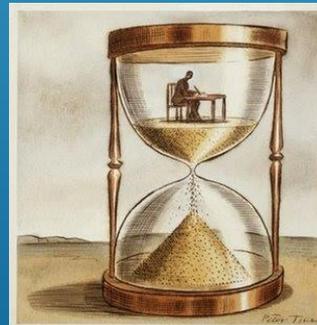


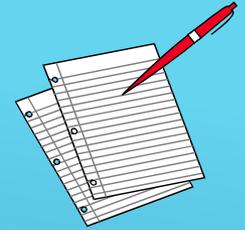
Planejamento de Projeto de Software: Estimativas de Esforço e Custo

Engenharia de Software

Rosana T. V. Braga
ICMC/USP



PLANO DE PROJETO DE SOFTWARE



I. Introdução

1. Escopo e propósito do documento
2. Objetivos do projeto

II. Organização de projeto

1. Organização da equipe e papéis envolvidos

III. Riscos do Projeto

1. Descrição dos riscos
2. Estratégias de redução dos riscos

IV. Recursos de software e hardware

1. Descrição dos recursos
2. Custos relacionados

V. Divisão do trabalho

(atividades do projeto, milestones e resultados de cada atividade)

VI. Cronograma

(dependência entre atividades, pessoas envolvidas, tempo para cada milestone)

VII. Mecanismos de Monitoração

VIII. Relatórios

Estimativas



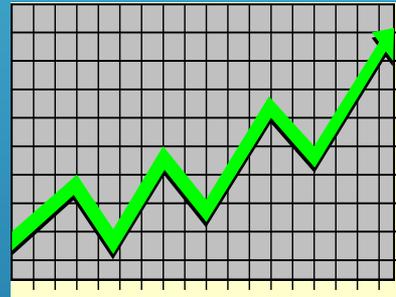
Nós oferecemos três tipos de serviço
BOM - BARATO - RÁPIDO
Você pode escolher dois:
Serviço BOM e BARATO não vai ser RÁPIDO
Serviço BOM e RÁPIDO não vai ser BARATO
Serviço RÁPIDO e BARATO não vai ser BOM

GERENCIAMENTO DE PROJETOS
Escolha Dois

ESTIMATIVAS

Necessidade de estimar quanto esforço ou horas de trabalho serão necessárias

Incertezas!!



ESTIMATIVAS DE SOFTWARE

- ▶ Exige experiência
- ▶ Acesso a boas informações históricas (métricas)
- ▶ Coragem de empenhar em previsões quantitativas, quando informação qualitativa é tudo que existe
- ▶ Estimar tem risco inerente
 - ▶ Esse risco leva à incerteza



O QUE DEVEMOS ESTIMAR?



1. Tamanho do produto

- ▶ Quantidade de software a ser produzida
- ▶ Ex. no. linhas de código, no. pontos de função, n.o de requisitos, pontos de casos de uso

2. Esforço

- ▶ Derivado da estimativa de tamanho
- ▶ Ex. dividindo a estimativa de tamanho por produtividade produz-se o esforço

