

Métricas do processo e do projecto de software



Aula 4

Sumário

Introdução

- ▣ O quê é?
- ▣ Quem faz?
- ▣ Porquê é importante?
- ▣ Qual é o produto?
- ▣ Como saber se está bem feita?

Medidas, métricas e indicadores

Introdução (I)

O quê é?

- ▣ Medidas quantitativas que permitem ter uma visão aprofundada da eficácia do processo e dos projectos.
- ▣ Reúnem-se e analisam-se dados básicos de qualidade e produtividade que quando comparados com dados anteriores permitem detectar melhoras e áreas problema

Quem faz?

- ▣ Os engenheiros reúnem as medidas
- ▣ Os gestores administrativos analisam os dados

Introdução (II)

- # Porquê é importante?
 - ▣ Não se pode melhorar ou que não é medido
- # Qual o produto?
 - ▣ Conjunto de métricas de software e plano de medição
- # Como fazer bem?
 - ▣ Com um plano de medição e métricas simples mas consistentes

Medidas, medição e métricas no contexto da Eng. De Software

	<i>Definição</i>	<i>Exemplo</i>
Medida	Valor quantitativo da extensão, quantidade, dimensões, capacidade ou tamanho de algum atributo do processo ou produto de software	nº de erros detectados na revisão de 1 módulo de sw
Medição	Acto de determinar uma medida	investigação de um nº de revisões de módulos para recopilar medidas do nº de erros encontrados em cada revisão
Métrica	Medida quantitativa do grau de posse de um atributo dado por parte de um sistema, componente ou processo	Promédio de erros detectados por revisão ou nº de erros encontrados por pessoa e hora em revisões

Indicadores

- # Um indicador é uma métrica ou conjunto de métricas que proporcionam uma visão aprofundada do processo de software, do projecto ou do próprio software
- # Um indicador permite ajustar o produto, o projecto ou o processo para melhorar as coisas

Indicadores: processo e projecto

- # Os indicadores de processo permitem avaliar o modelo de processo, as tarefas e produtos de trabalho. São recolhidos a partir de todos os projectos durante algum tempo
- # Os indicadores de projecto permitem:
 - ▣ Avaliar o estado do projecto
 - ▣ Monitorizar riscos potenciais
 - ▣ Detectar áreas problemas
 - ▣ Ajustar o fluxo e tarefas
 - ▣ Avaliar o controlo de qualidade da equipa de trabalho
- # Muitas vezes, as mesmas métricas de software servem para ambos indicadores

Métricas de software

Domínios

- ▣ Processo
- ▣ Projecto
- ▣ Produto

Métricas do processo de software

- # A eficácia do processo é determinada a partir de medidas como:
 - ▣ N° erros detectados antes da entrega
 - ▣ Defeitos informados aos utilizadores
 - ▣ Produtos de trabalho entregues
 - ▣ Quantidade de esforço humano e tempo consumido
- # Métricas privadas
 - ▣ Dados individuais (índices de defeitos, defeitos x módulo, erros detectados durante o desenvolvimento)
- # Métricas públicas
 - ▣ Dados públicos (defeitos x função, erros encontrados em revisões formais e linhas de código o pontos de função x módulo e função)

Métricas do projecto

- # As métricas do processo são estratégicas, as do projecto, tácticas
- # Com o avanço do projecto, as medidas de tempo e esforço são comparadas com as estimativas para fazer ajustes
- # Do trabalho técnico surgem outras medidas como: índices de produção e erros
- # As métricas do projecto servem para:
 - ▣ Planear o desenvolvimento
 - ▣ Avaliar a qualidade dos produtos

Um modelo de métricas

- # Outro modelo sugere:
 - ▣ Entradas: quantidade de recursos.
 - ▣ Saídas: medidas de entregas ou produtos
 - ▣ Resultados: eficácia das entregas
- # Pode ser aplicado a o projecto e a o processo

Medições do software

- # Directas (+ fáceis de medir)
 - ▣ Linhas de código, velocidade execução, tamanho em memória, defeitos reportados
- # Indirectas (+ difíceis de medir)
 - ▣ Funcionalidade
 - ▣ Qualidade
 - ▣ Complexidade
 - ▣ Eficiência
 - ▣ Fiabilidade
 - ▣ Facilidade de uso
 - ▣ Flexibilidade

Métricas de software orientadas ao tamanho

projecto	LDC	Esforço	Costo	Pag.Dc	Erros	Defeitos
Alfa	12100	24	168	365	134	29
Beta	27200	62	440	1224	321	86
Gamma	20200	43	314	1050	256	64
...

