

4. Proceso de software y métricas de proyectos

Ingeniería del Software 03-04
Antonio Navarro

1

Ingeniería del Software 03-04
Antonio Navarro

2

Índice

- Introducción
- Medidas, métricas e indicadores
- Métricas en el proceso y del proyecto
 - Introducción
 - Métricas del proceso y mejoras en el proceso de software.
 - Métricas del proyecto.
 - Métricas del software. Clasificación.

Índice

- Métricas de productividad
 - Orientadas al tamaño.
 - Orientadas a la función.
 - Otras métricas
 - Factores que inciden en la productividad.
- Relación entre líneas de código y puntos de función
- Métricas de calidad

Ingeniería del Software 03-04
Antonio Navarro

3

Índice

- Introducción.
- Errores.
- Medida de la calidad.
- Eficacia de la eliminación de defectos.
- Fiabilidad del software.
- Línea base de Métricas
- Conclusiones

Ingeniería del Software 03-04
Antonio Navarro

4

Introducción

- La existencia de medidas numéricas facilita el conocimiento de un fenómeno
- Las métricas del software *miden el software de computadora*
- Estas métricas sirven para:
 - Utilizarlas en el proyecto para ayudar en la estimación, control de calidad, evaluación de la productividad y control de proyectos.

Ingeniería del Software 03-04
Antonio Navarro

5

Introducción

- El desarrollador de software evalúa la calidad de los productos y trabajos técnicos.
- Ayudar en la toma de decisiones *tácticas* según avanza el proyecto.
- Aplicarlas al proceso con la idea de mejorarlo.

Ingeniería del Software 03-04
Antonio Navarro

6

Introducción

- Hay cuatro razones para medir:
 - Caracterizar.
 - Evaluar.
 - Predecir.
 - Mejorar.

Ingeniería del Software 03-04
Antonio Navarro

7

Medidas, métricas e indicadores

- Una *medida* proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.
 - E.g., un programa tiene 10.000 LDC (líneas de código).

Ingeniería del Software 03-04
Antonio Navarro

8

Medidas, métricas e indicadores

- La *medición* es el acto de determinar una medida
 - E.g., Ana será la encargada de medir las LDC de cada módulo del sistema.

Medidas, métricas e indicadores

- Una *métrica* es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado
 - E.g., la *productividad* de este proyecto fue de 500 (LDC/persona-mes)

Medidas, métricas e indicadores

- Las medidas no sirven para comparar, necesitamos métricas
 - E.g., en el país *A* ganan 1000 (€/pm), y en el país *B* ganan 1500 (€/pm) → ¿viven mejor en el país *B* que en el país *A*?
Una *Big Mac* cuesta 3€ en el país *A*, y en el país *B* cuesta 5€. Echemos cuentas.
País *A*: $1000(\text{€/pm})/3(\text{€/BM}) = 333,33 \text{ (BM/pm)}$
País *B*: $1500(\text{€/pm})/5(\text{€/BM}) = 250 \text{ (BM/pm)}$

Conclusión: no sabemos donde se vive mejor, pero en el país *A* una persona durante un mes puede comer un 33% más de *Big Macs* que en el país *B*

Medidas, métricas e indicadores

- Es decir,
 - La medida captura una característica individual.
 - La medición permite capturar dicha característica.
 - La métrica permite relacionar y comparar mediciones.

Medidas, métricas e indicadores

- Las métricas son el fundamento de los *indicadores*
- Un *indicador* es una métrica o combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software o del producto en si.

Métricas en el proceso y del ... Introducción

- Nuestros objetivos son establecer:
 - Métricas del proyecto → indicadores del proyecto.
 - Métricas del proceso → indicadores del proceso.
- Los *indicadores del proyecto* permiten al gestor:
 - Evaluar el estado del proyecto en curso.
 - Seguir la pista de riesgos potenciales.

