

Análise e Desenvolvimento de Sistemas

Laboratório de Engenharia de Software

Métricas de Software



Agenda

- Métricas de software
- Exemplos de métricas
- Normalização de métricas
- Elaboração de estimativas
- A técnica de Pontos de Caso de Uso

Agenda

- Métricas de software
- Exemplos de métricas
- Normalização de métricas
- Elaboração de estimativas
- A técnica de Pontos de Caso de Uso

Métricas de software

- Métricas servem para ...

- ❖ Caracterizar algo;
- ❖ Avaliar algo;
- ❖ Predizer algo;
- ❖ Melhorar algo;

No produto, processo e projeto

- Definições básicas

- ❖ **Medida**: fornece uma **indicação quantitativa** da extensão, quantidade, dimensões, capacidade ou tamanho de algum produto ou atributo de um processo;
- ❖ **Métrica**: é uma **medida quantitativa** do grau para o qual um sistema, componente ou processo possui um dado **atributo**;
- ❖ **Indicador**: é uma **métrica** ou **combinação de métricas** que fornece uma posição sobre o processo, projeto ou produto.

Métricas de software

- **Métricas de produto**

- ❖ Concentram-se em **atributos** específicos dos **produtos** de **trabalho** gerados no **processo** de **engenharia** de **software**;
- ❖ **Exemplos de produtos de Engenharia de Software que se aplicam métricas:**
 - Métricas sobre **modelos** de **análise**;
 - Métricas sobre **modelos** de **projeto**: complexidade dos algoritmos, complexidade da arquitetura, complexidade dos fluxos de dados;
 - Métricas sobre **código**;
 - Métricas sobre a **eficácia** do **processo** no **produto**: eficiência na remoção de erros.

Métricas de software

- **Métricas de produto**

- ❖ **Exemplos de métricas concretas**

- **Erros** não descobertos antes da entrega;
- **Defeitos** do produto que são descobertos pelo cliente;
- **Produtos** que são gerados no processo;
- **Esforço** humano empregado no desenvolvimento de um produto;
- **Tempo** gasto no desenvolvimento de um produto;
- Conformidade do tempo de desenvolvimento do produto com **cronogramas**.

Métricas de software

- Métricas relacionadas à qualidade do produto
 - **Correção**: o grau que um **programa** opera de **acordo** com sua **especificação**. Uma métrica comum para correção é a relação **defeitos/KLOC**;
 - **Manutenibilidade**: o grau de **facilidade** de se **alterar** um **programa**. Uma métrica simples para manutenibilidade é o **tempo médio para modificação**:

$$MTTC = t_{\text{analisar modificação}} + t_{\text{projetar a modificação}} + t_{\text{implementar a modificação}} + t_{\text{testar a modificação}} + t_{\text{distribuir a modificação}}$$

Integridade: o grau que um **programa** **resiste** a **ataques** **externos**:

$$integridade = \sum_{i \in \text{TipoAtaque}} ((1 - ameaca_i) \times (1 - seguranca_i))$$

Métricas de software

- **Métricas relacionadas à qualidade do produto**
 - **Usabilidade:** representa o **grau** que um programa é **fácil** de usar. **Métricas úteis:**
 - **Aptidão física** e/ou intelectual necessária para aprender a lidar com o sistema;
 - **Tempo** necessário para se **tornar** o **usuário** moderadamente **eficiente** no uso do sistema;
 - **Aumento líquido de produtividade;**
 - **Medidas subjetivas de atitudes** dos usuários.
 - **Eficiência na remoção de defeitos:** permite descobrir se os erros são cada vez mais eliminados em etapas anteriores do processo:

$$DRE_i = \frac{E_i}{E_i + E_{i+1}}$$

Métricas de software

■ Métricas de processo

- ❖ Focam a **qualidade alcançada** como uma consequência de um processo que é aplicado de forma repetível ou gerenciável
- ❖ Empregam **qualidade estatística** assegurada do software, com a categorização e análise dos erros encontrados;
- ❖ O objetivo principal é **promover a eficiência na remoção de defeitos**, evitando a propagação de erros de uma fase a outra do projeto;
- ❖ **Reutilizam métricas** de processos aplicados anteriormente para **predizer comportamentos futuros**.

