Tema 09 – Métricas

Ingeniería del Software

Héctor Gómez Gauchía
Dep. Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense Madrid

Trabajando con Rubén Fuentes , 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

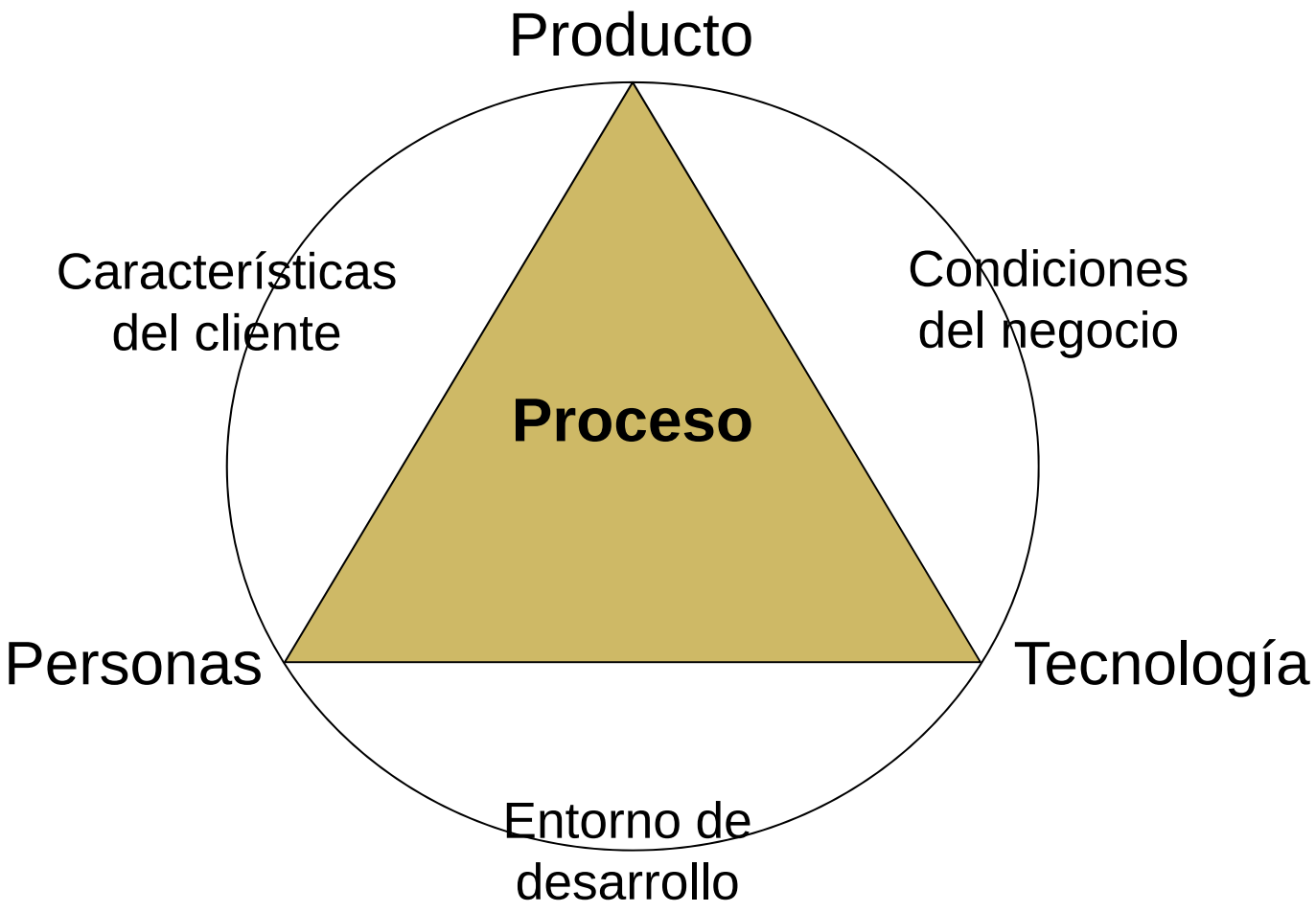
Introducción

- 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

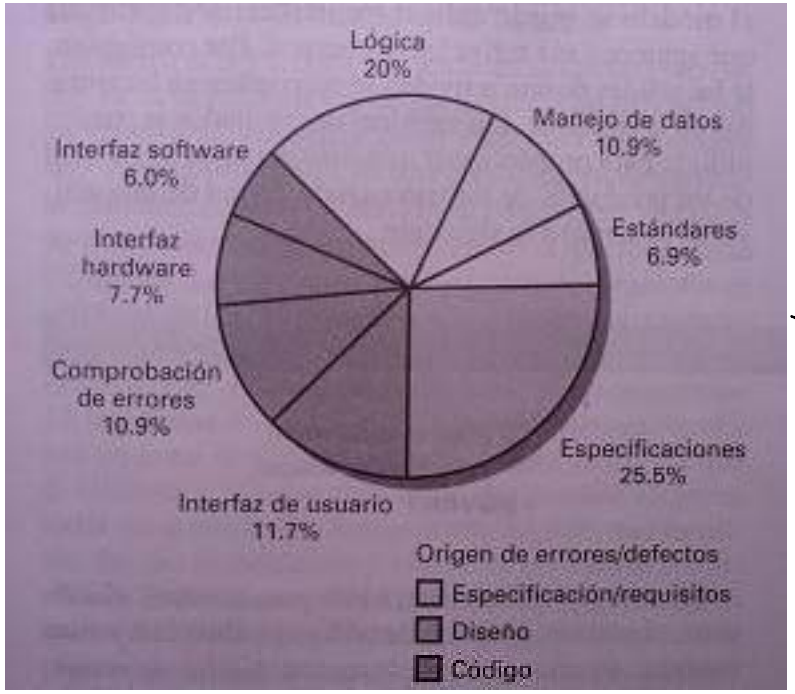
- La utilización de métricas e indicadores fiables da lugar a una mejora estadística del proceso del software.
- Esta mejora se basa en un análisis de fallos que identifica la causa y origen de *errores* y *defectos* para varios proyectos de software.
  - El *error* es un fallo en un producto generado 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 ellos 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 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 externos

- 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 a 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 necesarios 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 = 328,77 €/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 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 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 la 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

IO2

Diapositiva 39

IO2 8ª edición en inglés

IO; 29/11/2011

EXTRAS

Métricas del software

- Las métricas del software *miden el software de computadora*.
- Sirven para:
  - Gestionar el proyecto.