Medidas, métricas e indicadores

- E.g., en el país *A*, no han aumentado los sueldos en los últimos tres años, pero el *índice Big Mac* se ha duplicado en ese periodo
- E.g., la productividad media de nuestra empresa es de 500(LDC/pm) y en el último proyecto ha sido de 250(LDC/pm)

Métricas en el proceso y del ... Introducción

- Detectar áreas problemáticas antes de que se conviertan en críticas.
- Ajustar el flujo y las tareas de trabajo.
- Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo de la IS.
- Los indicadores del proceso permiten:
 - Al gestor, evaluar lo que funciona y lo que no.

Métricas en el proceso y del ...

Introducción

- A la organización, tener una visión profunda de la eficacia de un proceso ya existente.
- Técnicamente no existe gran diferencia entre las métricas del proyecto y del proceso
- Podemos concebir las métricas del proceso como recopilaciones de métricas del proyecto

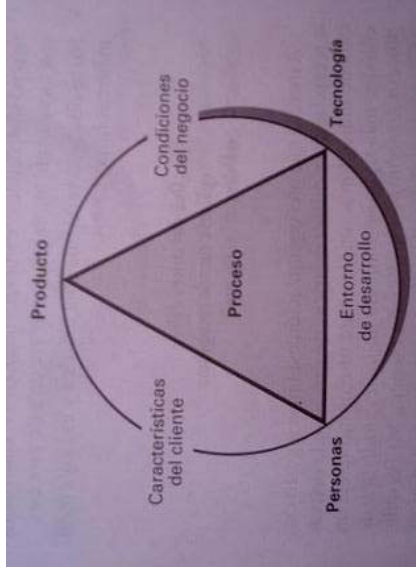
Métricas del proceso y del...

Métricas del proceso y mejora...

- Métricas del proceso → indicadores del proceso → mejora en el proceso
- Si la gestión se basa en el personal, problema y proceso, ¿por qué nos centramos en mejorar el proceso?
- Por qué el proceso es un factor clave y *controlable* para mejorar la calidad del software y el rendimiento de la organización

Métricas del proceso y del...

Métricas del proceso y mejora...



El proceso y diversos factores de un proyecto

Métricas del proceso y del...

Métricas del proceso y mejora...

- ¿Cómo vamos a medir el proceso?
- Como ya hemos comentado, las métricas del proceso se extraen de las métricas del proyecto
- En cualquier caso hay métricas *privadas* y otras *públicas*

Métricas del proceso y del... Métricas del proceso y mejora...

- Métricas privadas:
 - Índices de defectos.
 - Errores de desarrollo.
- Públicas para el equipo:
 - Índices de defectos.
 - Errores de desarrollo.
 - LDC.
 - PF.

Métricas del proceso y del... Métricas del proceso y mejora...

- No utilizar métricas para amenazar a particulares o equipos.
- Si una métrica identifica un área problemática no se debería considerar como negativa.
- Hay que interpretar todas las métricas en su conjunto, y no primar una en particular.

Métricas del proceso y del... Métricas del proceso y mejora...

- Las métricas del proceso pueden ser muy útiles, pero hay que saber interpretarlas
- Unas normas básicas de interpretación son
 - Utilizar el sentido común al interpretar los datos.
 - Proporcionar una realimentación regular a particulares y equipos.
 - No utilizar métricas para evaluar a particulares.
 - Establecer métricas claras y objetivos para alcanzarlas.

Métricas del proceso y del... Métricas del proceso y mejora...

- La utilización de métricas e indicadores fiables da lugar a una *mejora estadística del proceso del software*
- Esta mejora se basa en un análisis de fallos que identifica la causa y origen de *errores* y *defectos* para varios proyectos de software
 - *Error*: fallo en un producto generado durante el proceso de IS que es detectado antes de la entrega al cliente.

Métricas del proceso y del... Métricas del proceso y mejora...