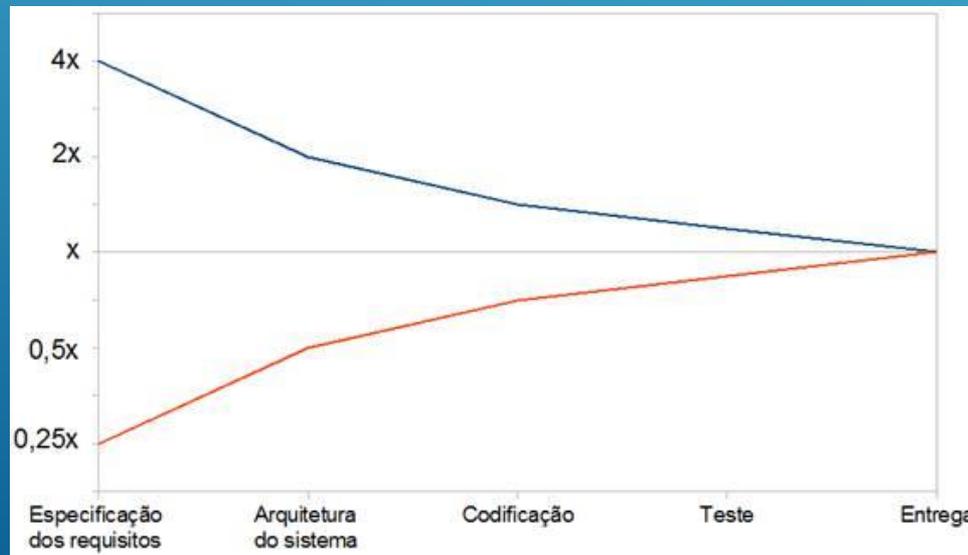
3. Prazo

- ▶ Geralmente são dirigidos a datas fornecidas pelo Cliente

OPÇÕES PARA ESTIMATIVAS

Adiar a estimativa até que o projeto esteja mais adiantado

Estimativas 100% precisas são obtidas quando o projeto estiver finalizado!



OPÇÕES PARA ESTIMATIVAS

1. Usar um ou mais modelos para estimar o esforço do projeto
 1. **Modelos não paramétricos**

define um tempo x para o projeto
 2. **Modelos paramétricos**
 - a. Pontos de função
 - b. Casos de uso
 - c. Histórias
2. Usar informação de projetos anteriores
 1. **Métricas de software**
 - a. medidas quantitativas de projetos finalizados



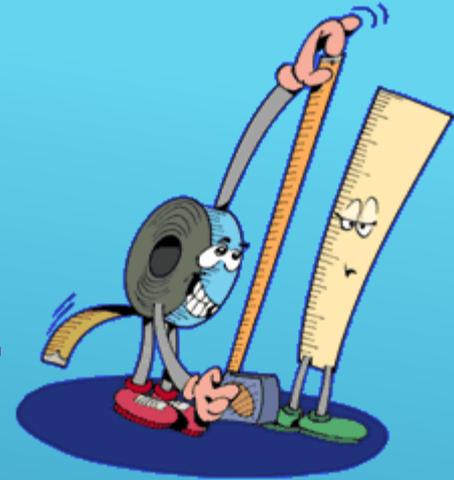
MÉTRICAS DE SOFTWARE

▶ *“Dois times de projeto de software diferentes registram todos os erros encontrados durante o processo de engenharia de software. **Time A** encontrou **342** erros durante o processo e o **Time B** encontrou **184** erros. Qual time foi mais eficaz para descobrir erros durante o processo de desenvolvimento?”*

Necessidade de conhecer o tamanho e complexidade dos projetos. Se as medidas são **normalizadas** é possível criar métricas de software que permitem comparar amplamente os projetos.

MÉTRICAS DE SOFTWARE

MEDIDAS DO SOFTWARE



MEDIDAS DIRETAS

- Custo
- Esforço (Pessoas-Mês)
- Linhas de Código
- Número de Erros
- Velocidade de processamento

MEDIDAS INDIRETAS

- Corretitude
- Manutenibilidade
- Integridade
- Usabilidade

PREVISÕES PODEM SER COMPLEXAS!!!



ESTIMATIVAS MAIS PROVÁVEIS...

- ▶ **Modelagem algorítmica de custo**
 - ▶ Informação históricas para previsão
- ▶ **Julgamento de especialistas**
 - ▶ Cada um estima o custo
- ▶ **Estimativa por analogia**
 - ▶ Considera projetos similares
- ▶ **Lei de Parkinson**
 - ▶ *O trabalho se expande para preencher o tempo disponível*
- ▶ **Atribuição de preço a ganhar**

TÉCNICAS PARA ESTIMATIVAS

- Baseada em linha de código
- Baseada em pontos de casos de uso
- Baseada em pontos de história



ESTIMATIVA BASEADA EM LINHA DE CÓDIGO

- Técnica mais antiga para estimativa
- Utiliza LOC ou KLOC
- Adequada para alguns tipos de linguagens de programação

Qual é o principal problema dessa técnica?