Agenda

- Métricas de software
- Exemplos de métricas
- Normalização de métricas
- Elaboração de estimativas
- A técnica de Pontos de Caso de Uso

Exemplos de métricas

- **Métricas de projeto**

- ❖ Medem **esforço/tempo** por **tarefa** de engenharia de software;
- ❖ **Alguns exemplos de métricas de projeto**
 - Número de **erros** por **hora** de **revisão**;
 - **Diferenças** ente as **datas prevista** e **real** de um cronograma;
 - **Alterações** de **projeto** e suas **características**;
 - **Distribuição** dos **esforços** nas **tarefas** de engenharia de software.

Exemplos de métricas

■ Métricas com base no esforço

- É uma comum **medir** o **esforço** de **trabalho empregado** por uma **pessoa** em um período de tempo (mês, semana, dia, hora);
- Assim, é comum utilizar termos tais como **PESSOA-MÊS (PM)**, **PESSOA-HORA** etc;
- **CUIDADO:** é um termo **multiplicativo** → PESSOA x MÊS!;
- Por **exemplo**, um esforço de **10PM** pode ser o **esforço** realizado por **2 pessoas** em **5 meses** de trabalho ou de **5 pessoas** em **2 meses** de trabalho (respeitando os limites físicos e lógicos, é claro).

Exemplos de métricas

- **Métricas com base no esforço**
 - Exemplos de **métricas úteis** que empregam esforço:
 - **Erros** por **pessoa-mês**;
 - **Linhas de código (LOC)** por **pessoa-mês**;
 - **Custo** por **pessoa-mês**;
 - **Defeitos** por **pessoa-mês**.

Agenda

- Métricas de software
- Exemplos de métricas
- Normalização de métricas
- Elaboração de estimativas
- A técnica de Pontos de Caso de Uso

Normalização de métricas

■ Definição

- ❖ Consiste na **definição** e **uso** de uma **base** para as métricas;
- ❖ São **aplicadas** ao **processo** e ao **produto**.
- ❖ **Tipos de métricas normalizadas**
 - * **Normalização orientada ao tamanho:** utiliza como valor de normalização uma **medida** de **tamanho** do software.
Exemplos:
 - **Linhas de código (LOC);**
 - **Mil linhas de código (KLOC);**
 - * **Normalização orientada a funções:** utiliza como valor de normalização **medida** associada à **funcionalidade** do software. **Exemplos:**
 - **Pontos de função;**
 - **Pontos de caso de uso.**

Normalização de métricas

■ Métricas orientadas a tamanho

- ❖ Dependem de **estimativa** do **tamanho** do software;
- ❖ Quando de está no **início** do **projeto**, esta **estimativa** é bem grosseira...
- ❖ Embora **simples** de **entender** (linha de código), **não é justa: compare** um mesmo **programa** em **assembly** com um **programa** em **Java**...
- ❖ Mas as **métricas** (e a **qualidade** do **software**) são imediatamente **reconhecidas**. **Exemplos**
 - **Erros** por **KLOC**;
 - **Defeitos** por **KLOC**;
 - **Custo** por **KLOC**;
 - **Páginas** de **documentação** por **KLOC**.

Normalização de métricas

- **Métricas orientadas a função**

- ❖ São **preferíveis** do que as métricas orientadas a tamanho, **pois**:

- São **independentes** de uma **linguagem de programação**;
 - **Utilizam dados mensuráveis** a partir do **domínio de informação do problema**;
 - São mais **justas** que as **métricas orientadas a tamanho**;
 - Tornam simples a **acomodação da reutilização** de código e orientação a objetos.

- ❖ **Exemplos**

- **Erros por pontos de função**;
 - **Defeitos por pontos de função**;
 - **Custo por pontos de função**;
 - **Páginas de documentação por pontos de função**.

