Métricas, Estimación y Planificación en Proyectos de Software

Cuando se planifica un proyecto se tiene que obtener estimaciones del costo y esfuerzo humano requerido por medio de las mediciones de software que se utilizan para recolectar los datos cualitativos acerca del software y sus procesos para aumentar su calidad.

De las Métricas

En la mayoría de los desafíos técnicos, las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo, *el producto se mide para intentar aumentar su calidad*.

El principio, podría parecer que la necesidad de la medición e s algo evidente. Después de todo es lo que nos permite cuantificar y por consiguiente gestionar de forma más efectiva. Pero la realidad puede ser muy deferente. Frecuentemente la medición con lleva una gran controversia y discusión.

1. ¿Cuáles son las métricas apropiadas para el proceso y para el producto?
2. ¿Cómo se deben utilizar los datos que se recopilan?
3. ¿Es bueno usar medidas para comparar gente, procesos o productos?

Estas preguntas y otras tantas docenas de ellas siempre surgen cuando se intenta medir algo que no se ha medido en el pasado.

La medición es muy común en el mundo de la ingeniería. Medimos potencia de consumo, pesos, dimensiones físicas, temperaturas, voltajes, señales de ruidos por mencionar algunos aspectos. Desgraciadamente la medición se aleja de lo común en el mundo de la ingeniería del software. Encontramos dificultades en ponernos de acuerdo sobre que medir y como va evaluar las medidas.

Hay varias razones para medir un producto.

1. Para indicar la calidad del producto.
2. Para evaluar la productividad de la gente que desarrolla el producto.
3. Par evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
4. Para establecer una línea de base para la estimación
5. Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Las mediciones del mundo físico pueden englobarse en dos categorías: medidas directas y medidas indirectas.

Medidas Directas. En el proceso de ingeniería se encuentran el costo, y el esfuerzo aplicado, las líneas de código producidas, velocidad de ejecución, el tamaño de memoria y los defectos observados en un determinado periodo de tiempo.

Medidas Indirectas. Se encuentra la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, etc.

MÉTRICAS DEL SOFTWARE.

Son las que están relacionadas con el desarrollo del software como funcionalidad, complejidad, eficiencia.

MÉTRICAS TÉCNICAS: Se centran en lasa características de software pro ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo esta hecho.

MÉTRICAS DE CALIDAD: proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

MÉTRICAS DE PRODUCTIVIDAD. Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que voy a diseñar.

MÉTRICAS ORIENTADAS A LA PERSONA. Proporcionan medidas e información sobre la forma que la gente desarrolla el software de computadoras y sobre todo el punto de vista humano de la efectividad de las herramientas y métodos. Son las medidas que voy a hacer de mi personal que va hará el sistema.

MÉTRICAS ORIENTADAS AL TAMAÑO. Es para saber en que tiempo voy a terminar el software y cuantas personas voy a necesitar. Son medidas directas al software y el proceso por el cual se desarrolla, si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño como se muestra en la siguiente figura:

PROYECTO	ESFUERZO	$	KLDC	PAGS. DOC	ERRORES	GENTE
999-01	24	168	12.1	365	29	3
CCC-04	62	440	27.2	1124	86	5
FFF-03	43	314	20.2	1050	64	6
.	.	.	.	.	.	.
.	.	.	.	.	.	.

La tabla lista cada proyecto del desarrollo del software de los últimos años correspondientes, datos orientados al tamaño de c/u. Refiriéndonos a la entrada de la tabla del proyecto 999-01 se desarrollaron 12.1 KLDC (*miles de líneas de código*) con un esfuerzo de 24 personas mes y un costo de 168 mil dólares. Debe tenerse en cuenta que el esfuerzo y el costo registrados en la tabla incluyen todas las actividades de la ingeniería de software como son análisis, diseño, codificación y prueba. Otra información del proyecto 222-01 indica que se desarrollaron 365 paginas mientras que se encontraron 29 errores tras entregárselo al cliente, dentro del primer año de utilización también sabemos que trabajaron 3 personas en el desarrollo del proyecto.