    - Ayudan en su estimación y control, en el control de calidad y en la evaluación de la productividad.
  - Evaluar la calidad de los productos y trabajos técnicos.
  - Ayudar en la toma de decisiones *tácticas* según avanza el proyecto.
  - Aplicarlas en la mejora del proceso.

Medidas y métricas (1/2)

- Las medidas no sirven para comparar, necesitamos métricas.
  - Ej. 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 (BM) cuesta 3€ en el país A, y en el país B cuesta 5€.
  Echemos cuentas…

  País A → 1000 €/PM / 3 €/BM = 333,33 BM/PM

  País B → 1500 €/PM / 5 €/BM = 300 BM/PM

  Conclusión: no sabemos donde se vive mejor, pero en el país A una persona durante un mes puede comer un 11% más de Big Macs que en el país B.

Medidas y métricas (2/2)

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

Métricas del proceso

- ¿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*.

- Discusión: ¿Qué sentido tiene distinguir entre métricas privadas y públicas? ¿Cuáles pueden ser? ¿Para quiénes son privadas y para quiénes públicas?

Usos de las métricas del proyecto

- Las métricas del proyecto tienen dos usos fundamentales:
  - Minimizar la planificación de desarrollo
    - Sirven de guía para realizar los ajustes necesarios que eviten retrasos y mitiguen problemas y riesgos potenciales.
  - Evaluar la calidad de los productos en el momento actual
    - Permiten modificar el enfoque técnico para mejorar la calidad si es necesario.