ESTIMATIVA BASEADA EM LINHA DE CÓDIGO

- 1) Decompor o software em **funções menores** que possam ser estudadas individualmente.
- 2) Usando **dados históricos** (ou intuição), fornecer para cada subfunção valores de LOC *otimista, mais provável, pessimista*.

Funções	LOC			Esperado
	otimista(a)	mais provável(b)	pessimista(c)	
função1	1800	2400	2650	
função2	4100	5200	7400	
função3	4600	6900	8600	
função4	2950	3400	3600	
função5	4050	4900	6200	
função6	2000	2100	2450	
função7	6600	8500	9800	

ESTIMATIVA BASEADA EM LINHA DE CÓDIGO

3) Determinar o número esperado (**E**) da variável de estimativa para cada subfunção: $E = (a + 4b + c) / 6$

Funções	LOC			Esperado
	otimista(a)	mais provável(b)	pessimista(c)	
função1	1800	2400	2650	2340
função2	4100	5200	7400	5380
função3	4600	6900	8600	6800
função4	2950	3400	3600	3350
função5	4050	4900	6200	4850
função6	2000	2100	2450	2140
função7	6600	8500	9800	8400
			LOC ESTIMADO	33360

4) Determinar o valor estimado **LOC ESTIMADO**.

ESTIMATIVA BASEADA EM LINHA DE CÓDIGO

Determinação do **Esforço** e do **Custo** :

- De projetos passados (dados históricos) obtém-se:

Produtividade Média = 3.206,86 LOC/pessoa-mês

Custo Médio = 0,30 \$/LOC

- Da última tabela obtém-se LOC ESTIMADO = 33.360

ESFORÇO = LOC ESTIMADO / Produtividade Média

ESFORÇO = 33.360 / 3.206,86 = 10,4 pessoa-mês

CUSTO = LOC ESTIMADO x Custo Médio

CUSTO = 33.360 x 0,30 = 10.008 \$

MÉTRICAS

MÉTRICAS ORIENTADAS À FUNÇÃO

São derivadas de medidas indiretas do software e do processo através do qual ele é desenvolvido

Exemplo: **PF** - Pontos por Função

(Albrecht 1979)

MÉTRICAS

MÉTRICA ORIENTADA À FUNÇÃO - PF

Concentra-se na funcionalidade ou utilidade do software

Os PFs são derivados usando uma relação empírica baseada em medidas do domínio de informação e da complexidade do software

MÉTRICAS

MÉTRICA ORIENTADA À FUNÇÃO - PF

PONTOS POR FUNÇÃO É APLICADO ATRAVÉS DE 3 PASSOS:

1) Completar a seguinte tabela:

Parâmetro	Contagem		fator de ponderação			
			Simplex	Médio	Complexo	
nro de entradas do usuário	<input type="text"/>	x	3	4	6	<input type="text"/>
nro de saídas do usuário	<input type="text"/>	x	4	5	7	<input type="text"/>
nro de consultas do usuário	<input type="text"/>	x	3	4	6	<input type="text"/>
nro de arquivos	<input type="text"/>	x	7	10	15	<input type="text"/>
nro de interfaces externas	<input type="text"/>	x	5	7	10	<input type="text"/>
Contagem-Total						<input type="text"/>

MÉTRICAS

MÉTRICA ORIENTADA À FUNÇÃO - PF

2) Responder as questões 1-14, considerando a escala de 0 a 5:



