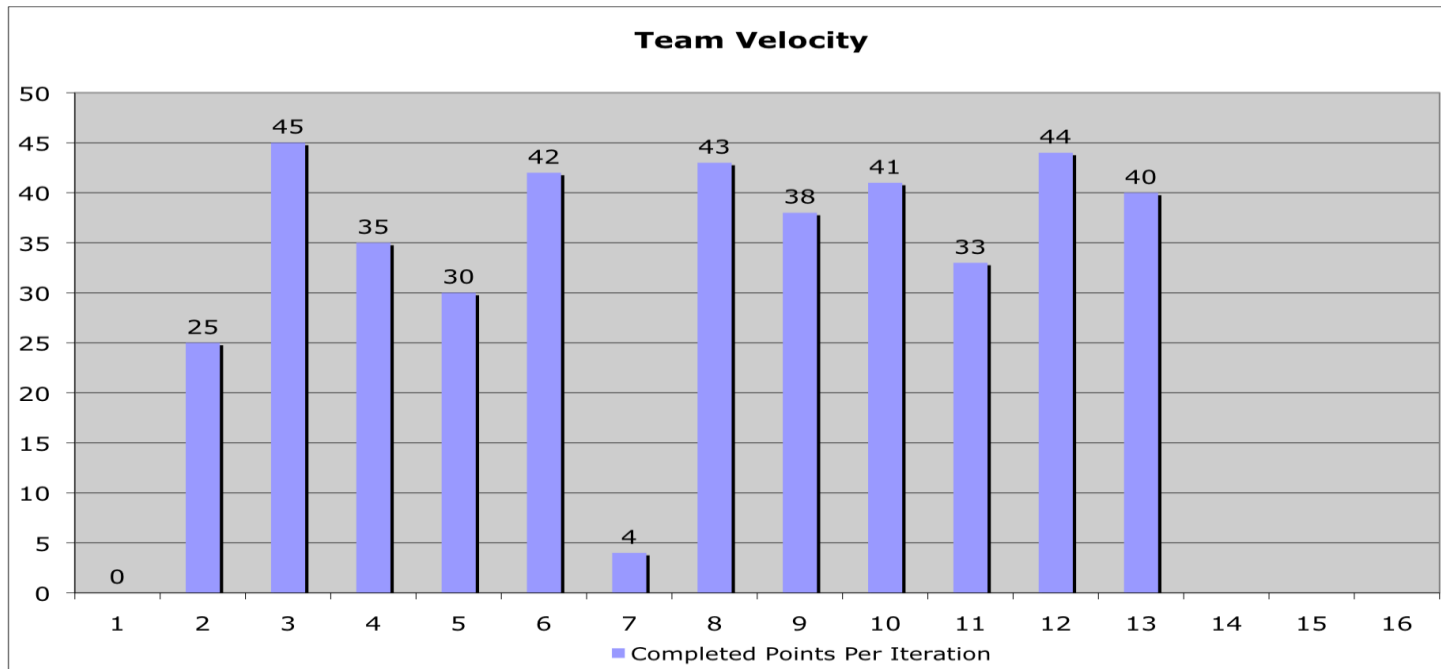
“Nuestra mayor prioridad es satisfacer al cliente por medio de entregas tempranas y continuas de software valioso, funcionando.”

“El Software funcionando es la principal medida de progreso.”