- # Erros por KLDC
- # Defeitos por KLDC
- # Custo por KLDC
- # Páginas de documentação por KLDC
- # LDC por pessoa-mês
- # Custo por página de documentação

COCOMO

- # Modelo de estimação do esforço e custo de desenvolvimento em função ao n° de linhas de código
- # *Model 1.* The Basic COCOMO model is a static, single-valued model that computes software development effort (and cost) as a function of program size expressed in estimated lines of code (LOC).
- # *Model 2.* The Intermediate COCOMO model computes software development effort as a function of program size and a set of "cost drivers" that include subjective assessments of product, hardware, personnel and project attributes.
- # *Model 3.* The Advanced COCOMO model incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step (analysis, design, etc.) of the software engineering process.

COCOMO BASICO

$$\# E = a_b \times \text{KLOC} \times b_b$$

$$\# D = c_b \times E \times d_b$$

Parâmetros:

Software Project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

COCOMO Intermediário

1. Product attributes

- a. required reliability
- b. size of data base
- c. complexity

2. Hardware attributes

- a. run-time performance constraints
- b. memory constraints
- c. volatility of the virtual machine environment
- d. required turnaround time

3. Personnel attributes

- a. analyst capability
- b. software engineer capability
- c. applications experience
- d. virtual machine experience
- e. programming language experience

4. Project attributes

- a. use of software tools
- b. application of software engineering methods
- c. required development schedule

COCOMO Intermediário

- # Each attribute is rated on a 6 point scale that ranges from "very low" to "extra high"
- # An effort multiplier is determined from Boehm tables and the product of all effort multipliers results is an *effort adjustment factor* (EAF).
- # Typical values for EAF range from 0.9 to 1.4.
- # Fórmula: $E = a_i \text{ KLOC } b_i \times \text{EAF}$
 - E = effort applied in person-months
 - KLOC is the estimated number of delivered lines of code for the project.

Parâmetros:

Software project	a_i	b_i
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

Métricas de software orientadas à função

<i>Pto. De função</i>	<i>Soma</i>	<i>Simple</i>	<i>Médio</i>	<i>Complexo</i>	<i>Total</i>
Nº Entradas utilizador		3	4	6	
Nº Saídas utilizador		4	5	7	
Nº consultas		3	4	6	
Nº ficheiros		7	10	15	
Nº interfaces		5	7	10	
<i>Total Geral</i>					
<p>$PF = total \times [0,65 + 0,016(F_i)]$</p> <p>Fi = Valores de ajuste (0-5): necessidade de cópias de segurança, necessidade de comunicação de dados, importância do performance, complexidade do processamento interno, necessidade de suportar várias instalações</p>					

Métricas ampliadas de ponto de função

Pontos de características

- ▣ Aplicações de tempo real, controlo e embebidos
- ▣ Conta a complexidade do algoritmo (inversão de matrizes, descodificação de cadeias de bits, etc)

Boeing

- ▣ Sistemas de tempo real e produtos de engenharia
- ▣ Conta as capacidades de controlo e função
 - ▣ Dimensão funcional: n° operações internas
 - ▣ Dimensão controlo: n° transições de estado