1. O sistema exige backup e recuperação confiáveis?
2. É requerida comunicação de dados?
3. Existem funções de processamento distribuído?
4. O desempenho é crítico?
5. O sistema funcionará num sistema operacional existente e intensamente utilizado?
6. São requeridas entrada de dados *on-line*?
7. As entradas *on-line* requerem que as transações de entrada sejam construídas com várias telas e operações?
8. Os arquivos são atualizados *on-line*?
9. Entradas, saídas, arquivos e consultas são complexos?
10. O processamento interno é complexo?
11. O código é projetado para ser reusável?
12. A conversão e a instalação estão incluídas no projeto?
13. O sistema é projetado para múltiplas instalações em diferentes organizações?
14. A aplicação é projetada de forma a facilitar mudanças e o uso pelo usuário?

MÉTRICAS

MÉTRICA ORIENTADA À FUNÇÃO - PF

3) Ajustar os Pontos por Função de acordo com a complexidade do sistema, através da seguinte fórmula:

$$PF = \text{Contagem-Total} \times \left(0,65 + 0,01 \times \sum_{i=1}^{14} (F_i) \right)$$

F_i = valores de ajuste da complexidade das perguntas 1-14

**MÉTRICAS
DERIVADAS**



PRODUTIVIDADE = PF / pessoas-mês

QUALIDADE = erros / PF

CUSTO = \$ / PF

DOCUMENTAÇÃO = pags.docum. / PF

MÉTRICAS

MÉTRICAS ORIENTADAS À FUNÇÃO

VANTAGENS:

- Independentes da linguagem
- Ideal para aplicações que usam linguagem não procedimental
- Baseados em dados mais fáceis de serem conhecidos durante a evolução do projeto

DESVANTAGENS:

- Cálculo baseado em dados subjetivos
- Não é uma medida direta; é apenas um número

ESTIMATIVA BASEADA EM CASOS DE USO (PCU)

- ▶ Estimativa de custo e de esforço.
 - ▶ Criada em 1993 por Gustav Karner.
 - ▶ Baseada em Pontos de Função.
 - ▶ Utiliza casos de uso.
 - ▶ Simplicidade e facilidade de uso.
 - ▶ Independência de linguagem e de paradigma.
- 

ESTIMATIVA BASEADA EM CASOS DE USO (PCU)

- ▶ Cálculo:
- ▶ Determinar:
 - ▶ Somatório dos Pontos por Caso de Uso (SPCU).
 - ▶ Somatório dos Pontos por Ator (SPA).
- ▶ Determinar:
 - ▶ Fator de Complexidade Técnica (FCT).
 - ▶ Fator Ambiental (FA).
- ▶ Totalizar:
 - ▶ **$PCU = FCT * FA * (SPCU+SPA)$**

SOMATÓRIO DOS PONTOS POR CASO DE USO (SPCU)

- ▶ Classificam-se os casos de uso em:
 - ▶ Simples = 5 Pontos
 - ▶ Até 3 transações
 - ▶ Médio = 10 Pontos
 - ▶ Até 7 transações
 - ▶ Complexo = 15 Pontos
 - ▶ Mais que 7 transações

EXEMPLO TPV

CASO DE USO: COMPRAR ITENS COM DINHEIRO

Seqüência Típica de Eventos

Ação do ator	Resposta do sistema
1. Este caso de uso começa quando o Cliente chega ao TPV com itens para comprar	
2. O Caixa registra o identificador de cada item Se há mais de um do mesmo item, o caixa também entra a quantidade	3. Determina o preço do item e adiciona informação sobre o item à transação de venda corrente A descrição e o preço do item são apresentados
4. Quando termina a entrada dos itens, o Caixa indica ao TPV que as entradas estão completas	5. Calcula e apresenta o total da venda
6. O Caixa informa o total ao cliente	
7. O Cliente entrega o pagamento em dinheiro – o “pagamento em dinheiro” – possivelmente maior que o total da venda	
8. O Caixa registra a quantidade de dinheiro recebida	9. Exibe o valor do troco a ser devolvido ao cliente
10. O Caixa deposita o dinheiro recebido e retira o troco devido O Caixa entrega ao cliente o troco e o recibo impresso	11. Registra a venda completada (<i>logs</i>)
12. O Cliente sai com os itens comprados	

Seqüências alternativas:

- **Linha 2:** Identificador de item inválido digitado. Indicar o erro.
- **Linha 7:** O Cliente não tem dinheiro suficiente. Cancelar a transação de venda.