# VELOCIDAD

Medida de producto

Es "LA" métrica del agilismo. Mide la cantidad de story points que son aceptados por el PO al final de un sprint.  
Se calcula al final del sprint. Se calcula, no se estima

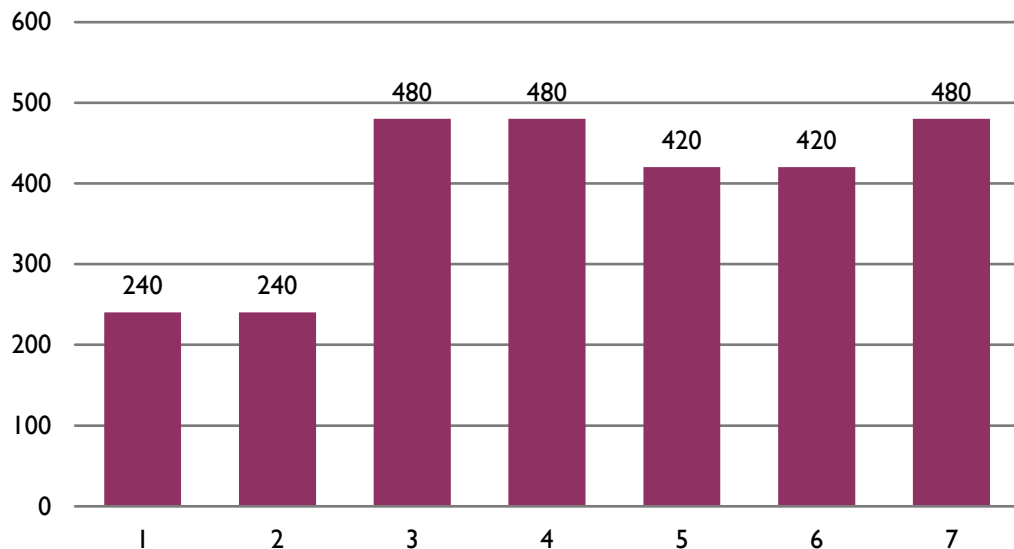


# CAPACIDAD

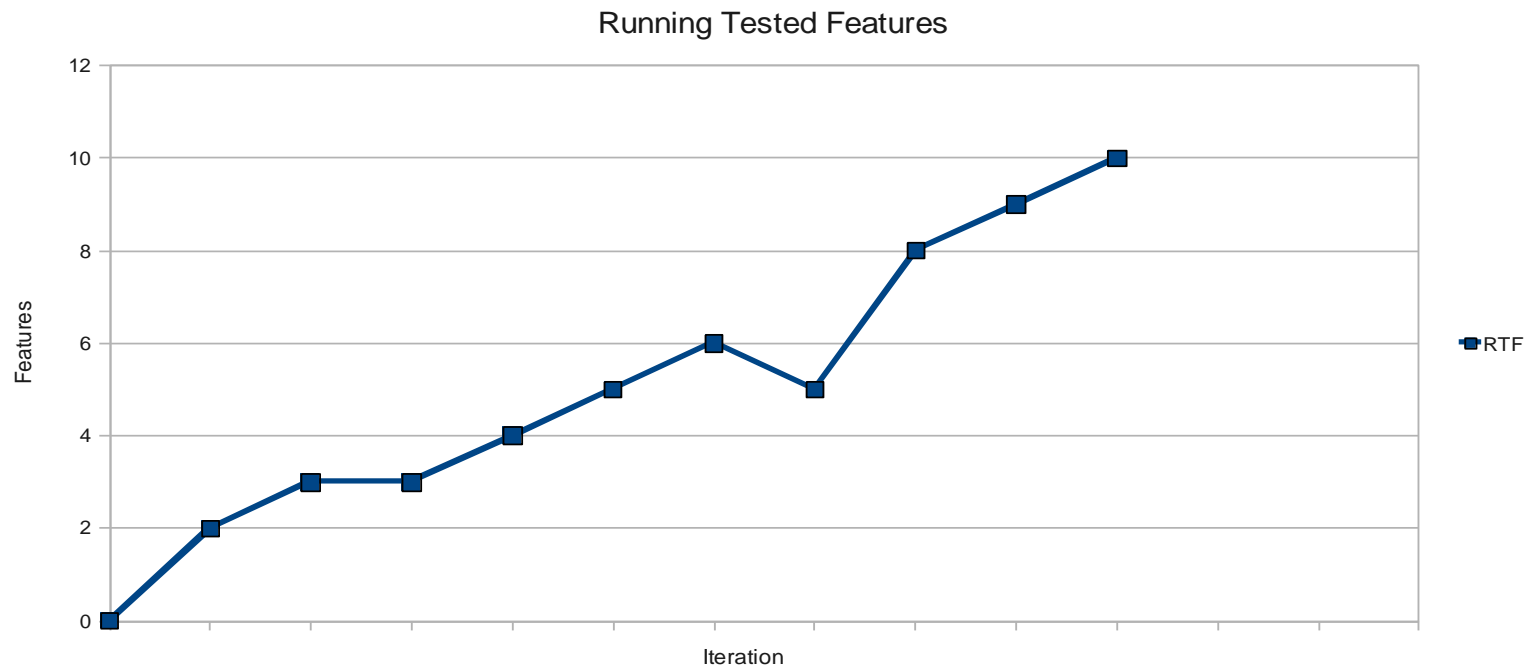
Medida de proyecto  
Se estima para saber cuánto trabajo se puede hacer por sprint, es un compromiso. Se calcula en el Sprint Planning.

