Tema 09 – Métricas

Ingeniería del Software

Rubén Fuentes Fernández
Dep. Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense Madrid

Trabajando con Antonio Navarro, Juan Pavón y Pablo Gervás

Contenidos

- Introducción
  - Problemática
- Aspectos de la solución
  - Medidas, métricas e indicadores
  - Métricas del proceso
  - Métricas del proyecto
  - Métricas de productividad
  - Métricas de calidad
  - Línea base de métricas

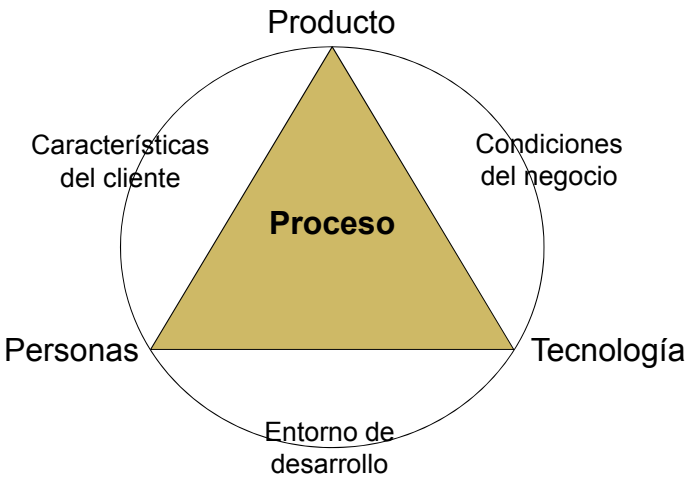
Introducción

- La existencia de medidas numéricas facilita el conocimiento de un fenómeno.
- Hay cuatro razones para medir:
  - Caracterizar
  - Evaluar
  - Predecir
  - Mejorar

Medidas, métricas e indicadores

- A la hora de medir manejamos 3 conceptos diferentes:
  - Una *medida* proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.
    - Ej. un programa tiene 10000 LDC (Líneas De Código)
  - Una *métrica* es una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.
    - Ej. la productividad del proyecto fue de 500 LDC/PM (LDC / Persona-Mes)
  - 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 sí.
    - Ej. la productividad media de nuestra empresa es de 500 LDC/PM y en el último proyecto ha sido de 250 LDC/PM.

Proceso y factores del proyecto



Mediciones en el proceso y del proyecto

- El objetivo es doble, porque se busca establecer:
  - Métricas del proyecto → indicadores del proyecto
  - Métricas del proceso → indicadores del proceso
- Las métricas del proceso son estratégicas.
  - Determinan el curso del proceso de producción de productos.
  - Múltiples proyectos.
- Las métricas del proyecto son tácticas.
  - Determinan el curso del proyecto actual.
- Técnicamente no existe gran diferencia entre ambas.
  - Podemos concebir las métricas del proceso como recopilaciones de métricas del proyecto.

Indicadores en el proceso y el proyecto

- Los *indicadores del proyecto* permiten al gestor:
  - Evaluar el estado del proyecto en curso.
  - Seguir la pista de riesgos potenciales.
  - 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 Ingeniería del Software (IS).
- Los *indicadores del proceso* permiten:
  - Al gestor, evaluar lo que funciona y lo que no.
  - A la organización, tener una visión profunda de la eficacia de un proceso ya existente.

Métricas y mejora del proceso

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

Métricas privadas y públicas del proceso

- Métricas privadas:
  - Índices de defectos
  - Errores de desarrollo
- Públicas para el equipo:
  - Índices de defectos
  - Errores de desarrollo
  - LDC
  - Puntos Función (PF)

Uso de las métricas del proceso

- 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.
  - Establecer métricas claras y objetivos para alcanzarlas.
  - 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.
  - No utilizar métricas para evaluar o amenazar a particulares o equipos.

Mejora estadística del proceso

- 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.
  - El *error* es un fallo en un producto generado durante el proceso de IS que es detectado antes de la entrega al cliente.
  - El *defecto* es un fallo detectado después de la entrega al cliente.

Análisis de fallos del proceso

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.
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.

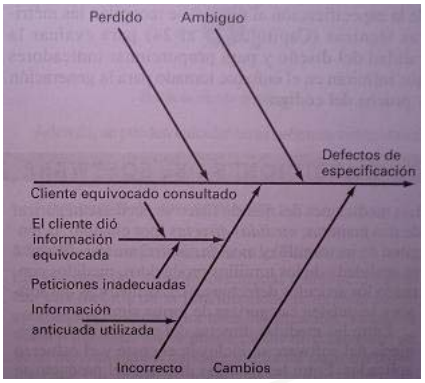
Diagrama de frecuencias de fallos

- Aplicando los pasos 1 y 2 del proceso de análisis de fallos se puede elaborar un diagrama de frecuencias con la distribución de fallos.