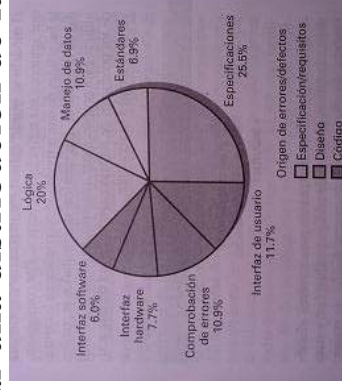
- *Defecto*: fallo detectado después de la entrega al cliente.
- El análisis de fallos funciona:
 1. Se categorizan por origen todos los errores y defectos de varios proyectos.
 2. Se registra el *coste* de corregir cada error o defecto.
 3. El número de errores y de defectos de cada categoría se cuentan y se ordenan decrecientemente

Ingeniería del Software 03-04
Antonio Navarro

25

Métricas del proceso y del... Métricas del proceso y mejora...

- Aplicando los pasos 1 y 2 se puede desarrollar una distribución de fallos



Causas de fallos y su origen para varios proyectos

Ingeniería del Software 03-04
Antonio Navarro

27

Métricas del proceso y del... Métricas del proceso y mejora...

4. Se computa el coste global de errores y defectos de cada categoría.
5. Los datos resultantes se analizan para detectar las categorías que producen el coste más alto para la organización.
6. Se desarrollan planes para modificar el proceso con el intento de eliminar (o reducir la frecuencia de apariciones de) la clase de errores y defectos que sean más costosos.

Ingeniería del Software 03-04
Antonio Navarro

26

Métricas del proceso y del... Métricas del proceso y mejora...

- También podemos optar por desarrollar un *diagrama de espina* para ayudar a diagnosticar los datos presentados en el diagrama de frecuencias.
- Las líneas horizontales identifican problemas, y las verticales posibles causas
- Damos diagramas para cada origen de defecto y los estudiamos para mejorar el proceso

Ingeniería del Software 03-04
Antonio Navarro

28

Métricas del proceso y del... Métricas del proceso y mejora...

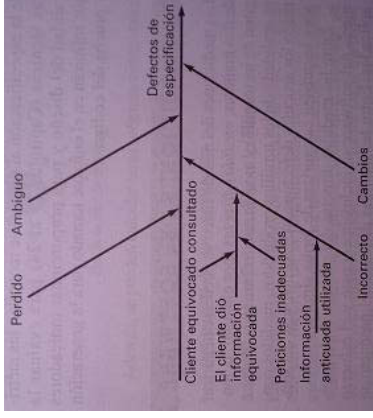


Diagrama de espina

Métricas del proceso y del... Métricas del proyecto

- A medida que avanza el proyecto, las medidas del esfuerzo y el tiempo se comparan con las planificación.
- El gestor utiliza estos datos para supervisar y controlar el avance.
- Además, para medidas en las técnicas de diseño y programación existen *métricas técnicas*

Métricas del proceso y del... Métricas del proyecto

- Las métricas del proceso son *estratégicas*: determinan el curso del proceso de producción de software
- Las métricas del proyecto son *tácticas*: determinan el curso del proyecto actual
- La primera aplicación de las métricas del proyecto ocurre durante la estimación (datos históricos).

Métricas del proceso y del... Métricas del proyecto

- La utilización fundamental de las métricas del proyecto son dos:
 - Minimizar la planificación de desarrollo, guiando los ajustes necesarios que eviten retrasos y mitiguen problemas y riesgos potenciales.
 - Evaluar la calidad de los productos en el momento actual, modificando el enfoque técnico para mejorar la calidad, si es necesario.