En los rendimientos del sistema y los rudimentarios datos contenidos en la tabla se puede desarrollar, para cada proyecto un conjunto de métricas sencillas de productividad y calidad orientadas al tamaño. Se obtienen las siguientes formulas:

Productividad = KLDC/persona-mes

Calidad = errores/KLDC

Documentación = pags. Doc/ KLDC

Costo = $/KLDC

- persona-mes es el esfuerzo

MÉTRICAS ORIENTADAS A LA FUNCIÓN. Son medidas indirectas del software y del proceso por el cual se desarrolla. En lugar de calcularlas las LDC, las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.

Las métricas orientadas a la función fueron el principio propuestas por Albercht quien sugirió un acercamiento a la medida de la productividad denominado método del punto de función. Los puntos de función que obtienen utilizando una función empírica basando en medidas cuantitativas del dominio de información del software y valoraciones subjetivos de la complejidad del software.

Los puntos de función se calculan rellenando la tabla como se muestra en la siguiente figura:

Calculo de métricas de punto de función.

Factor de ponderación						
Parámetro de medición	**Cuenta**		**Simple**	**Medio**	**Complejo**	
Numero de entradas de usuario		X	3	4	6	=
Numero de salidas de usuario		X	4	5	7	=
Numero de peticiones de usuario		X	3	4	6	=
Numero de archivos		X	7	10	15	=
Numero de interfaces externas		X	5	7	10	=
Cuenta = Total	→					

Se determinan 5 características del ámbito de la información y los cálculos aparecen en la posición apropiada de la tabla. Los valores del ámbito de información están definidos de la siguiente manera.

1. *Números de entrada de usuario:* se cuenta cada entrada del usuario que proporcione al software diferentes datos orientados a la aplicación. Las entradas deben ser distinguidas de las peticiones que se contabilizan por separado.
2. *Numero de salida del usuario:* se encuentra cada salida que proporciona la usuario información orientada ala aplicación. En este contexto las salidas se refieren a informes, pantalla, mensajes de error. Los elementos de datos individuales dentro de un informe se encuentran por separado.
3. *Números de peticiones al usuario:* una petición esta definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva. Se cuenta cada petición por separado.
4. *Numero de archivos:* se cuenta cada archivo maestro lógico, o sea una agrupación lógica de datos que puede ser una parte en una gran base de datos o un archivo independiente.
5. *Numero de interfaces externas:* se cuentan todas las interfaces legibles por la maquina por ejemplo: archivos de datos, en cinta o discos que son utilizados para transmitir información a otro sistema.

Cuando han sido recogidos los datos anteriores se asocian el valor de complejidad a cada cuenta. Las organizaciones que utilizan métodos de puntos de función desarrollan criterios para determinar si una entrada es denominada simple, media o compleja. No obstante la determinación de la complejidad es algo subjetivo.

Para calcular los puntos de función se utiliza la siguiente relación.

PF = CUENTA_TOTAL * [0.65 + 0.01 * SUM(fi)]

Donde CUENTA_TOTAL es la suma de todas las entradas de PF obtenidas de la tabla anterior.

Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad basados en las respuestas a las cuestiones señaladas de la siguiente tabla.

Evaluar cada factor en escala 0 a 5.

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

Fi

1. ¿Requiere el sistema copias de seguridad y recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es critico el rendimiento?
5. ¿Será ejecutado el sistema en un entorno operativo existente y frecuentemente utilizado?
6. ¿Requiere el sistema entrada de datos interactivo?
7. ¿Requiere la entrada de datos interactivo que las transiciones de entrada se lleven a cabo sobre múltiples o variadas operaciones?
8. ¿Se actualizan los archivos maestros en forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidos en el diseño la conversión y la instalación?
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

Los valores constantes de la ecuación anterior y los factores de peso aplicados en las encuestas de los ámbitos de información han sido determinados empíricamente.

