Métricas en los 3 enfoques (Tradicional, Ágil, Lean)

Una métrica es un valor para construir indicadores que faciliten la toma de decisiones. Es la medida, la presencia o el grado de un atributo en el aspecto que yo quiero medir (es decir, cuanto de ese atributo hay).

Una métrica es objetiva. Es un numero que se tiene que poder medir.

Hay muchas métricas en el enfoque tradicional que vienen a cumplir con el objetivo de entregarle valor al cliente.

El dominio de las métricas del software se divide en:

* Métricas de proceso
* Métricas de proyecto
* Métricas de producto

Las personas NO se miden.

Las del proceso son públicas, son de la organización y se obtienen consolidando las métricas del proyecto (se toman muchos proyectos y se despersonalizan). Hablan de la organización y lo que le sucede a la misma. Se utilizan fundamentalmente para mejorar el proceso buscando obtener los números que nos interesan tener.

Las métricas del proyecto son privadas, solo es visible para los involucrados en el mismo. Recién se publican a nivel organizacional cuando se despersonalizan y pasan a formar parte de las métricas de proceso.

Las métricas del producto miden al software. Una forma de medir producto es la medición de las líneas de código (aunque sea fácil, no sirve para nada). Lo que sí sirve es medir es requerimientos, calidad, etc. y como no se pueden medir directamente se utilizan los epifenómenos. Estos últimos buscan obtener un valor numérico de lo anterior mencionado, por ejemplo, la cantidad de requerimientos que se satisfacen del cliente, o la cantidad de clics para determinar la usabilidad, cantidad de defectos.

Las métricas que se van a tomar dependen de la organización. Las métricas no son perfectas.

Las métricas básicas para un proyecto dentro del enfoque tradicional son:

* Tamaño del producto
* Esfuerzo (son horas personas lineales)
* Tiempo (calendario)
* Defectos (Ej.: defectos por severidad y densidad de defectos por bloque de código)

El ámbito donde se generan las métricas es el proyecto y luego esas se trasladan al producto.

(VER EJEMPLOS DE METRICAS)

La precisión en las métricas es cara.

Métricas en ambientes AGILES:

Se basan en la idea de que “la mejor medida de progreso es el software funcionando”.

Velocidad: cantidad de story points (producto) que presentamos y el PO nos aceptó. No se estima, se calcula. Mide software funcionando.

Capacidad (velocidad estimada): se usa en el sprint planing para configurar el Sprint Backlog. Se mide en horas ideales (equipos iniciales) o en story points (equipos más maduros).

Running Tested Features (RTF): ya casi no se usa, porque mide en cantidades absolutas.

La capacidad (métrica de proyecto) y velocidad (métrica de producto) se miden en un contexto de un sprint. Como la experiencia no se puede extrapolar, no se pueden tomar métricas del proceso (cada equipo define el suyo y lo cambia a conveniencia).

Métricas en KANBAN (es un framework o método para la mejora de procesos):

Apuntan a medir procesos.

Cycle Time (le importa al equipo)

Lead Time (le importa al cliente)

Touch Time (toma solo las columnas de trabajo)

Eficiencia del Ciclo del Proceso

La cobertura mide cuanto de lo que se entrega fue probado (casos de pruebas ejecutados) y es un porcentaje (cantidad de cosas probadas por sobre el total del producto).

////////////////////////////////////////////////////////////////////////////////////////////////////

Del CMMI (estudiar significado) hay que saber que tiene 3 constelaciones:

* Dev
* Servicios
* Adquisición (no es lo mismo que terciarización)

Tiene 2 representaciones (hay que saber ambas diferencias):

* En etapas (se habla de madurez de la organización, medida con niveles de 1 a 5, y determina las cosas que esta puede hacer)
* Continua (mide capacidad de área de procesos, y se las evalúa de manera individual, puede estar de 0 a 5 niveles)

Las constelaciones son como especializaciones

Volatilidad de requerimientos

NO va a preguntar de modelo de calidad de producto

Lo mas importante de la diferencia entre SCRUM y KANBAN es que el primero trabaja con sprints de duración fija con una planificación establecida, y lo que busca KANBAN es flujo continuo