Métricas del proceso y del... Métricas del software

- Como el contexto de uso identifica al tipo de métrica, nos referiremos a las métricas del producto y del proceso como *métricas del software*

Métricas de productividad Orientadas al tamaño

- Se obtiene considerando las medidas de productividad y normalizándolas por el tamaño del código, es decir las *Líneas De Código* (LDC)
- Se basan en la utilización de registros sencillos para las *medidas* más relevantes para nuestro proyecto

Métricas del proceso y del... Métricas del software

MÉTRICAS DEL SOFTWARE	Productividad	Calidad
Tamaño	$\frac{\text{euros}}{\text{LDC}} \frac{\text{pgDoc}}{\text{KLDC}}$	$\frac{\text{errores}}{\text{KLDC}} \frac{\text{defectos}}{\text{KLDC}}$
PF, PC, PF3D)	$\frac{\text{euros}}{\text{PF}} \frac{\text{pgDoc}}{\text{PF}}$	$\frac{\text{errores}}{\text{PF}} \frac{\text{defectos}}{\text{PF}}$
Otras	$\frac{\text{LDC}}{\text{per-mes}} \frac{\text{PF}}{\text{per-mes}} \frac{\text{euros}}{\text{pgDoc}}$	$\frac{\text{errores}}{\text{per-mes}} \frac{\text{per-mes}}{\text{TMC}}, \frac{\text{desperdicios}}{\text{EED}}, \frac{\text{integridad}}{\text{EED}}, \frac{\text{disponibilidad}}{\text{PFBD, FDF, TMF}}$

Métricas del software. Clasificación

Métricas de productividad Orientadas al tamaño

Proyecto	LDC	Esfuerzo*	Coste*	Paginas	Errores	Defectos	Personas
			€(000)	Doc.			
P1	12.100	24	120	365	134	29	3
P2	27.200	62	314	1224	321	86	5
P3	20.200	43	224	1050	256	64	6

* Incluye todas las actividades de IS (análisis, diseño, codificación y prueba)

Medidas relevantes para el establecimiento de métricas

Métricas de productividad

Orientadas al tamaño

- ¿Qué es el *esfuerzo*?
esfuerzo = #personas * #tiempo
- Es una medida que indica que *da igual* tener dos personas trabajando tres meses, que tres personas trabajando dos meses
 $e = 3(p) * 2(m) = 6(pm)$
 $e = 2(p) * 3(m) = 6(pm)$

Métricas de productividad

Orientadas al tamaño

- Métricas orientadas al tamaño
 - Coste: #euros/#LDC
↓mejor
e.g. P1: $120000(€)/12100(LDC) = 9,92 (€/LDC)$
 - Documentación: #pgDoc/#KLDC
↑mejor
e.g. P2: $1224(pgDoc)/27,2(KLDC) = 45(pgDoc/KLDC)$

Métricas de productividad

Orientadas al tamaño

- ¿Cómo calcular las LDC?
 - Debe contabilizarse cada línea nueva o modificada.
 - Las líneas para la instrumentación de código (e.g. para las pruebas) no deben incluirse en el tamaño total, salvo que tengan un carácter definitivo.
 - Las líneas de código de programas de prueba tan solo se contabilizan si se desarrollan con el nivel de calidad exigido al entregar el producto.

Métricas de productividad

Orientadas al tamaño

- Se contabilizan las líneas correspondientes a las llamadas al sistema operativo.
- No se consideran los comentarios.
- No se contabiliza el pseudocódigo.
- Cada ocurrencia de *macro* o *include* se considera como una línea.
- El código generado por *macros* o *includes* solo se considera una vez.

Métricas de productividad Orientadas al tamaño

- Las LDC no están comúnmente aceptadas
- Ventajas:
 - Fácil de calcular.
 - Existen muchos modelos de estimación basados en LDC.
 - Existen muchas medidas de LDC

Métricas de productividad Orientadas al tamaño

- Inconvenientes:
 - Dependientes de los lenguajes de programación.
 - Perjudican a los programas cortos, pero bien diseñados.
 - Difícil uso en estimación debido al nivel de detalle.