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

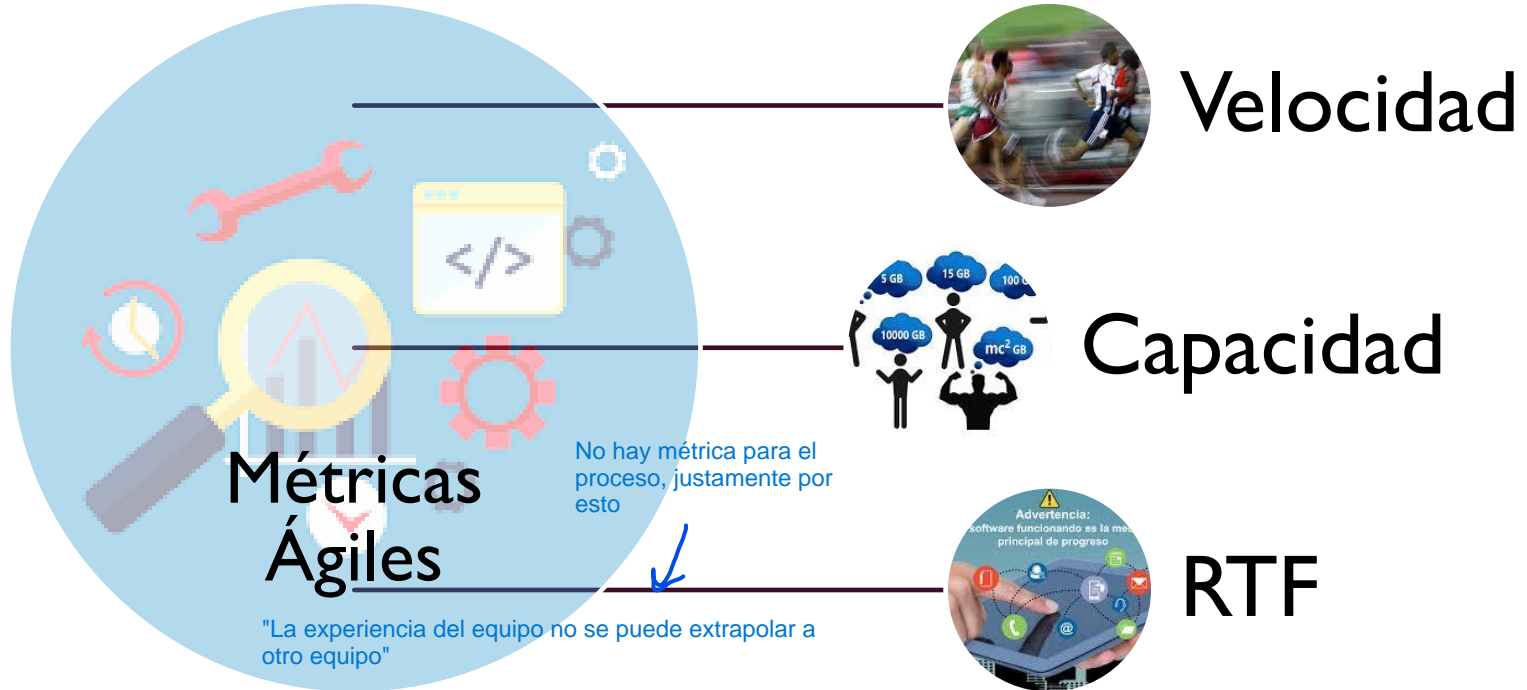
## Capacidad



# RUNNING TESTED FEATURES (RTF)

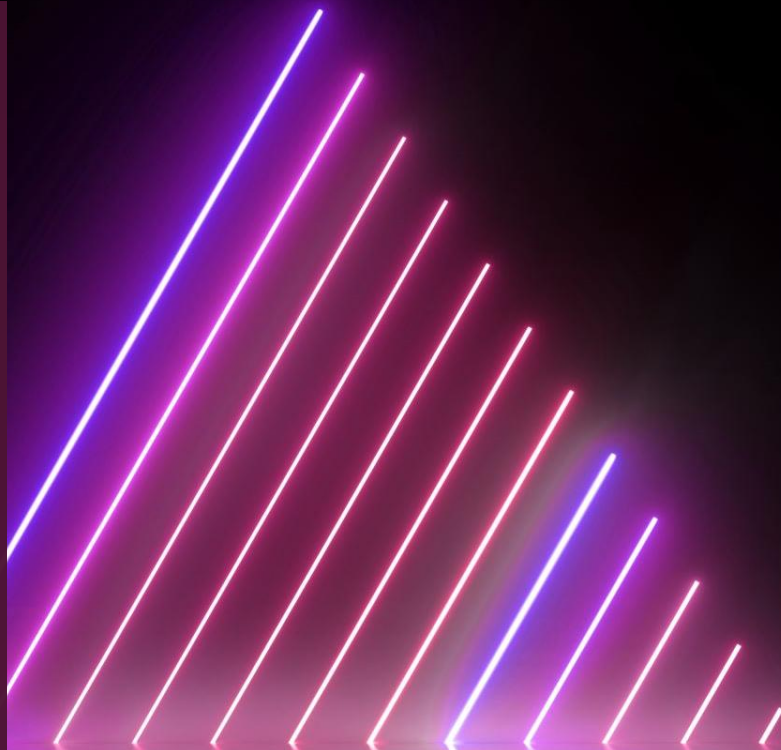


# RESUMIENDO...



# MÉTRICAS EN KANBAN

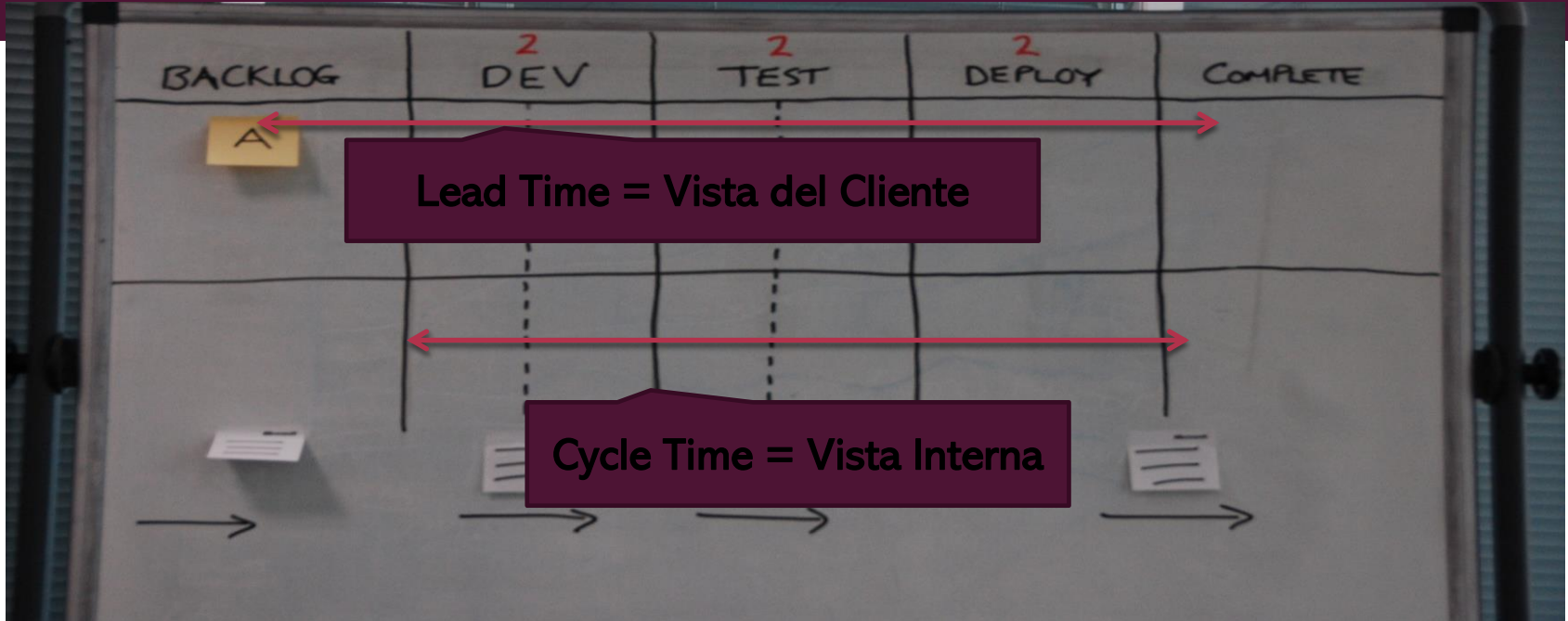
PARA PROCESOS EMPÍRICOS CON  
ENFOQUE LEAN



# KANBAN: Métricas Clave

Framework de mejora de procesos

Todas las métricas miden proceso





# KANBAN: MÉTRICAS CLAVE

## Cycle Time (Tiempo de ciclo)

- Es la métrica que registra el tiempo que sucede entre el inicio y el final del proceso, para un ítem de trabajo dado. Se suele medir en días de trabajo o esfuerzo.
- Medición más mecánica de la capacidad del proceso
- **Ritmo de Terminación**

## Lead Time (Tiempo de entrega)

- Es la métrica que registra el tiempo que sucede entre el momento en el cual se está pidiendo un ítem de trabajo y el momento de su entrega (el final del proceso). Se suele medir en días de trabajo.
- **Ritmo de entrega**

# KANBAN: MÉTRICAS CLAVE

## Touch Time (Tiempo de Tocado)

- El tiempo en el cual un ítem de trabajo fue realmente trabajado (o "tocado") por el equipo.
- Cuántos días hábiles pasó este ítem en columnas de "trabajo en curso", en oposición con columnas de cola / buffer y estado bloqueado o sin trabajo del equipo sobre el mismo.

$$\textit{Touch Time} \leq \textit{Cycle Time} \leq \textit{Lead Time}$$

## Eficiencia del Ciclo de Proceso

$$\% \text{ Eficiencia ciclo proceso} = \text{Touch Time} / \text{Elapsed Time.}$$

Lo ideal es 1. Significa que no hay cola en el proceso.