14 Transações

Mais que 7 transações



Caso de Uso Complexo

15 Pontos

SOMATÓRIO DOS PONTOS POR ATOR (SPA)

- ▶ Classificam-se os atores em:
 - ▶ Simples = 1 Ponto
 - ▶ Ator acessa o sistema por meio de um outro sistema, por uso de uma API
 - ▶ Médio = 2 Pontos
 - ▶ Ator acessa o sistema por meio de uma interface texto
 - ▶ Complexo = 3 Pontos
 - ▶ Ator acessa o sistema por meio de uma interface gráfica

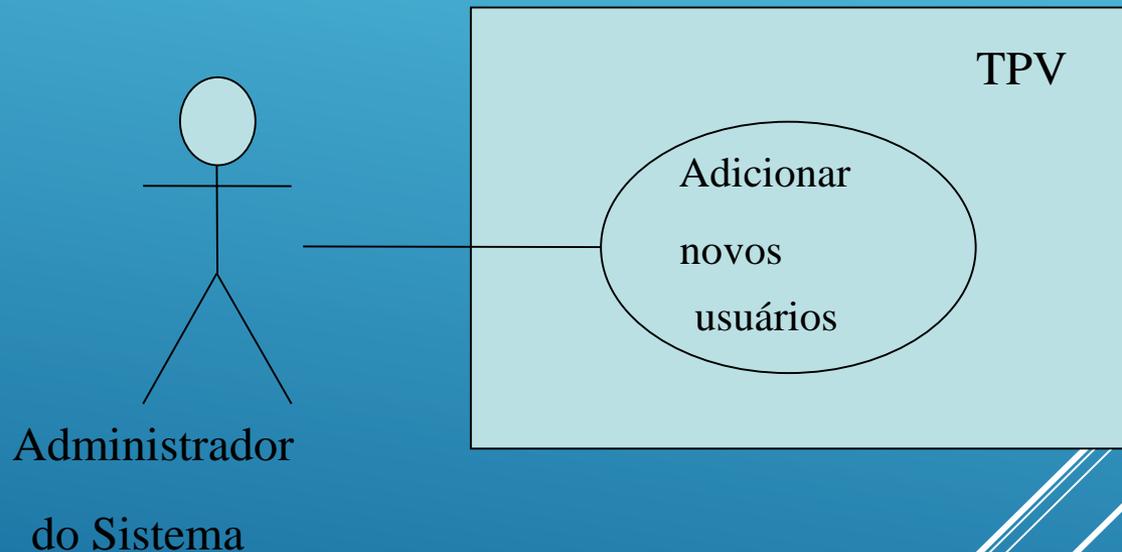
EXEMPLO TPV

ATOR: ADMINISTRADOR DO SISTEMA

Interface Gráfica

Ator Complexo

3 Pontos



CÁLCULO DO FATOR DE COMPLEXIDADE TÉCNICA (FCT)

- ▶ A partir da tabela de FCT, calcular a influência de cada fator.
 - ▶ A influência de cada fator pode variar de 0 a 5
 - ▶ Irrelevante a Essencial

Tabela de Fatores de Complexidade Técnica



Fatores relativos a requisitos não-funcionais

Fator de Complexidade Técnica	Descrição	Peso
F1	Sistema distribuído	2
F2	Tempo de Resposta	1
F3	Eficiência	1
F4	Processamento complexo	1
F5	Código reusável	1
F6	Facilidade de instalação	0,5
F7	Facilidade de uso	0,5
F8	Portabilidade	2
F9	Facilidade de mudança	1
F10	Concorrência	1
F11	Recursos de segurança	1
F12	Acessível por terceiros	1
F13	Requer treinamento especial	1