Utilização de métricas

■ Banco de dados de métricas

- ❖ Serve como um **histórico** que armazena a **evolução** das **métricas** nos diversos projetos de uma empresa;
- ❖ Permite **analisar tendências** na **qualidade** do produto, **processo** e **projeto**, bem como realizar **estimativas** para novos projetos;
- ❖ **Exemplo:**

projeto	LOC	dur.	S(000)	pp.doc	erros	defeitos	pessoas
alpha	12,100	24	168	365	134	29	3
beta	27,200	62	440	1224	321	86	5
gamma	20,200	43	314	1050	256	64	6
⋮	⋮	⋮	⋮	⋮			

Agenda

- Métricas de software
- Exemplos de métricas
- Normalização de métricas
- **Elaboração de estimativas**
- A técnica de Pontos de Caso de Uso

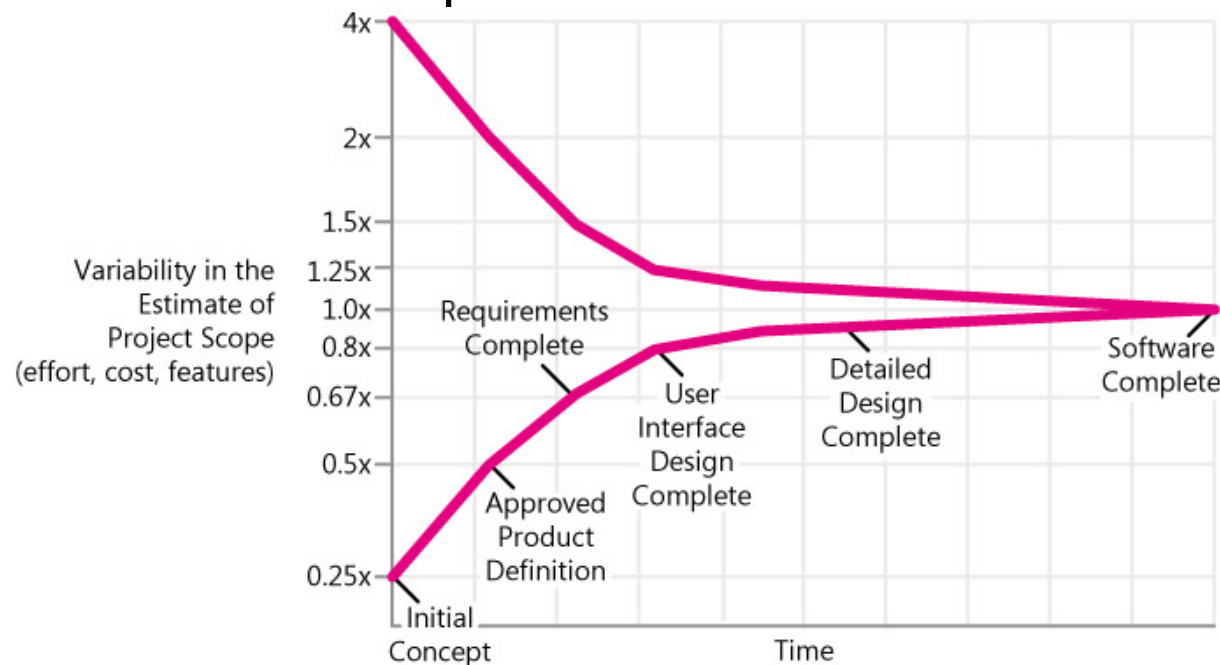
Elaboração de estimativas

- **Estimativas de recursos do projeto**
 - ❖ **Pessoas:** capacidade, especialidade estimativa de carga de trabalho;
 - ❖ **Software:** disponibilidade de componentes;
 - ❖ **Ambiente de Software:** compiladores, editores, diagramadores, sistemas de configuração, ferramentas de acompanhamento, documentação, recursos de hardware, suporte de rede.
 - ❖ **Para realizar estimativas é necessário:**
 - O **escopo** do projeto deve estar **definido**;
 - A **decomposição** das **tarefas** de **engenharia** de **software** e **funções** do **sistema** deve estar pronta;
 - **Métricas** de **projetos** **passados**;
 - Utilizar **mais** de uma **técnica** de **estimativa**.

Elaboração de estimativas

■ Um pouco sobre estimativas

- ❖ Independente do sistema de normalização adotado, **quando não se tem um produto é necessário estimá-lo**;
- ❖ As **estimativas não são fixas**: as **iniciais são grosseiras** e, com o **passar do tempo**, são **ajustadas com valores mais realistas**;
- ❖ Este fato é sintetizado pelo **Cone da Incerteza**:



Elaboração de estimativas

■ Técnicas

- ❖ **Experiência** em projetos passados;
- ❖ Técnicas tipo ***bottom-up***:
 - **Decomposição de tarefas** e estimativas de esforço;
 - **Estimativas por tamanho** (por função ou LOC);
- ❖ Técnicas tipo ***top-down***:
 - **Modelos empíricos** ($estimativa = f(parâmetros)$).
- ❖ **Ferramentas** específicas.