Causas de fallo y su origen para varios proyectos

Diagrama de espina de fallos

- También se puede optar por desarrollar un diagrama de espina para ayudar a diagnosticar los datos presentados en el diagrama de frecuencias.
- En el diagrama de espina las líneas horizontales identifican problemas y las verticales posibles causas.
- Se dan diagramas para cada origen de defectos y se estudian para mejorar el proceso.



Métricas del proyecto

- Las métricas del proyecto se emplean para determinar el curso del proyecto actual.
  - La primera aplicación de las métricas del proyecto ocurre durante la estimación.
    - Se emplean datos históricos para hacerla.
  - Después el gestor utiliza los datos de las métricas para supervisar y controlar el avance.
    - A medida que avanza el proyecto, las medidas del esfuerzo y el tiempo se comparan con las de la planificación.
    - Permiten modificar el enfoque técnico para mejorar la calidad si es necesario.
- En el proyecto también se emplean *métricas técnicas* o de producto para medir las técnicas de diseño y programación.

Métricas del software

- El contexto de uso identifica al tipo de métrica.
- Por ello nos referiremos globalmente a las métricas del producto y del proceso como *métricas del software*.

Métrica del SW	Productividad	Calidad
Tamaño	€ / LDC PgDoc / KLDC	errores / KLDC defectos / KLDC
PF (resp. PC)	€ / PF PgDoc / PF	errores / PF defectos / PF
Otras	LDC / PM PF / PM € / PgDoc	errores / PM errores / (errores + defectos)

Métricas de productividad

- Orientadas al tamaño
  - Se obtienen considerando las medidas de productividad y normalizándolas por el tamaño del código (ej. LDC) o del equipo (ej. persona).
- Orientadas a la función
  - Se obtienen considerando las medidas de productividad y normalizándolas por una medida de la *funcionalidad* (ej. PF) entregada por la aplicación.
  - Como la funcionalidad no se puede medir directamente, se debe derivar indirectamente de otras medidas directas.
- Otras
  - Cruciales pero no están normalizadas por tamaño ni funcionalidad, aunque sí por ejemplo por esfuerzo (ej. PM).

Pautas para el cálculo de LDC

- Debe contabilizarse cada línea nueva o modificada.
- Las líneas para la instrumentación de código no deben incluirse en el tamaño total, salvo que tengan un carácter definitivo.
  - Ej. código usado para pruebas.
- 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.
- 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* sólo se considera una vez.

LDC: ventajas y limitaciones

- Las LDC no están comúnmente aceptadas.
  - Su cálculo no es trivial.
- Ventajas:
  - Fácil de calcular.
  - Existen muchos modelos de estimación basados en LDC.
  - Existen muchas medidas de LDC.
- Inconvenientes:
  - Dependientes de los lenguajes de programación.
  - Perjudican a los programas cortos pero bien diseñados.
  - Difícil su uso en estimación debido al nivel de detalle que requieren.

Puntos de función

- La funcionalidad de un programa viene representada por el Punto de Función (PF), que se deriva de las mediciones del software.
- Se basan en una combinación de características del programa:
  - entradas del usuario
  - salidas (presentadas) al usuario
  - consultas del usuario (interacciones o peticiones)
  - archivos usados por el sistema
  - interfaces externas
- Ver *Tema 7 - Estimación de proyectos*

Puntos de característica

- 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*, es decir, algoritmos.
  - Esto hace a la medida de PF inadecuada para sistemas de ingeniería o empotrados.
- Solución: ampliar los parámetros de medición para tener en cuenta los algoritmos.

Definición del punto de característica

- El PC amplia 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.
  - Ej. 10, 15, 20…

PF: ventajas y limitaciones

- 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.

Relación entre LDC y PF por lenguaje

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 de código)	16
SQL	12

LDC y PF son medidas en principio independientes, pero se puede hacer una estimación informal del número de LDC necesarias para construir un PF. Supone que la funcionalidad está relacionada con el tamaño.

Métricas de productividad: otras

$$productividad = \#LDC/\#PM$$

- ↑ mejor
- Ej. productividad(P3) = 20200 LDC / 43 PM = 469,77 LDC/PM

$$productividad = \#PF/\#PM$$

- ↑ mejor

$$coste\ de\ documentación = \#euros/\#PgDoc$$

- ↓ mejor
- Ej. coste de documentación(P) = 12000 € / 366 PgDoc = 32,77 €/PgDoc

Variabilidad de la productividad

- Existen una serie de factores que afectan a la productividad.
  - Si uno de los factores es favorable (desfavorable) la productividad será significativamente más alta (más baja).
- Factores:
  - *Humanos* → Tamaño y experiencia de la organización de desarrollo
  - *Problema* → La complejidad del problema que se debe resolver y el número de cambios en las restricciones o los requisitos de diseño.
  - *Proceso* → Técnicas de análisis y diseño que se utilizan, lenguajes y técnicas de revisión.
  - *Producto* → Fiabilidad y rendimiento del sistema.
  - *Recursos* → Disponibilidad de herramientas CASE y recursos de hardware y software.