EXEMPLO TPV

CÁLCULO DO FATOR DE COMPLEXIDADE TÉCNICA (FCT)

Fator de Complexidade Técnica	Descrição	Peso	Influência	TOTAL
F1	Sistema distribuído	2	4	8
F2	Tempo de Resposta	1	5	5
F3	Eficiência	1	3	3
F4	Processamento complexo	1	3	3
F5	Código reusável	1	3	3
F6	Facilidade de instalação	0,5	3	1,5
F7	Facilidade de uso	0,5	4	2
F8	Portabilidade	2	3	6
F9	Facilidade de mudança	1	3	3
F10	Concorrência	1	3	3
F11	Recursos de segurança	1	3	3
F12	Acessível por terceiros	1	3	3
F13	Requer treinamento especial	1	3	3

CÁLCULO DO FATOR DE COMPLEXIDADE TÉCNICA (FCT)

- ▶ A partir da tabela de FCT, calcular a influência de cada fator.
- ▶ Utilizar a fórmula abaixo para cálculo do FCT:

$$FCT = C_1 + C_2 \sum_{i=1}^{13} F_i \text{Peso} ; \quad C_1 = 0,6 \quad C_2 = 0,01$$

Tabela de Fatores de Complexidade Técnica

Fator de Complexidade Técnica	Descrição	Peso
F1	Sistema distribuído	2
F2	Tempo de Resposta	1
F3	Eficiência	1
F4	Processamento complexo	1
F5	Código reusável	1
F6	Facilidade de instalação	0,5
F7	Facilidade de uso	0,5
F8	Portabilidade	2
F9	Facilidade de mudança	1
F10	Concorrência	1
F11	Recursos de segurança	1
F12	Acessível por terceiros	1
F13	Requer treinamento especial	1

EXEMPLO TPV

CÁLCULO DO FATOR DE COMPLEXIDADE TÉCNICA (FCT)

Fator de Complexidade Técnica	Descrição	Peso	Influência	TOTAL
F1	Sistema distribuído	2	4	8
F2	Tempo de Resposta	1	5	5
F3	Eficiência	1	3	3
F4	Processamento complexo	1	3	3
F5	Código reusável	1	3	3
F6	Facilidade de instalação	0,5	3	1,5
F7	Facilidade de uso	0,5	4	2
F8	Portabilidade	2	3	6
F9	Facilidade de mudança	1	3	3
F10	Concorrência	1	3	3
F11	Recursos de segurança	1	3	3
F12	Acessível por terceiros	1	3	3
F13	Requer treinamento especial	1	3	3

$$FCT = 0.6 + 0.01 * 46.5$$

$$FCT = 1.065$$

CÁLCULO DO FATOR AMBIENTAL (FA)

- ▶ A partir da tabela de FA, calcular a influência de cada fator.
 - ▶ A influência de cada fator pode variar de 0 a 5.
 - ▶ Irrelevante a Essencial

Tabela de Fatores Ambientais



Fatores relativos ao nível de competência da equipe

Fator Ambiental	Descrição	Peso
F1	Familiaridade com o processo de desenvolvimento.	1,5
F2	Desenvolvedores em meio expediente.	-1
F3	Presença de analistas experientes	0,5
F4	Experiência com a aplicação em desenvolvimento.	0,5
F5	Experiência em Orientação a Objetos.	1
F6	Motivação	1
F7	Dificuldade com a linguagem de programação	-1
F8	Requisitos estáveis	2

EXEMPLO TPV

CÁLCULO DO FATOR AMBIENTAL (FA)