Elaboração de estimativas

- **Estimativas por decomposição de tarefas**
 - ❖ A partir do **escopo** determinar todas as **funções** do **software** e hierarquizá-las;
 - ❖ Construir a matriz **Produto x Processo**:

Conteúdo das células:

O **esforço** em **PM** necessário para completar a tarefa;

Recursos a serem utilizados;

Datas de início e de término das atividades;

Produtos a serem elaborados;

Consequências de cada tarefa.

ATIVIDADES COMUNS DO PROCESSO	Comunicação	Planejamento	Modelagem	Construção	Entrega
Tarefas de Engenharia de Software					
Funções do produto					
Entrada de texto					
Formatação e edição					
Edição automática					
Layout de página					
Indexação automática e TOC					
Gerenciamento de arquivos					
Produção de documentos					

Início: 23/01/2005
Término: 30/01/2005
Custo: R\$1000,00
Recursos: 1PM

Elaboração de estimativas

- Estimativas por tamanho do software

- ❖ Dependem da **estimativa por função** ou da **estimativa de LOC**, bem como de **dados históricos** do projeto;
- ❖ **Exemplo:** o **número de pessoas alocadas** em um **projeto** pode ser **estimada** conhecendo-se um histórico de **LOC/PM**, bem como tendo-se uma estimativa de **duração** e **LOC** para o novo projeto:

$$\text{numPessoas} = LOC \div (\text{Duracao} * (LOC/PM))$$

Elaboração de estimativas

■ Estimativas por modelos empíricos

- ❖ Modelos empíricos seguem normalmente uma fórmula do tipo:

$$E = A + B \times (est)^C$$

- ❖ Onde ***E*** é o esforço em **pessoas-mês**, ***est*** é uma estimativa (LOC ou por função) e ***A***, ***B*** e ***C*** são **parâmetros do modelo**:

Para modelos baseados em LOC, os mais conhecidos na literatura são:

- $E = 5.2 \times (KLOC)^{0.91}$ modelo de Walston-Felix;
- $E = 5.2 + 0.73 \times (KLOC)^{1.16}$ modelo de Bailey-Basili;
- $E = 3.2 \times (KLOC)^{1.05}$ modelo simples de Boehm;
- $E = 5.288 \times (KLOC)^{1.047}$ modelo de Doty para $KLOC > 9$;

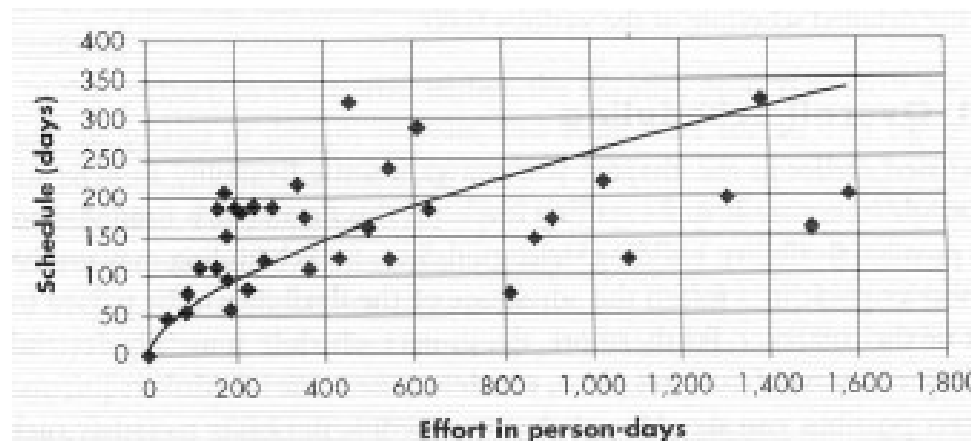
E para estimativas baseadas em FP temos os seguintes modelos:

- $E = -13.39 + 0.0545 \times FP$ modelo de Albrecht-Gaffney;
- $E = 60.62 \times 7.728 \times 10^{-8} \times FP^3$ modelo de Kemerer;
- $E = 585.7 + 15.12 \times FP$ modelo de Mattison, Barnett e Mellichamp;

