## Métricas - Síntesis (Ideas principales)

### Métricas de proceso (Públicas)

Las métricas de proceso se utilizan con propósitos estratégicos y tienen como objetivo generar indicadores que permitan mejorar los procesos de software a largo plazo.

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

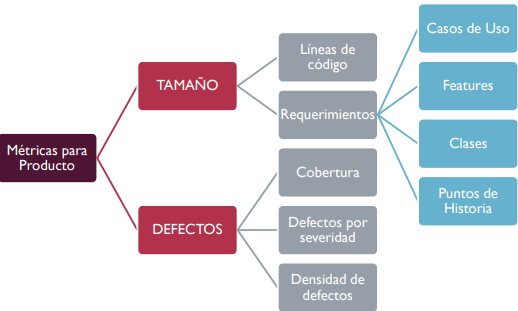
Tengo métricas de proceso despersonalizando métricas de producto y de proyecto.

Métrica: defectos por severidad que tiene la organización en sus productos (métrica de proceso)

### Métricas de producto

Las métricas de producto se utilizan con propósitos técnicos, tienen como objetivo generar indicadores en tiempo real de la eficacia del análisis, el diseño, la estructura del código, la efectividad de los casos de prueba y calidad global del software a construir.

Métrica: defectos por severidad en esta versión de mi producto de software (métrica de producto)



### Métricas de proyecto (Privadas)

Las métricas de proyecto se utilizan con propósitos tácticos, es decir, tiene como objetivo generar información que sirva como base a líderes de proyectos y al equipo para adaptar el desarrollo de los proyectos y de las actividades técnicas.

Se vincula con la triple restricción, mido cosas relacionadas al tiempo, recursos y alcance.

Ejemplo: esfuerzo, esfuerzo por cada persona, tiempo que llevo cada iteración, etc.



**MÉTRICAS DE SOFTWARE EN EL ENFOQUE TRADICIONAL**

* **Tamaño del producto. PRODUCTO**

Una cosa es el tamaño del producto, y otra cosa es el esfuerzo que yo voy a necesitar para construir ese producto de ese tamaño.

Se utiliza para estimar el esfuerzo requerido en el desarrollo, el mantenimiento y la prueba del software.

Usualmente en este caso se estima la cantidad de líneas de código. Esto es ineficiente, por lo que actualmente se miden funcionalidades, requerimientos, alcances, casos de uso, etc.

Responde al que.

* **Defectos. PRODUCTO**

Medir sobre el producto las diferencias con lo que se espera. Es necesario construirse tomando en cuenta la gravedad del defecto.

Los defectos del software son errores, fallas o problemas encontrados en el software durante el desarrollo, la prueba o el uso.

* **Esfuerzo. PROYECTO**

Se mide en horas persona lineales. Asumir que es una persona haciendo las cosas una por vez, no se tiene en cuenta el trabajo paralelo, el solapamiento, la cantidad de gente que va a trabajar. no tengo en cuenta el tiempo de descanso.

El esfuerzo del software representa la cantidad de trabajo humano necesario para desarrollar o mantener un sistema de software.

bolsa de horas donde no tengo en cuenta que puedo hacer más de una cosa a la vez.

Responde al cómo.

* **Tiempo (Calendario). PROYECTO**

Del esfuerzo puedo crear el calendario. El tiempo se mide en semanas, meses. El tiempo responde al cuándo.

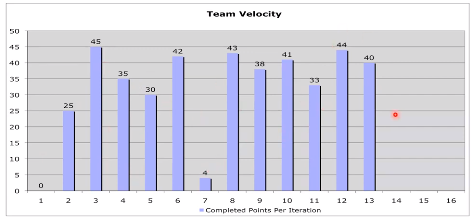
Esta métrica se refiere al tiempo requerido para completar una tarea o proyecto de desarrollo de software. Se utiliza para establecer plazos y programar actividades, y puede estar relacionada con el esfuerzo del software.

**LAS TRADICIONALES MIDEN PROCESO, PRODUCTO, PROYECTO.**

**MÉTRICAS DE SOFTWARE EN AMBIENTES ÁGILES - SCRUM PARA GESTION DE DESARROLLO DE PRODUCTO**

Regla de Oro Ágil sobre Métricas En enfoques ágiles nuestra mayor prioridad es satisfacer al cliente por medio de entregas tempranas y continuas de software valioso, es decir el software funcionando es la principal medida de progreso, por lo cual no se debe tomar a las métricas como una actividad, sino como lo que son, una salida. Esto quiere decir que se debe medir lo que sea necesario y nada más, es decir lo que agregue valor para el cliente.

* **Velocidad - se mide en puntos de historia -**

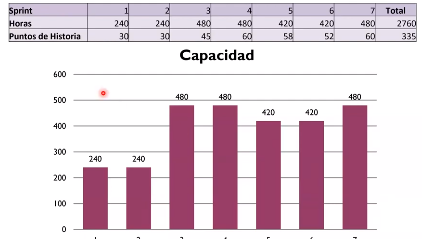
La velocidad es una observación empírica de la **capacidad del equipo para completar el trabajo por iteración,** comprobable entre iteraciones de un equipo dado.

Es importante aclarar, que la velocidad no se estima, se **calcula**.".

“cuántos punto de historia puedo entregarle al product owner al final de una iteración y él nos aceptó”

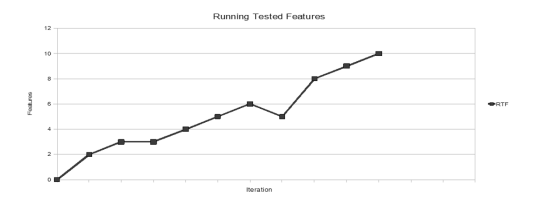
En definitiva, la velocidad está determinada por la cantidad de Storie Points que el Product Owner acepto en un sprint, y se toma en la Sprint Review.

* **Capacidad**.

Es una **estimación** de **cuánto trabajo puede completarse en un período de tiempo basado en la cantidad de tiempo ideal disponible del equipo**. Es teórica, y resulta de gran utilidad al momento de estimar, derivando de esta la velocidad.

Como se puede medir: Esfuerzo (horas hombre) o Puntos de Historia (Story Points)

* **RTF - running tested features**

El software funcionando es la mejor medida de progreso y es una medida directa de los resultados entregados.

cantidad de características de software que están en producción, que han sido testeadas y que están funcionando. Cuento cuantas características entregue por cada sprint. En lugar de user, como en velocidad, cuento features. “Tantas features en nro° de iteración”

**EL FOCO EN ÁGIL ES MEDIR PRODUCTO. Como la experiencia no se puede extrapolar no tiene mucho sentido medir proceso, no dice mucho. Por eso miden a nivel proyecto y producto.**

**MÉTRICAS EN KANBAN**

**Esto se mide para cada pieza de trabajo que pasa por el tablero. Después eso se puede consolidar haciendo promedios.**

* **Cycle Time (Tiempo de ciclo) - Vista interna.**

Ritmo de terminación

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

* **Lead Time (Tiempo de entrega) - Vista del cliente.**

Ritmo 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). (Desde que el cliente me pide algo, hasta que yo se lo entregó) Se suele medir en días de trabajo.

* **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.



* **Eficiencia del Ciclo de Proceso**

**% Eficiencia ciclo proceso = Touch Time / Elapsed Time.**