Métricas de productividad Orientadas a la función

- Se obtienen considerando las medidas de productividad y normalizándolas por una medida de la *funcionalidad* entregada por la aplicación
- Como la funcionalidad no se puede medir directamente, se debe derivar indirectamente de otras medidas directas

Métricas de productividad Orientadas a la función

- La funcionalidad de un programa viene representada por el *Punto de Función* (PF), que se deriva de las mediciones del software
- Se calcula en base a la expresión
$$PF = \text{cuenta-total} * (0,65 + 0,01 * \sum_{i=1..14} F_i)$$
donde

Métricas de productividad

Orientadas a la función

Parámetros de medición	Factor de ponderación			Cuenta	Simple	Medio	Complejo	=
Número de entradas de usuario		×	3		4		6	
Número de salidas de usuario		×	4		5		7	
Número de peticiones de usuario		×	3		4		6	
Número de archivos		×	7		10		15	
Número de interfaces externas		×	5		7		10	
Cuenta total								

Cálculo de de *cuenta-total*

Métricas de productividad

Orientadas a la función

- *Peticiones de usuario*. Entradas interactivas que producen la generación de alguna respuesta del software inmediata en forma de de salida interactiva.
- *Archivos*. Se cuenta cada archivo maestro lógico, i.e., cada grupo de datos que puede ser una parte de una gran base de datos o sistema de archivos.

Métricas de productividad

Orientadas a la función

- Parámetros de medición:
 - *Entradas de usuario*. Entradas de usuario que proporcionan diferentes datos orientados a la aplicación.
 - *Salidas de usuario*. Salidas que proporcionan al usuario información orientada a la aplicación (e.g. informes, pantallas, mensajes de error, etc.).

Métricas de productividad

Orientadas a la función

- *Interfaces externas*. Interfaces legibles por la máquina que se utilizan para transmitir información a otro sistema (e.g. cinta, red, etc.).
- Los *valores de ajuste complejidad* (F_i) se calculan respondiendo a las siguientes preguntas en una escala desde 0 (no importante o aplicable) hasta 5 (absolutamente esencial):

Métricas de productividad Orientadas a la función

1. ¿Requiere el sistema copias de seguridad y de recuperación fiables?
2. ¿Se requiere comunicación de datos.
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?

Ingeniería del Software 03-04
Antonio Navarro

49

Métricas de productividad Orientadas a la función

10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidas en el diseño la conversión e 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 ser fácilmente utilizada por el usuario?

Ingeniería del Software 03-04
Antonio Navarro

51

Métricas de productividad Orientadas a la función

6. ¿Requiere el sistema entrada de datos interactiva?
7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
8. ¿Se actualizan los archivos maestros de forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?

Ingeniería del Software 03-04
Antonio Navarro

50

Métricas de productividad Orientadas a la función

- Una vez calculado el valor PF, las métricas son análogas a las orientadas al tamaño
 - Coste: #euros/#PF
↓mejor
 - Documentación: #pgDoc/#PF
↑mejor

Ingeniería del Software 03-04
Antonio Navarro

52

Métricas de productividad Orientadas a la función

- La medida de *Punto de Característica* (PC) es una ampliación de la medida de PF
- La medida de PF tiene su origen en aplicaciones de gestión
- Prima por tanto la dimensión de *datos*, obviando cuestiones de complejidad *funcional*

Métricas de productividad Orientadas a la función

- Consiste en ampliar la tabla de *cuenta-total* de PF con el parámetro de medición *algoritmos*
- Un *algoritmo* es un problema de cálculo limitado que se incluye dentro de un programa
- El factor de ponderación depende de la importancia que se quiera dar a este parámetro (e.g. 10, 15, 20)

Métricas de productividad Orientadas a la función

- Esto hace a la medida de PF inadecuada para sistemas de ingeniería o empotrados
- Solución: ampliar los parámetros de medición para tener en cuenta a los algoritmos
- El PC es una ampliación de la medida de PF aplicable a sistemas con una fuerte componente funcional (e.g. tiempo real)

Métricas de productividad Orientadas a la función

- Los PF *tampoco* están comúnmente aceptados
- Ventajas
 - Independientes del lenguaje de programación.
 - Permiten hacer estimaciones más fácilmente.
- Inconvenientes
 - Basadas en *cálculos subjetivos*.
 - Parámetros y factores no evidentes.
 - No tienen un significado físico directo.