Fator Ambiental	Descrição	Peso	Influência	Total
F1	Familiaridade com o processo de desenvolvimento.	1,5	3	4,5
F2	Desenvolvedores em meio expediente.	-1	3	-3
F3	Presença de analistas experientes	0,5	3	1,5
F4	Experiência com a aplicação em desenvolvimento.	0,5	5	2,5
F5	Experiência em Orientação a Objetos.	1	4	4
F6	Motivação	1	3	3
F7	Dificuldade com a linguagem de programação	-1	3	-3
F8	Requisitos estáveis	2	3	6

15,5

CÁLCULO DO FATOR AMBIENTAL (FA)

- ▶ A partir da tabela de FA, calcular a influência de cada fator.
- ▶ Utilizar a fórmula abaixo para cálculo do FA.

$$FA = C_1 + C_2 \sum_{i=1}^8 F_i \text{Peso}; \quad C_1 = 1,4 \quad C_2 = -0,03$$

Tabela de Fatores Ambientais

Fator Ambiental	Descrição	Peso
F1	Familiaridade com o processo de desenvolvimento.	1,5
F2	Desenvolvedores em meio expediente.	-1
F3	Presença de analistas experientes	0,5
F4	Experiência com a aplicação em desenvolvimento.	0,5
F5	Experiência em Orientação a Objetos.	1
F6	Motivação	1
F7	Dificuldade com a linguagem de programação	-1
F8	Requisitos estáveis	2

EXEMPLO TPV

CÁLCULO DO FATOR AMBIENTAL (FA)

Fator Ambiental	Descrição	Peso	Influência	Total
F1	Familiaridade com o processo de desenvolvimento.	1,5	3	4,5
F2	Desenvolvedores em meio expediente.	-1	3	-3
F3	Presença de analistas experientes	0,5	3	1,5
F4	Experiência com a aplicação em desenvolvimento.	0,5	5	2,5
F5	Experiência em Orientação a Objetos.	1	4	4
F6	Motivação	1	3	3
F7	Dificuldade com a linguagem de programação	-1	3	-3
F8	Requisitos estáveis	2	3	6

$$FA = 1,4 - 0.03 * 15,5$$

$$FA = 0,935$$

15,5

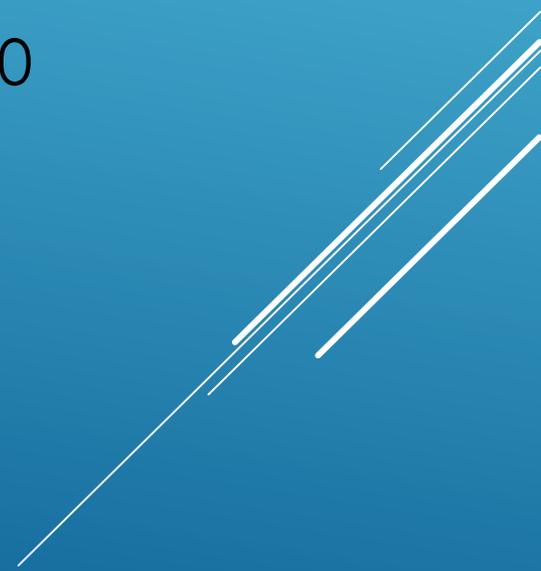
CÁLCULO DOS PONTOS DE CASO DE USO AJUSTADOS

- $PCU = FCT * FA * (SPCU + SPA)$
- $PCU = 1,065 * 0,935 * (15 + 3)$
- **$PCU = 17,895$**

Cálculo do esforço:

- $E = PCU * IP$ (índice de produtividade)
- $E = 17,895 * 15$
- **$E = 268,4$ horas**

Cálculo do esforço

- ▶ Índice de produtividade:
 - ▶ Baseado em projetos anteriores
 - ▶ Quantidade e horas (média) para um PCU
 - ▶ Karner sugere 20 horas por PCU
 - ▶ Outros autores sugerem entre 15 e 30
- 