Elaboração de estimativas

■ Estimativas por modelos empíricos

- ❖ **Conhecido** ou fixando-se o esforço (em **PM**), pode-se estimar o tempo (em meses);
- ❖ Um **método** a ser utilizado é a **regressão não-linear** dos dados de **cronograma x esforço** de projetos passados:



❖

- ❖ **Regra prática:**

$$TEMPO = \sqrt{ESFORÇO}$$

Elaboração de estimativas

- **Regras práticas para estimativas**
 - **Distribuições típicas para cronograma e esforço durante um processo de engenharia de software:**
 - **Cronograma:**
 - 20% projeto alto nível;
 - 20% projeto detalhado;
 - 40% construção;
 - 20% integração e testes;
 - **Esforço:**
 - Projeto, construção e integração/testes, na proporção 1:2:1;
 - **Comparação de estimativas**
 - Fazer 3 estimativas: otimista, provável e pessimista e então ponderar esses resultados para se obter uma estimativa mais sólida:

$$EV = (S_{otimista} + 4S_{provavel} + S_{pessimista})/6$$

Agenda

- Métricas de software
- Exemplos de métricas
- Normalização de métricas
- Elaboração de estimativas
- **A técnica de Pontos de Caso de Uso**

A técnica de Pontos de Caso de Uso

■ Conceitos

- Técnica desenvolvida por **Gustav Karner** em **1993** e que serve para **estimar o tamanho de sistemas de software orientados a objeto** em que os **requisitos do sistema** foram **baseados em casos de uso**;
- A ideia é **dimensionar o software** a partir de sua **funcionalidade esperada**, modelada pelos **casos de uso**;
- O **resultado é apenas um número – pontos de caso de uso** – que **quanto maior for, maior será a dificuldade de construir o software** e, **consequentemente**, demandará **mais esforço e custo maior**;
- **Neste método**, ao se obter o **número de pontos de caso de uso**, pode-se obter, por **fórmula empírica**, o **esforço estimado**.

A técnica de Pontos de Caso de Uso

■ **Resumo do método**

- Calcular Unadjusted Use Case Weight (**UUCW**);
- Calcular Unadjusted Actor Weight (**UAW**);
- Calcular Technical Complexity Factor (**TCF**);
- Calcular Environmental Complexity Factor (**ECF**);
- Calcular Unadjusted Use Case Point:(**UUCP**): $UUCP = UUCW + UAW$
- Finalmente, calcular **Use Case Point (UCP)**: $UCP = UUCP \times TCF \times ECF$;
- Por fim, pode-se obter por meio de fórmula empírica uma estimativa do esforço.

A técnica de Pontos de Caso de Uso

- **Calcular Unadjusted Use Case Weight (UUCW)**
 - Expressa a **complexidade** dos **casos de uso** do sistema;
 - **Calcular** o valor do **peso de caso de uso não ajustado (UUCW)**, com a **tabela** a seguir:

Tipo de caso de uso	Descrição	Peso	Número de casos	Peso*Número de casos
Simples	De 1 a 3 transações <u>ou</u> número de classes ≤ 5 no software	5		
Médio	De 4 a 7 transações <u>ou</u> número de classes ≥ 6 e ≤ 10 no software	10		
Complexo	8 ou mais transações <u>ou</u> número de classes ≥ 11 no software	15		
UUCW (soma das multiplicações):				

A técnica de Pontos de Caso de Uso

- **Calcular Unadjusted Use Case Weight (UUCW)**
 - O que **significa** uma **transação** de **caso de uso** neste método?
 - Normalmente **não é** o mesmo que um **passo enumerado** de um **caso de uso**;
 - Deve ser **entendido como** “uma entrada que o ator produziu no sistema e a reação que o sistema produziu em resposta”;
 - **Exemplos**

(1) The user selects an X and Ys and submits. } Transaction 1
 (2) The system searches for hits and shows the results.