Medidas relevantes: orientadas al tamaño

- Las medidas incluyen todas las fases de la IS.
  - Análisis, diseño, codificación y prueba.

Proy.	Tipo	Leng.	LDC	Esfuerzo	Coste	PgDoc	Errores	Defectos	Pers.
P1	Tienda web	PHP	12100	24	120	365	134	29	3
P2	Tienda web	J2EE	27200	62	314	1224	321	86	5
P3	Tienda web	J2EE	20200	43	224	1050	256	64	6

Medidas relevantes: esfuerzo

Métricas de productividad: orientadas al tamaño - ejemplos

Características (1/2)

- 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.
  - Ej. informes, pantallas o mensajes de error.
- Peticiones de usuario
  - Entradas interactivas que producen la generación de alguna respuesta del software inmediata en forma de salida interactiva.

Características (2/2)

- 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.
- Interfaces externas
  - Interfaces legibles por la máquina que se utilizan para transmitir información de/a otro sistema.
  - Ej. red, servicios web o cintas.

Cálculo de valores

- Para cada una de las características anteriores se recogen varios datos.
  - Número de veces que aparecen en el proyecto
  - Complejidad de cada aparición
    - A elegir entre simple, media y compleja.
    - Cada nivel de complejidad tiene un peso.
      - El peso influye en el cálculo de los PF.

Peso a asignar			
Características	Simple	Media	Compleja
Entradas del usuario	3	4	6
Salidas al usuario	4	5	7
Consultas del usuario	3	4	6
Ficheros	7	10	15
Interfaces externos	5	7	10

Entradas de usuario: clasificación

- Entradas de usuario que proporcionan diferentes datos orientados a la aplicación.

Dificultad de las entradas – Ficheros accedidos	Número de campos o atributos de la entrada		
	1-4	5-15	> 16
0-1	Baja	Baja	Media
2	Baja	Media	Alta
> 3	Media	Alta	Alta

Salidas de usuario: clasificación

- Salidas que proporcionan al usuario información orientada a la aplicación.

Dificultad de las salidas – Ficheros accedidos	Número de campos o atributos de la salida		
	1-5	6-19	> 20
0-1	Baja	Baja	Media
2-3	Baja	Media	Alta
> 4	Media	Alta	Alta

Peticiones de usuario: clasificación

- Entradas interactivas que producen la generación de alguna respuesta del software inmediata en forma de salida interactiva.
- Son todos aquellos procesos que están formados por una combinación de entradas y salidas, produciendo una consulta a los datos.
- El flujo de datos deberá tener dos direcciones.
- Como consecuencia de una consulta no se modifican los datos del sistema.
- La complejidad de la consulta viene dada por la mayor entre la entrada y la salida.

Diapositiva 54

IO5 Corregir "Como consecuencia de una consulta no se modifican los datos del sistema." porrque es una entrada y una salida a la vez.
IO; 17/06/2011

Ficheros: clasificación

- 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.
- Es un grupo de datos relacionados, tal como los percibe el usuario y que son mantenidos por la aplicación.
- Los ficheros se cuentan una sola vez, independientemente del número de procesos que los acceden.

Dificultad de los ficheros lógicos – Registros	Número de campos o atributos de la salida		
	1-19	20-50	> 51
1	Baja	Baja	Media
2-5	Baja	Media	Alta
> 6	Media	Alta	Alta

Interfaces externas: clasificación

- Interfaces legibles por la máquina que se utilizan para transmitir información de/a otro sistema.
- Es un grupo de datos relacionados, tal como los percibe el usuario, referenciados por la aplicación y que son mantenidos por otra aplicación.
- Son ficheros internos de otra aplicación.