Métricas de calidad

- La base de la IS es la *calidad*.
- La calidad se mide en base a métricas:
  - Métricas técnicas o del producto
    - Calidad de análisis, diseño, codificación y prueba.
  - Métricas de calidad
    - Efectividad de las actividades de control y garantía de calidad.
    - Relacionadas entre otros aspectos con errores, defectos, tiempo de cambio, integridad, facilidad de uso…

Métricas de calidad: errores

$$tasa\ de\ errores = \#errores/\#KDLC$$

- ↓ mejor
  - Ej. tasa de errores(P2) = 321 errores / 12,1 KLDC = 26,53 errores/KLDC

$$tasa\ de\ errores = \#errores/\#PF$$

- ↓ mejor

$$tasa\ de\ errores = \#errores/\#PM$$

- ↓ mejor
  - Ej. tasa de errores(P3) = 256 errores / 43 PM = 5,95 errores/PM

Métricas de calidad: corrección

- Grado en que el software lleva a cabo su función requerida.
  - Un *defecto* es una falta verificada de conformidad con los requisitos.

$$corrección = \#defectos/\#KLDC$$
$$corrección = \#defectos/\#PF$$

- ↓ mejor
  - Ej. corrección(P1) = 29 defectos / 12,1 KLDC = 2,4 defectos/KLDC

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.

$$EED = \frac{E}{E + D}$$

- donde:
  - *E* es el número de errores encontrados antes de la entrega
  - *D* número de defectos
- El objetivo es EED = 1
- Nótese que si E es muy grande, EED estará próxima a 1
  - Cuanto más errores encontremos antes de la entrega, mejor funcionarán las técnicas de garantía de calidad.
- Se puede considerar:
  - Globalmente para el proyecto
  - Al nivel de actividad de ingeniería

Métricas de calidad: 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.
- Varias métricas posibles:
  - 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.
  - Una métrica orientada al coste son los *desperdicios*.
    - Coste en corregir defectos encontrados después de haber distribuido el software a los usuarios finales.

Métricas de calidad: integridad (1/2)

- Mide la habilidad de un sistema para resistir ataques (accidentales o intencionados) contra su seguridad.
- El ataque puede producirse en cualquier componente del software (programas, datos o documentos).
- Para medir la integridad se miden la seguridad y la amenaza, las cuales se estiman o deducen de la evidencia empírica.
  - *amenaza*: probabilidad de que se produzca un ataque de un tipo determinado en un momento determinado.
  - *seguridad*: probabilidad de que se pueda repeler el ataque de un tipo determinado en un momento determinado.

Métricas de calidad: integridad (2/2)

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

- Ej. peligro de borrado de la base de datos de la aplicación.

  La aplicación P1 no oculta los ficheros y no hace *backup*. Se estima que la probabilidad de la amenaza es 0,7 y la seguridad es 0.

  La aplicación P2 oculta los ficheros y hace *backup*. Se estima que la probabilidad de la amenaza es 0,2 y la seguridad es 0,8.

  integridad(P1, borrado) = 1 – 0,7 * (1 - 0) = 0,3

  integridad(P2, borrado) = 1 – 0,2 * (1 – 0,8) = 0,96

Métricas de 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 requerida para aprender el sistema.
  - Tiempo requerido para llegar a ser moderadamente eficiente en el uso del sistema.
  - 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.

Línea base de métricas

- Una *línea base de métricas* es una recopilación de métricas que sirve para establecer indicadores.
  - No tiene nada que ver con el concepto de *línea base* que se maneja en planificación ni en Gestión de Configuración del Software.
- 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

Conclusiones

- La Ingeniería del Software depende de la realización de mediciones apropiadas de sus distintos componentes.
  - Evaluación
  - Seguimiento
  - Mejora
- Estas mediciones pueden tener como objetivo distintos elementos de la Ingeniería del Software y sus proyectos
  - Proceso, proyecto y calidad
- así como aspectos de estos
  - Corrección, facilidad de uso, facilidad de mantenimiento, integridad…
- Atendiendo a su propósito diferenciamos entre medidas, métricas e indicadores.

Glosario

- EED = Eficacia en la Eliminación de Defectos
- IS = Ingeniería del Software
- KLDC = Miles de LDC
- LDC = Líneas de Código
- PC = Punto de Característica
- PF = Puntos de Función
- PM = Persona-Mes
- TMC = Tiempo Medio de Cambio

Referencias

- R. Pressman: Ingeniería del Software. Un enfoque práctico, 7ª edición. McGraw-Hill, 2010.
  - Capítulo 28
- I. Sommerville: Ingeniería del Software, 7ª edición. Addison Wesley, 2007.
  - Capítulo 5.4