(1) The user selects an X. } Transaction 1
 (2) The system shows the Y s connected with this X.
 (3) The user also selects one or more Y s and submits. } Transaction 2
 (4) The system searches for hits and shows the results.

https://www.ibm.com/developerworks/rational/library/edge/09/mar09/collaris_dekker/

A técnica de Pontos de Caso de Uso

- **Calcular Unadjusted Actor Weight (UAW)**
 - Expressa a **complexidade** dos **atores** do sistema;
 - **Calcular** o valor do **peso** de ator não ajustado (**UAW**), com a **tabela** a seguir:

Tipo de ator	Descrição	Peso	Número de atores	Peso*Número de atores
Simples	O ator é uma interface de sistema externo (API)	1		
Médio	O ator é uma interface dirigida por um protocolo	2		
Complexo	O ator é um humano interagindo com uma interface (GUI)	3		
UAW (soma das multiplicações):				

A técnica de Pontos de Caso de Uso

- **Calcular Technical Complexity Factor (TCF)**
 - Expressa a complexidade dos **fatores técnicos** do sistema;
 - **Primeiro**, calcular o **fator técnico (TF)** na tabela a seguir:

Fator técnico	Descrição	Peso	Complexidade	Peso*Complexidade
T1	Sistema distribuído	2		
T2	Eficiência	1		
T3	Eficiência para o usuário	1		
T4	Complexidade interna de processamento	1		
T5	Reutilização	1		
T6	Facilidade de instalação	0,5		
T7	Facilidade de utilização	0,5		
T8	Portabilidade	2		
T9	Facilidade de modificar	1		
T10	Concorrência	1		
T11	Características especiais de segurança	1		
T12	Provê acesso para terceiros	1		
T13	É necessário treinamento especial para o usuário	1		
Fator total (TF) (soma das multiplicações):				

Valores inteiros no intervalo [0..5]
0 = não contribui;
5 = muito importante

A técnica de Pontos de Caso de Uso

- **Calcular Technical Complexity Factor (TCF)**
 - Depois, **calcular o TCF** assim:

$$\text{TCF} = 0.6 + (0.01 * \text{TF})$$

A técnica de Pontos de Caso de Uso

- **Calcular Environmental Complexity Factor (ECF)**
 - Expressa a complexidade do **ambiente** de desenvolvimento do sistema;
 - **Primeiro**, calcular o **fator de ambiente (EF)** na tabela a seguir:

Fator do ambiente	Descrição	Peso	Impacto	Peso*Impacto
E1	Familiaridade com UML	1,5		
E2	Experiência com a aplicação	0,5		
E3	Experiência com orientação a objetos	1		
E4	Capacidade do analista líder	0,5		
E5	Motivação	1		
E6	Requisitos estáveis	2		
E7	Trabalhadores em tempo parcial	-1		
E8	Dificuldade na linguagem de programação	-1		
Fator total (EF) (soma das multiplicações):				

Valores inteiros no intervalo [0..5]
0 = não contribui;
5 = muito importante

A técnica de Pontos de Caso de Uso

- **Calcular Environmental Complexity Factor (ECF)**
 - Depois, calcular o **ECF** assim:

$$\text{ECF} = 1.4 + (-0.03 * \text{EF})$$

A técnica de Pontos de Caso de Uso

- **Cálculos finais**

- Calcular **Unadjusted Use Case Point (UUCP)** – pontos de caso de uso não ajustados:

$$\text{UUCP} = \text{UUCW} + \text{UAW}$$

- Calcular **Use Case Point (UCP)** – pontos de caso de uso – este é o valor desejado!