Dificultad de las interfaces externas – Registros lógicos	Número de campos o atributos		
	1-19	20-50	> 51
1	Baja	Baja	Media
2-5	Baja	Media	Alta
> 6	Media	Alta	Alta

Cuenta total

Puntos de función	Complejidad						Total
	Simple		Media		Compleja		
	#N	Peso	#N	Peso	#N	Peso	
Entradas		3		4		6	
Salidas		4		5		7	
Consultas del usuario		3		4		6	
Ficheros lógicos		7		10		15	
Interfaces externas		5		7		10	
Total de puntos de función sin ajustar							

Factores de complejidad

- Las aplicaciones tienes además ciertas características globales que inciden en su complejidad total.
- Estas características se resumen en los llamados Factores de Complejidad (FC).
  - Se eva

Valor del FC	Significado
0	Sin influencia, factor no presente
1	Influencia insignificante, muy baja
2	Influencia moderada o baja
3	Influencia media, normal
4	Influencia alta, significativa
5	Influencia muy alta, esencial

FC1 – Comunicación de datos

- Los datos usados en el sistema se envían o reciben por líneas de comunicaciones
- Valoración
  - 0 Sistema aislado del exterior
  - 1 Batch, usa periféricos E o S remotos
  - 2 Batch, usa periféricos E y S remotos
  - 3 Captura de datos en línea o teleproceso que pasa los datos o sistema de consulta
  - 4 Varios teleprocesos con mismo protocolo
  - 5 Varios protocolos. Sistema Abierto y con inter-faces de todo tipo al exterior

FC2 – Proceso distribuido

- Existen procesos o datos distribuidos y el control de éstos forma parte del sistema
- Valoración
  - 0 Sistema totalmente centralizado
  - 1 Sistema realiza procesos en un equipo, salidas usadas vía software por otros equipos
  - 2 Sistema captura, los trata en otro
  - 3 Proceso distribuido, transacciones en una sola dirección
  - 4 Idem, transferencia en ambas direcciones
  - 5 Procesos cooperantes ejecutándose en distintos equipos

FC3 – Objetivos de rendimiento

- Si el rendimiento es un requisito del sistema, es decir, es crítico algún factor como tiempo de respuesta o cantidad de operaciones por hora. Se tendrá que hacer consideraciones especiales durante el diseño, codificación y mantenimiento
- Valoración
  - 0 Rendimiento normal (no se da énfasis)
  - 1 Se indican requisitos, no medida especial
  - 2 Crítico en algunos momentos. Procesos acabados antes de próxima sesión de trabajo
  - 3 Tiempo de respuesta es crítico
  - 4 ... en diseño hacer análisis de rendimiento en tiempo respuesta o cantidad operaciones/hora
  - 5: .. uso herramientas para alcanzar el rendimiento demandado por el usuario

FC4 – Integración de la aplicación

- El sistema tendrá que ejecutarse en un equipo en el que coexistirá con otros compitiendo por los recursos, teniendo que tenerse en cuenta en las fase de diseño
- Valoración
  - 0 No se indican restricciones
  - 1 Existen las restricciones usuales
  - 2 Características de seguridad o tiempos
  - 3 Restricciones en algún procesador
  - 4 El software deberá funcionar con restricciones de uso en algún procesador
  - 5 Restricciones especiales para aplicación en los componentes distribuidos del sistema

FC5 – Tasa de transacciones

- La tasa de transacciones será elevada. Se tendrá que hacer consideraciones especiales durante el diseño, codificación e instalación
- Valoración
  - 0 No se prevén picos
  - 1 Se prevén picos poco frecuentes (mensual)
  - 2 Se prevén picos semanales
  - 3 Se prevén horas punta, diarias
  - 4 Tasa de transacciones tan elevada que en diseño se hace análisis de rendimiento
  - 5 Análisis de rendimiento en diseño, implementación e instalación

FC6 – Entrada de datos interactiva

- La entrada de datos será directa desde el usuario a la aplicación de forma interactiva
- Valoración
  - 0 Todo es Batch.
  - 1 1% < entradas interactivas < 7%
  - 2 8% < entradas interactivas < 15%
  - 3 16% < entradas interactivas < 23%
  - 4 24% < entradas interactivas < 30%
  - 5 Entradas interactivas > 30%