Una vez calculado los puntos de función se usan de forma analógica a las LDC como medida de la productividad, calidad y otros productos del software.

Productividad = PF / persona-mes

Calidad = Errores / PF

Costo = Dólares / PF

Documentación = Pags. Doc / PF

La medida de puntos de función se diseño originalmente para ser utilizadas en aplicación de sistemas de información de gestión. Sin embargo, algunas aplicaciones se les denomina puntos de características.

La medida del punto de característica da cabida a aplicaciones cuya complejidad algoritmica es alta. Las aplicaciones de software de tiempo real de control de procesos y de sistemas que

encontrados tienden a tener una complejidad algoritmica alta y por tanto fácilmente tratables por puntos de características.

Para calcular los puntos de características, nuevamente se cuentan y ponderan los valores del ámbito de información, como se describió anteriormente. Además, las métricas de punto de característica tienen en cuenta otra característica del software, los algoritmos.

Un algoritmo se define como un problema de complejidad computacional limitada que se incluye dentro de un determinado programa de computadora. La inversión de una matriz, la decodificación de una cadena de bits o el manejo de una interrupción son todo ellos ejemplos de algoritmos.

Para calcular los puntos de característica, se utiliza la siguiente tabla.

Puntos de característica

	FACTOR DE PONDERACIÓN			
Parámetro de medición	**Cuenta**		**Complejo**	
Numero de entradas de usuario		X	4	=
Numero de salidas de usuario		X	5	=
Numero de peticiones de usuario		X	4	=
Numero de archivos		X	7	=
Numero de interfaces externas		X	7	=
Algoritmos		X	3	=
Cuenta = Total				

Se usa único valor de peso para cada uno de los parámetros de medida y se calcula el valor del punto característica global mediante la ecuación.

PF = CUENTA_TOTAL * [0.65 + 0.01 * SUM(fi)]

Debe tenerse en cuenta que los puntos de característica y los puntos de función representan lo mismo. "funcionalidad o utilidad" en forma de software.

Actividades Obligatorias

Métricas orientadas al tamaño

Proyecto	Esfuerzo	$	KLDC	Pag. doc	Errores	Gente
Farmacia	30	168,500	12,100	378	29	5
Hospital	60	578,300	39,443	921	540	20

- Calcular:

  a) Productividad = KLDC/esfuerzo

  - Hopital = ?
  - farmacia = ?

  b) Calidad = Errores/KLDC

  - Hospital = ?
  - Farmacia = ?

  c) Costo = $/KLDC

  - Hospital = ?
  - Farmacia = ?

  d) Documentación = Pags. doc/KLDC

  - Hospital=?
  - Farmacia=?

Métricas orientadas a la función

- Se tiene un sistema el cual cuenta con 3 entradas de catalogo productos, proveedores y clientes. Una pantalla de la elaboración de facturas, 4 tipos de reportes proporcionados tanto en pantalla como en papel. Estas representaciones son: factura, lista de inventario, estado de cuenta de los clientes y estado de cuenta con los proveedores. Además la entrada de factura tiene alrededor de 30 peticiones, el sistema genra alrededor de 30 archivos además de estar conectado a un lector óptico y una impresora. Calcula los puntos de función.

Parámetro de medida	Cuenta		Factor de peso medio		
Numero de entradas al usuario	4	*	4	=	16
Numero de salidas al	8	*	5	=	40

usuario					
Numero de peticiones al usuario	30	*	4	=	120
Numero de archivos	30	*	10	=	300
Numero de interfaces externas	2	*	7	=	14
			Cuenta total	=	490

Contestación de las preguntas basada en la complejidad media

1=0; 2=5; 3=3; 4=5; 5=5;

6=5; 7=1; 8=5; 9=2; 10=2;

11=4; 12=0; 13=0; 14=4

- Fi =?
- PF = ?
- Productividad = ?
- Calidad = ?
- Costo = ?
- Documentación = ?

