

Clase 24/10 Teórico

Métricas de software en los diferentes enfoques de gestión

Introducción

Es un número o un valor que se obtiene como resultado de un proceso de medición. Permite construir indicadores que facilitan la toma de decisiones.

Métrica → la presencia o el grado de valor sobre un atributo que yo quiero medir.

Es objetiva, por lo tanto debe ser cuantitativa.

Se debe poder medir.

Métricas de software

El dominio de las métricas del software se divide en:

- Métricas de proceso
- Métricas de proyecto
- Métricas de producto

Las personas o las personas no se miden. Esto hace que las personas mienten y se pierda la transparencia.

Métrica de proceso

Las métricas de proceso se obtienen o se consolidan a partir de las métricas de proyecto.

Métricas de proceso son públicas para toda la organización y hablan de comportamiento organizacional. Se utilizan fundamentalmente para mejorar los procesos.

Las métricas de proceso se encuentran despersonalizadas.

Por ej: eficiencia en la remoción de defectos (mejor cuando lo encontramos lo antes posible, peor es cuando lo encuentra el cliente) se debe tener en cuenta varios proyectos y productos diferentes y se debe despersonalizar las métricas.

Métricas de proyecto

Métricas de proyecto son privadas, porque son difíciles de despersonalizar. Lo cual no quita la existencia de transparencia dentro del equipo del proyecto.

Métricas de producto

Contar algo que tenga que ver con el producto.

Gestión tradicional:

- Cantidad de líneas de código
- Cantidad de casos de uso por complejidad
- Cantidad de defectos conocidos

Hay aspectos que no se puede medir por ejemplo la calidad necesitamos usar epifenómenos, cosas relacionadas para poder medir, por ejemplo porcentaje de requerimientos cumplidos del cliente.

En el plan de proyecto se establecen las métricas que se van a utilizar. Se debe establecer el qué el como , el quien y el cuando se va a medir. Si no la métrica no sirve.

Las métricas no son perfectas, pueden tener errores. Pero nos permiten tener una visibilidad objetiva por lo cual se justifica usar métricas.

Métricas básicas de la gestión tradicional

Estas métricas debería tener para arrancar desde cero con una definición de métricas de un proyecto de software:

- Tamaño del producto → se recomienda algo relacionado con los requerimientos (CU por complejidad)
- Esfuerzo → horas persona lineales (en ágil se llama horas ideales), horas persona lineales por actividades o workflows

- Tiempo (calendario) → unidad de medida puede cambiar de acuerdo al tamaño del proyecto (días, semanas, meses)
- Defectos → cantidad de defectos por severidad o densidad de defectos (cuantos defectos por bloque de código ya sea CU, módulo, etc.

El sueño del pibe

Hay necesidades e intereses diferentes de acuerdo al rol del equipo

Manténgalo simple

La precisión es cara. Lo importante es mantener las cosas lo más simple posible.

Tenemos que satisfacer necesidades y expectativas de múltiples personas, debemos balancear para poder entregar las métricas con la información que cada persona requiera.

Métricas de software en Ambientes Ágiles

Una filosofía minimalista sobre las métricas:

Sacar todas la métricas que no tienen que ver con el software funcionando.

Se basa en los principios ágiles:

- “El software funcionando es la principal medida de progreso”
- “Nuestra mayor prioridad es satisfacer al cliente por medio de entregas tempranas y continuas de software valioso.

Velocidad

Cantidad de puntos de historia (producto) que presentamos y el PO nos aceptó.

La velocidad no se estima, se calcula.

Capacidad (Velocidad estimada según algunos autores)

Se estima para determinar hasta cuantos story points nos podemos comprometer a realizar en un sprint. Se pueden medir en horas ideales o en story points.

Si el equipo es maduro y no necesita la desagregación de las user stories en tareas puede medir la capacidad en puntos de historia.

Running Tested Features (RTF)

Cuántas características se implementaron y testearon. No se usa porque mide cantidades absolutas (features, user story), no tiene en cuenta la cantidad de story points o la complejidad de cada feature.

Todas estas métricas se miden en el contexto de un sprint

Velocidad → métrica de producto

Capacidad → métrica de proyecto

La experiencia no se extrapola de un proyecto a otro proyecto en ágil → En ágil no hay métricas de proceso.

Métricas en Kanban

Framework para la mejora de proceso → las métricas apuntan al proceso.

Lead Time

Vista del cliente. Tiempo desde que el cliente nos pidió algo hasta que lo entregamos.

Cycle Time

Vista interna o del cliente. Desde que empecé a poner foco en el trabajo hasta que se lo entregamos al cliente. Ritmo de terminación.

Touch Time

Mide tiempo que un ítem de trabajo pasa en las columnas de trabajo.

Eficiencia del Ciclo de proceso

$\text{Eficiencia ciclo proceso} = \text{Touch Time} / \text{Lead Time (elapsed time)}$

Hay desperdicios evitable e inevitables → hay algunos desperdicios que podemos eliminar y otros que no.

Métricas orientadas al servicio (de kanban)

Miden nuestra calidad en la entrega de servicios:

- Expectativa de nivel de servicio que los clientes esperan.
- Capacidad del nivel de servicio al que el sistema puede entregar.
- Acuerdo de nivel de servicio que es acordado con el cliente (SLA).
- Umbral de la adecuación del servicio el nivel por debajo del cual este es inaceptable para el cliente.

Para producto siempre tenemos que medir

Métricas de producto

- Tamaño
 - Líneas de código
 - Requerimientos
 - Casos de uso
 - Features
 - Clases
 - Puntos de Historia
- Defectos
 - Cobertura (% del producto que cubrimos con nuestros caso de prueba ejecutados, porcentaje de cosas probadas sobre el total del producto)
 - Defectos por severidad
 - Densidad de defectos