FC7 – Eficiencia para el usuario final (1/3)

- Se demanda eficiencia para el trabajo del usuario, es decir, se tiene que diseñar e implementar la aplicación con interfaces fáciles de usar y con ayudas integradas
- Tipos de elementos asociados a la eficiencia del usuario:
  - Menús
  - Uso de ratón
  - Ayudas “en línea”
  - Movimiento automático del cursor.
  - Efectos de Scroll (papiro)
  - Teclas de función predefinidas
  - …

FC7 – Eficiencia para el usuario final (2/3)

- Tipos de elementos asociados a la eficiencia del usuario:
  - …
  - Lanzamiento de procesos batch desde las transacciones “en línea”
  - Selección mediante cursor de datos de la pantalla
  - Pantallas con muchos colores y efectos
  - Posibilidad de “hard copy”
  - Ventanas de pop-up
  - Aplicación bilingüe (cuenta por cuatro)
  - Aplicación multilingüe (mas de dos, cuenta por seis)

FC7 – Eficiencia para el usuario final (3/3)

- Valoración
  - 0 No se da énfasis al tema
  - 1 1 a 3 de los factores
  - 2 4 a 5 de los factores
  - 3 6 o más factores, sin requerir eficiencia
  - 4 ... con requisitos que implican estudio de los factores humanos en el diseño
  - 5 … se demandan prototipos y herramientas para verificar que se alcanzaran los objetivos

FC8 – Actualizaciones interactivas

- Los ficheros maestros y/o las bases de datos son modificados de forma interactiva
- Valoración
  - 0 No hay
  - 1 De 1 a 3 ficheros con información de control; cantidad baja y ficheros recuperables
  - 2 ... pero con 4 o más ficheros de control
  - 3 Actualización de ficheros importantes
  - 4 ... esencial la protección ante pérdidas
  - 5 Gran cantidad de actualizaciones interactivas; sistemas de recuperación muy automatizados

FC9 – Lógica de proceso interna compleja (1/2)

- La complejidad interna en un proceso esta en función de las siguientes características
  - Especificados algoritmos matemáticos complejos
  - Proceso con lógica compleja
  - Especificado muchas excepciones, consecuencia de transacciones incompletas, que deberán tratarse
  - Manejar múltiples dispositivos de entrada / salida
  - Se incorporarán sistemas de seguridad y control.

FC9 – Lógica de proceso interna compleja (2/2)

- Valoración
  - 0 Ninguna de las características
  - 1 1 Característica
  - 2 2 Características
  - ...
  - 5 Las 5 características

FC10 – Reusabilidad del código

- Reusabilidad del código
- Valoración
  - 0 No se prevé
  - 1 Reutilizar código en la misma aplicación
  - 2 Menos de un 10% de la aplicación tiene en cuenta las necesidades de + de 1 usuario
  - 3 El 10 % o más ...
  - 4 Aplicación preparada para ser reutilizable a nivel de código
  - 5 Aplicación preparada para ser reutilizable por medio de parámetros

FC11 – Conversión e instalación

- Se proveerán facilidades de conversión e instalación en el sistema. Se tendrán que hacer consideraciones especiales durante el diseño, codificación y pruebas para que la conversión desde el sistema antiguo sea fácil de realizar durante la puesta en marcha del sistema nuevo
- Valoración
  - 0 No se requiere conversión
  - 1 Se solicita facilidad de instalación
  - 2 Se solicitan procesos de conversión e instalación, no importantes para el proyecto
  - 3 ... si son importantes
  - 4 2 y herramientas conversión e instalación
  - 5 3 y herramientas conversión e instalación; sistema crítico para la empresa

FC12 – Facilidad de operación (1/2)