ESTIMACIÓN

Es una pequeña planeación sobre que es lo que va a ser mi proyecto. Una de las actividades cruciales del proceso de gestión del proyecto del software es la planificación. Cuando se planifica un proyecto de software se tiene que obtener estimaciones de esfuerzo humano requerido, de la duración cronológica del esfuerzo humano requerido, de la duración cronológica del proyecto y del costo. Pero en muchos de los casos las estimaciones se hacen valiéndose de la experiencia pasada como única guía. Si un proyecto es bastante similar en tamaño y funciona un proyecto es bastante similar en tamaño y funciona un proyecto pasado es probable que el nuevo proyecto requiera aproximadamente la misma cantidad de esfuerzo, que dure aproximadamente lo mismo que el trabajo anterior. Pero que pasa si el proyecto es totalmente distinto entonces puede que la experiencia obtenida no sea lo suficiente.

Se han desarrollado varias técnicas de estimación para el desarrollo de software, aunque cada una tiene sus puntos fuertes y sus puntos débiles, todas tienen en común los siguientes atributos.

1. Se han de establecer de antemano el ámbito del proyecto.
2. Como bases para la realización de estimaciones se usan métricas del software de proyectos pasados.
3. El proyecto se desgloba en partes más pequeñas que se estiman individualmente.

PLANEACIÓN DEL PROYECTO

La planeación efectiva de un proyecto de software depende de la planeación detallada de su avance, anticipado problemas que puedan surgir y preparando con anticipación soluciones tentativas a ellos. Se supondrá que el administrador del proyecto es responsable de la planeación desde la definición de requisitos hasta la entrega del sistema terminado. No se analizara la planeación que implica a la estimación de la necesidad de un sistema de software y la habilidad de producir tal sistema, la asignación de prioridad al proceso de su producción.

Los puntos analizados posteriormente generalmente son requeridos por grandes sistemas de programación, sine embargo estos puntos son validos también para sistemas pequeños.

Panorama. Hace una descripción general del proyecto detalle de la organización del plan y resume el resto del documento.

Plan de fases. Se analiza el ciclo de desarrollo del proyecto como es: análisis de requisitos, fase de diseño de alto nivel, fase de diseño de bajo nivel, etc. Asociada con cada fase debe de haber una fecha que especifique cuando se debe terminar estas fases y una indicación de como se pueden solapar las distintas fases del proyecto.

Plan de organización. Se definen las responsabilidades especificas de los grupos que intervienen en el proyecto.

Plan de pruebas. Se hace un esbozo general de las pruebas y de las herramientas, procedimientos y responsabilidades para realizar las pruebas del sistema.

Plan de control de modificaciones. Se establece un mecanismo para aplicar las modificaciones que se requieran a medida que se desarrolle el sistema.

Plan de documentación. Su función es definir y controlar la documentación asociada con el proyecto.

Plan de capacitación. Se describe la preparación de los programadores que participan en el proyecto y las instrucciones a los usuarios para la utilización del sistema que se les entregue.

Plan de revisión e informes. Se analiza como se informa del estado del proyecto y se definen las revisiones formales asociadas con el avance de proyecto.

Plan de instalación y operación. Se describe el procedimiento para instalar el sistema en la localidad del usuario.

Plan de recursos y entregas. Se resume los detalles críticos del proyecto como fechas programadas, marcas de logros y todos los artículos que deben entrar bajo contrato.

Indice. Se muestra en donde encontrar las cosas dentro del plan.

Plan de mantenimiento. Se establece un bosquejo de los posibles tipos de mantenimiento que se tienen que dar para futuras versiones del sistema.

ERRORES CLASICOS EN UN PROYECTO DE SOFTWARE

1. Mal análisis en los requerimientos.
2. Una mala planeación.
3. No tener una negociación (documento, contrato) con el cliente.
4. No hacer un análisis costo beneficio.
5. Desconocer el ambiente de trabajo de los usuarios.
6. Desconocer los usuarios que trabajan con el sistema.
7. Mala elección de recursos (hardware, software, humanos).