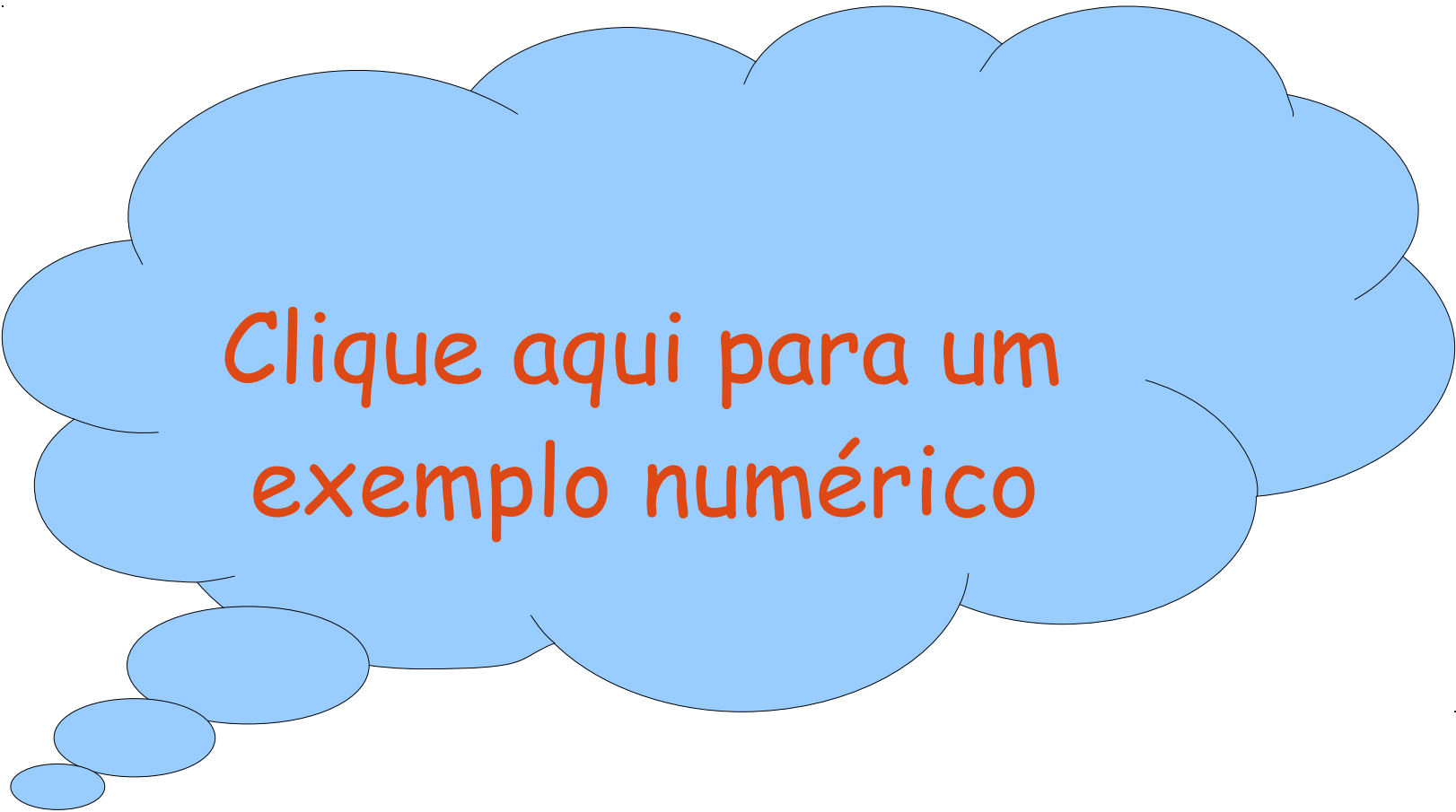
$$\text{UCP} = \text{UUCP} \times \text{TCF} \times \text{ECF}$$

A técnica de Pontos de Caso de Uso

- **Estimativa de esforço com pontos de caso de uso**
 - Determinar o **fator de produtividade (PF)**:
 - Contar quantos **fatores de E1 à E6 (slide 31) estão abaixo de 3** e quantos **fatores de E7 à E8 (slide 31) que estão acima de 3** – **somar essas duas contagens**:
 - **Se o total for 2 ou menos** utilizar **PF = 20** horas-homem/UCP;
 - **Senão, se for 3 ou 4**, utilizar **PF = 28** horas-homem/UCP;
 - **Senão, se for maior que 5**, **reestruturar o projeto** – alto **risco de fracassar!**
 - **Finalmente, aplique a fórmula:**

Homem-horas estimadas para o projeto = UCP*PF

A técnica de Pontos de Caso de Uso



**Clique aqui para um
exemplo numérico**

Referências

- JALOTE, P. **Software Project Management in Practice**. [S.l.]: Addison Wesley, 2002. Capítulo 1.
- McCONNELL, S. **Software Estimation: Demystifying the Black Art**. Microsoft Press, 2006.
- PRESSMAN, R. **Engenharia de Software**. 5. ed. [S.l.]: McGraw-Hill Interamericana, 2002. Capítulo 3.
- PFLEEGER, S. L. **Engenharia de Software: Teoria e Prática**. São Paulo, SP : Pearson/Prentice Hall, 2004. Capítulos 2 e 3.