- Facilitar la explotación real de la aplicación, dedicándole especial atención durante el diseño, codificación y pruebas del sistema
- Se pueden tener en cuenta las siguientes posibilidades de automatización:
    - Procesos de arranque, back-up y recuperación pero con intervención del operador
    - … sin intervención del operador (vale por 2)
    - Minimizar la necesidad de montar cintas u otros dispositivos de almacenamiento externo
    - Minimizar la necesidad de manejar papel

FC12 – Facilidad de operación (2/2)

- Valoración:
  - 0 No se especifica nada
  - 1 a 4 Sumar la cantidad de items de la lista anterior
  - 5 Sistema automático sin intervención humana

FC13 – Instalaciones múltiples

- El sistema ha de incluir los requisitos de diversas empresas o departamentos en donde se ejecutará (incluso plataformas). Estas características estarán presentes durante el diseño, codificación y pruebas
- Valoración
  - 0 1 solo lugar
  - 1 Múltiples lugares, mismo hardware y software
  - 2 En diseño se tiene en cuenta el caso (1)
  - 3 En diseño se tiene en cuenta múltiples entornos hardware y software
  - 4 Se documenta y planea para (1) y (2)
  - 5 Idem, para (3)

FC14 – Facilidad de cambios (1/2)

- Se tendrá que hacer consideraciones especiales durante el diseño, codificación y mantenimiento para que en el sistema sea fácil de introducir cambios y fácil de adaptar al usuario
- Puntos a considerar:
  - Consultas flexibles del usuario:
    - Simples - Con condiciones lógicas and/or que implican un único fichero lógico
    - Medias – Con condiciones lógicas sobre más de 1 fichero lógico (por 2)
    - Complejas - Con condiciones lógicas complejas que afectan a varios ficheros lógicos (por 3)
  - Parámetros de la aplicación con tablas ajenas al código:
    - El cambio se hace efectivo al arrancar el sistema
    - El cambio es interactivo (por 2)

FC14 – Facilidad de cambios (2/2)

- Valoración:
  - 0 No se especifica nada
  - 1 Un punto de valor 1
  - 2 Puntos por valor 2
  - 3: ...
  - 5 Puntos por valor 5

Tabla para el cálculo de los factores de complejidad

Id.	Factor de complejidad	Valor (1..5)
1	Comunicación de datos	
2	Proceso distribuido	
3	Objetivos de rendimiento	
4	Integración de la aplicación	
5	Tasa de transacciones	
6	Entrada de datos interactiva	
7	Eficiencia para el usuario final	
8	Actualizaciones interactivas	
9	Lógica de proceso interna compleja	
10	Reusabilidad del código	
11	Conversión e instalación	
12	Facilidad de operación	
13	Instalaciones múltiples	
14	Facilidad de cambios	
Factor de complejidad total		Σ

PF ajustados

- PFA = PFSA * (0,65 + (0.01 * FCT))

  PFA = Puntos de Función Ajustados

  PFSA = Puntos de Función Sin Ajustar

  FCT = Factor de Complejidad Total

- Cada factor de complejidad afecta en +/- 2,5% en los PFSA.

Métricas de productividad: orientadas a la función - ejemplos

Métricas de productividad: otras

- Cruciales pero no están normalizadas por LDC ni por PF.

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

- ↑ 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 = 328,77 €/PgDoc

Variación según los factores de productividad

- Variación de la productividad en función del factor según Jones.

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

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

Relación entre LDC y PF por lenguaje: observaciones

- 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.
- Jones se refiere a Sistemas de Información.

Relación entre LDC y PF por lenguaje: cálculo

FAB

- Compensamos la funcionalidad con la falta (exceso) de tamaño.

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

Relación entre LDC y PF por lenguaje: ejemplo

- Consideremos un sistema con 45000 LDC C y una mejora posterior de la interfaz gráfica de usuario de 10000 LDC C++
- Supongamos que la parte C es compleja y la parte C++ simple.

$PF_C = 45000 / (128*1,3) = 270$ PF

$PF_{C++} = 10000 / (30*0,85) = 392$ PF

$PF_{aplicación} = 270$ PF + 392 PF = 662 PF