Métricas OO

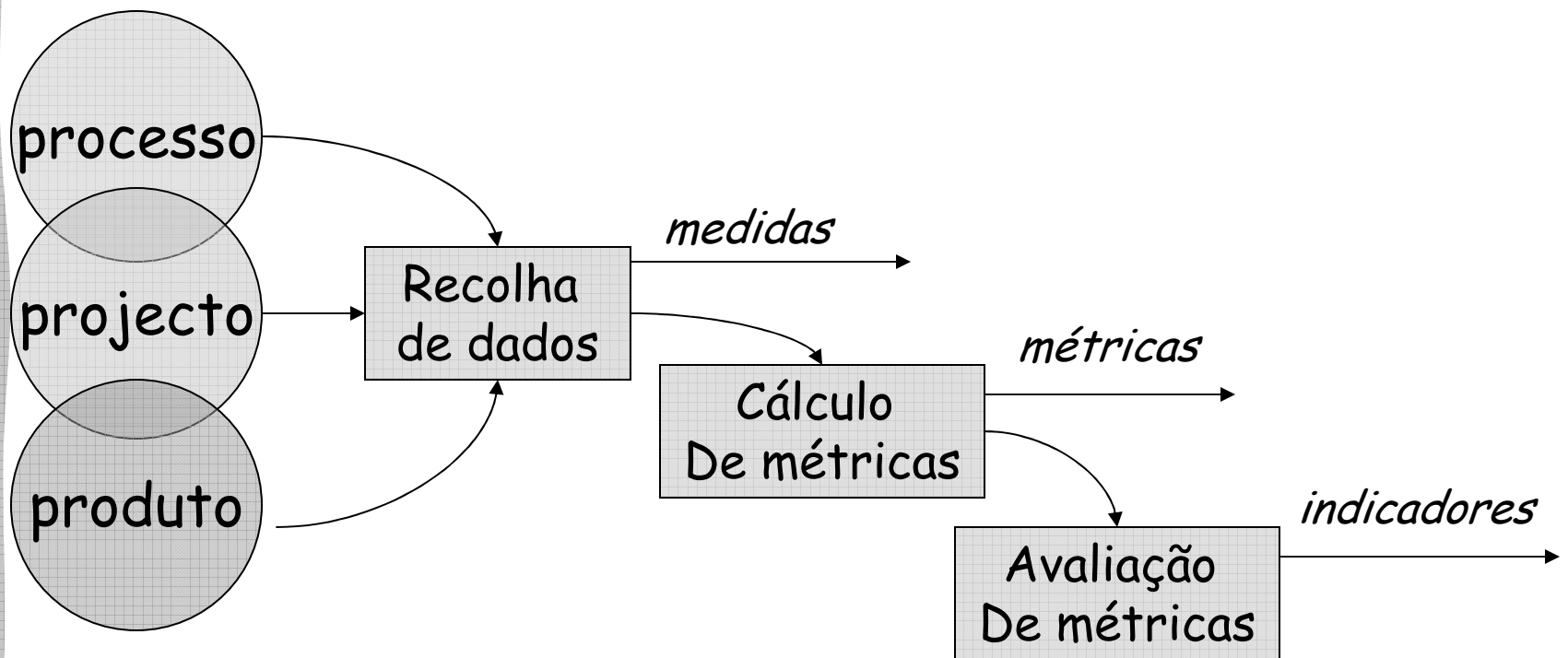
- # ObjectMetrix define e conta um conjunto standard de elementos de UML, nomeadamente:
 - ▣ use cases
 - ▣ classes
 - ▣ subsystems
 - ▣ components
 - ▣ interfaces
 - ▣ web pages
 - ▣ scripts.
- # These are the scope classifiers (types of scope element) that we use to measure the effort required for software development

Medidas de qualidade

- # Correção
- # Facilidade de manutenção
- # Integridade
- # Facilidade de uso
- # Eficácia na eliminação de defeitos

Integração das métricas no processo de eng. De software

Se não medimos, como sabemos quando melhoramos?



Exemplo de métrica para organizações pequenas

- # Objectivo: redução do tempo na avaliação e implementação solicitações de câmbio
- # Métricas:
 - ▣ Tempo para a completção do pedido
 - ▣ Esforço (horas-pessoa) para a avaliação
 - ▣ Tempo entre a avaliação e a atribuição da ordem de câmbio ao pessoal
 - ▣ Esforço (horas-pessoa) para a realização do câmbio
 - ▣ Erros descobertos durante o trabalho
 - ▣ Defeitos descobertos depois da entrega

Estabelecimento de um programa de métricas - Passos

1. Identificar os objectivos do negócio
2. Identificar o que se deseja saber ou aprender)
3. Identificar sub-objectivos
4. Identificar entidades e atributos associados
5. Formalizar objectivos da medição
6. Identificar perguntas que possam quantificar-se e os indicadores associados
7. Identificar os elementos de dados a recolher
8. Definir as medidas a usar e fazê-las operativas
9. Indicar acções para melhorar as medidas
10. Preparar um plano de implementação das medidas

Sucesso nas métricas - Tips

- # Start small
- # Explain why
- # Share data
- # Define data items and procedures
- # Understand trends

Traps

1. Lack of Management Commitment
2. Measuring Too Much, Too Soon
3. Measuring Too Little, Too Late
4. Measuring the Wrong Things
5. Imprecise Metrics Definitions
6. Using Metrics Data to Evaluate Individuals
7. Using Metrics to Motivate, Rather than to Understand
8. Collecting Data That Is Not Used
9. Lack of Communication and Training
10. Misinterpreting Metrics Data