# RESUMIENDO MÉTRICAS EN CADA ENFOQUE

## ■ Tradicionales

- Esfuerzo
- Tiempo
- Costos
- Riesgos

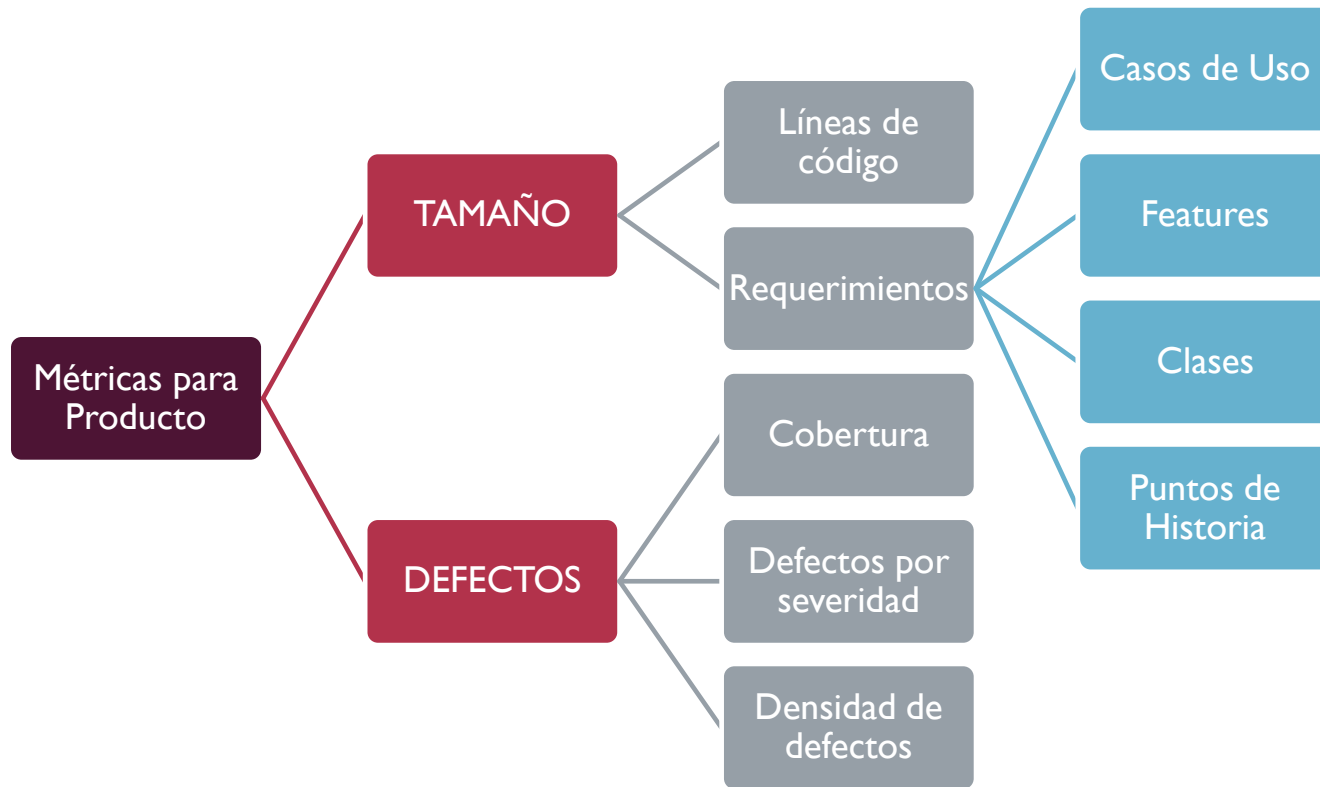
## ■ Ágiles

- Velocidad
- Capacidad
- Running Tested Features

## ■ Lean

- Lead Time
- Cycle Time
- Touch Time
- Eficiencia Proceso

# ¿Y PARA EL PRODUCTO SOFTWARE, QUÉ MÉTRICAS ?



## PARA PENSAR...

- Compara herramientas para entender no para juzgar.
- ¿Cuál es mejor?





GRACIAS POR  
SU ATENCIÓN

DUDAS, PREGUNTAS,  
CONSULTAS?????