Métricas de productividad

Otras métricas

- Son *cruciales* pero no están normalizadas por LDC ni por PF
- Productividad: $\#LDC/\#persona\text{-}mes$
↑mejor
e.g. P3: $20200(LDC)/43(pm) = 469,77$ (LDC/pm)
- Productividad: $\#PF/\#persona\text{-}mes$
↑mejor

Métricas de productividad

Otras métricas

- Coste documentación: $\#euros/\#páginas\ doc.$
↓mejor
P1: $120000(€)/365(pgDoc) = 328,77(€/pgDoc)$

Métricas de productividad

Factores que inciden en ...

- Los gestores no deben utilizar *directamente* las métricas de productividad para evaluar a la gente
- La razón reside en que no todos los proyectos son iguales
- Hay una serie de *factores* que afectan a la productividad:
 - *Factores humanos*. Tamaño y experiencia de la organización de desarrollo.

Métricas de productividad

Factores que inciden en ...

- *Factores del problema*. La complejidad del problema que se debe resolver y el número de cambios en las restricciones o los requisitos de diseño.
- *Factores del proceso*. Técnicas de análisis y diseño que utilizan, lenguajes y herramientas CASE y técnicas de revisión.
- *Factores del producto*. Fiabilidad y rendimiento del sistema.

Métricas de productividad

Factores que inciden en ...

- *Factores del recurso*. Disponibilidad de herramientas CASE y recursos de hardware y software.
- Si uno de los factores es favorable (desfavorable) la productividad será significativamente más alta (más baja)

Métricas de productividad

Factores que inciden en ...

Factor	Variación aproximada (%)
Humano	90
Problema	40
Proceso	50
Producto	140
Recurso	40

Variación de la productividad en función del factor según Jones

- Discusión: ¿no es contradictoria esta tabla con las *tres pes* de gestión?

Relación entre LDC y PF

- Las LDC y los PF son medidas en principio independientes
- ¿No es razonable suponer que la funcionalidad de un sistema y su tamaño están relacionadas? (e.g. MS-DOS vs. Windows XP)
- Se puede hacer una estimación *informal* del número de LDC necesarios para construir un PF

Relación entre LDC y PF

Lenguaje de programación	LDC/PF (media)
Ensamblador	320
C	128
COBOL	106
FORTRAN	106
Pascal	90
C++	64
Ada95	53
Visual Basic	32
Smalltalk	22
Powerbuilder (generador cod.)	16
SQL	12

Relación entre LDC y PF

Relación entre LDC y PF

- Conclusiones:
 - Cuanto más avanzado es un lenguaje, más expresivas son sus sentencias.
 - Jones no cursó *Sistemas Operativos* con COBOL como lenguaje de implementación.
 - Jones no cursó *Investigación Operativa* con SQL como lenguaje de implementación.

Relación entre LDC y PF

Complejidad	FAB
Muy simple	0,7
Simple	0,85
Media	1
Moderadamente compleja	1,2
Compleja	1,3

Factor de ajuste de backfiring

- Es decir, *compensamos* la funcionalidad con la falta (exceso) de tamaño

Relación entre LDC y PF

- Aunque podríamos aplicar directamente la relación anterior para calcular los PF en base a las LDC (*backfiring*), refinaremos un poco más esta técnica.
$$PF = LDC_{aplicación} / ((LDC/PF_{media}) * FAB)$$
- FAB es el *Factor de Ajuste de Backfiring*

Relación entre LDC y PF

- E.g. consideremos un sistema con 45.000 LDC C y una mejora de la IGU posterior de 10.000 LDC C++. Supongamos que la parte C es compleja, y la parte C++ simple
$$PF_C = 45000 / (128 * 1,3) = 270 \text{ PF}$$
$$PF_{C++} = 10000 / (30 * 0,85) = 392 \text{ PF}$$
$$PF_{aplicación} = 270(PF) + 392(PF) = 662 (PF)$$

Métricas de calidad

Introducción

- Base de IS: calidad
- Calidad de análisis, diseño, codificación, prueba: *métricas técnicas*
- Efectividad de las actividades de control y garantía de calidad: *métricas de calidad*

Métricas de calidad

Errores

- Tenemos:
 - #errores/#KLDC
↓mejor
e.g. P2: $321(e)/12,1(KLDC) = 26,53$ (e/KLDC)
 - #errores/#PF
↓mejor
- #errores/#persona-mes
↓mejor
e.g. P3: $256(e)/43(pm) = 5,95$ (e/pm)

Métricas de calidad

Medida de la calidad

- Vamos a ver una serie de *factores* que afectan a la calidad y como medirlos
- *Corrección*
 - Grado en que el software lleva a cabo su función requerida.
 - #defectos/#KLDC
↓mejor
e.g. P1: $29(d)/12,1(KLDC) = 2,4$ (d/KLDC)

Métricas de calidad

Medida de la calidad

- #defectos/#PF
↓mejor
- Un *defecto* es una falta verificada de conformidad con los requisitos.
- Facilidad de mantenimiento
 - Facilidad con la que se puede corregir un programa si se encuentra un error, se puede adaptar a su entorno si cambia, o mejorar si el cliente desea un cambio de requisitos

Métricas de calidad

Medida de la calidad

- Una métrica orientada al tiempo es el *Tiempo Medio de Cambio* (TMC): tiempo que se tarda en analizar la petición de cambio, en diseñar una modificación adecuada, implementar el cambio, en probarlo y en distribuir el cambio a todos los usuarios.
- Cuanto más fácil sea de mantener un programa, más bajo tendrá su TMC.

Métricas de calidad

Medida de la calidad

- Una métrica orientada al coste son los *desperdicios*: coste en corregir defectos encontrados después de haber distribuido el software a los usuarios finales.
- Integridad
 - Mide la habilidad de un sistema para resistir ataques (tanto accidentales como intencionados) contra su seguridad.

Métricas de calidad

Medida de la calidad

- El ataque puede producirse en cualquier componente del software (programas, datos o documentos).
- Para medir la integridad debemos medir la *seguridad* y la *amenaza*, las cuales se estiman o deducen de la evidencia empírica

Métricas de calidad

Medida de la calidad

amenaza: probabilidad de que se produzca un ataque de tipo determinado en un momento determinado.

seguridad: probabilidad de que se pueda repeler el ataque de un tipo determinado en un momento determinado

$\text{integridad} = \sum_{\text{ataques}} [1 - \text{amenaza} * (1 - \text{seguridad})]$

Métricas de calidad

Medida de la calidad

- e.g.

	P1 no oculta ficheros no hace backup Amenaza	Seguridad	P2 oculta ficheros hace backup Amenaza	Seguridad
.....				
borrado BD aplicación	0,7	0	0,2	0,8
.....				

$$\text{integridad}_{P1\text{borrado}} = 1 - 0,7 * (1 - 0) = 0,3$$
$$\text{integridad}_{P2\text{borrado}} = 1 - 0,2 * (1 - 0,8) = 0,96$$

Métricas de calidad

Medida de la calidad

- Facilidad de uso
 - Intento por medir lo *amigable* que puede ser un programa con el usuario.
 - Se puede medir en función de cuatro características:
 - Habilidad intelectual y/o física para aprender el sistema.
 - Tiempo requerido para llegar a ser moderadamente eficiente en el uso del sistema.

Métricas de calidad

Medida de la calidad

- Aumento neto de la productividad (sobre el sistema que reemplaza) medida cuando alguien utiliza el sistema de manera moderadamente eficiente.
- Valoración subjetiva (a veces obtenida mediante un cuestionario) de la disposición de los usuarios hacia el sistema.

Métricas de calidad

EED

- La *Eficacia de la Eliminación de Defectos* (EED) es una medida de la habilidad de filtrar de las actividades de la garantía de calidad y de control, al aplicarse a todas las actividades del marco de trabajo del proceso.

Métricas de calidad

EED

- Considerada globalmente para el proyecto:
 $EED = E/(E+D)$
donde
E: número de errores encontrados antes de la entrega
D: número de defectos
• Objetivo: $EED = 1$

Métricas de calidad

EED

$$EED_i = E_i / (E_i + E_{i+1})$$

donde

E_i : errores detectados en la actividad i de IS
 E_{i+1} : errores detectados en la actividad i+1 de IS que no se detectaron y provienen de la actividad i

- Objetivo $EED_i = 1$

Métricas de calidad

EED

- Nótese que si E es muy grande, EED estará próxima a 1 \rightarrow cuanto más errores encontremos antes de la entrega, mejor funcionarán las técnicas de garantía de calidad
- La EED también puede utilizarse para medir la habilidad de un equipo para encontrar errores antes de pasar a la siguiente AE

Métricas de calidad

Fiabilidad del software

- *Fiabilidad* del software: ausencia de fallos
- *Probabilidad de Fallo Bajo Demanda* (PFBD)
 - Mide la probabilidad de que un sistema falle cuando se le hace una petición de servicio.
 - $PFBD = \#fallos / \#peticiones$
 - \downarrow mejor.

Métricas de calidad

Fiabilidad del software

- e.g. una PFBD de 0,001 significa que el sistema tiene un fallo cada mil peticiones.
- *Frecuencia de fallo*
 - Mide la frecuencia de aparición de fallo de funcionamiento.
 - $FDF = \#fallos/\#unidad\ tiempo$
 - ↓ mejor
 - e.g. una FDF de 0,006 indica que se producen 3 fallos cada 500 unidades de tiempo

Ingeniería del Software 03-04
Antonio Navarro

85

Métricas de calidad

Fiabilidad del software

- *Disponibilidad*
 - Mide la disponibilidad de un sistema para ser usado.
 - $dispon. = \#tiempo\ disponible/\# tiempo\ funcionando$
 - ↑ mejor.
 - e.g. Una disponibilidad de 0,95 indica que el sistema está disponible 950 unidades de cada 1000 unidades de tiempo.
- Unidad de tiempo (CPU, días, etc.)

Ingeniería del Software 03-04
Antonio Navarro

87

Métricas de calidad

Fiabilidad del software

- Discusión: ¿es una probabilidad la FDF?
- *Tiempo medio de fallo (TMF)*
 - Mide el tiempo transcurrido entre fallos del sistema.
 - $TMF = \#unidades\ de\ tiempo/\#fallos$
 - ↑ mejor.
 - e.g. un TMF de 166,66 indica en 500 unidades de tiempo transcurridas se han producido 3 fallos.
 - Si no hay cambios, $TMF = 1/FDF$

Ingeniería del Software 03-04
Antonio Navarro

86

Línea base de métricas

- Una *línea base de métricas* es una recopilación de métricas que sirve para establecer indicadores
- Un ejemplo sencillo es la tabla de la t36
- No tiene nada que ver con el concepto de línea base que veremos en GCS

Ingeniería del Software 03-04
Antonio Navarro

88

Línea base de métricas

- Para ser útil debe tener los siguientes atributos:
 - Los datos deben ser razonablemente exactos.
 - Los datos deben extraerse del mayor número de proyectos que sea posible.
 - Las medidas deben ser consistentes.
 - Las aplicaciones deben ser semejantes para hacer la estimación.

Conclusiones

- Medir: conocer
- Medida, métrica e indicador
- Métricas del proceso, proyecto y software
- Métricas proceso: estratégicas
- Métricas proyecto: tácticas
- Mejoramos el proceso porque es *controlable*

Conclusiones

- Mejora estadística del proceso
- Normas de interpretación de métricas
- Métricas técnicas
- Métricas de productividad y calidad
- Métricas orientadas al tamaño y a la función
- Relación entre LDC y PF
- *Sentido común*