EXEMPLO DE UM SISTEMA SIMBÓLICO

SOMATÓRIO DOS PONTOS POR CASO DE USO (SPCU)

- ▶ 4 Casos de Uso: um simples (5), um médio (10) e dois complexos (30).
 - ▶ Total de Pontos (SPCU) = 45 Pontos
 - ▶ 5 Atores: dois simples (2), um médio (2) e dois complexos (6).
 - ▶ Total dos Pontos (SPA) = 10 Pontos
 - ▶ FCT = 1,065
 - ▶ FA = 0,935
- 

Exemplo de um Sistema Simbólico

Somatório dos Pontos por Caso de Uso (SPCU)

- ▶ Cálculo:

- ▶ $SPCU = 45$

- ▶ $SPA = 10$

- ▶ $FCT = 1,065$

- ▶ $FA = 0,935$

- ▶ $PCU = FCT * FA * (SPCU + SA)$

- ▶ $PCU = 1,065 * 0,935 * (45 + 10)$

- ▶ **PCU = 58,575 Pontos de Caso de Uso**

- ▶ *Quantas horas? Qual o custo?*

EXEMPLO DE UM SISTEMA SIMBÓLICO

SOMATÓRIO DOS PONTOS POR CASO DE USO (SPCU)

- ▶ **PCU = 58,575 Pontos**
- ▶ Karner sugere 20 horas/ponto
- ▶ $\text{Tempo} = 58,575 * 20$
- ▶ **Tempo = 1.171,5 horas**
- ▶ Hora de estágio R\$ 8,00
- ▶ **Custo R\$ 9.372,00**

EXEMPLO DE UM SISTEMA SIMBÓLICO

SOMATÓRIO DOS PONTOS POR CASO DE USO (SPCU)

- ▶ **PCU = 58,575 Pontos**
- ▶ Karner sugere 20 horas/ponto
- ▶ $\text{Tempo} = 58,575 * 20$
- ▶ **Tempo = 1.171,5 horas**
- ▶ Hora de estágio R\$ 8,00
- ▶ **Custo R\$ 9.372,00**
- ▶ **Um sistema de 4 Casos de Uso e 5 Atores!!!**

EXEMPLO DE UM SISTEMA SIMBÓLICO

SOMATÓRIO DOS PONTOS POR CASO DE USO (SPCU)

- ▶ **PCU = 58,575 Pontos**
- ▶ Karner sugere 20 horas/ponto
- ▶ $\text{Tempo} = 58,575 * 20$
- ▶ **Tempo = 1.171,5 horas**
- ▶ Hora de estágio R\$ 8,00
- ▶ **Custo R\$ 9.372,00**
- ▶ **Um sistema de 4 Casos de Uso e 5 Atores!!!**



Suas transações eram complexas!!!

EXEMPLO DE UM SISTEMA SIMBÓLICO

SOMATÓRIO DOS PONTOS POR CASO DE USO (SPCU)

- ▶ PCU = **58,575** Pontos
- ▶ **Outros trabalhos sugerem:
de 5 a 7 horas/ponto**
- ▶ Tempo = $58,575 * 6$
- ▶ Tempo = 351,45 horas
- ▶ Hora de estágio R\$ 8,00
- ▶ Custo R\$ 2.811,60
- ▶ Um sistema de 4 Casos de Uso e 5 Atores!
 - ▶ Mais realista! 😊

ESTIMATIVA BASEADA EM PONTOS DE HISTÓRIA

- ▶ Utilizada pelos métodos ágeis XP e Scrum
- ▶ Medida de esforço relativa à equipe de desenvolvimento
- ▶ Feita pela equipe:



Quanto tempo n pessoas levariam para terminar uma história?

ESTIMATIVA BASEADA EM PONTOS DE HISTÓRIA (PH)

- ▶ Um ponto de história é o esforço de desenvolvimento de uma pessoa durante um dia ideal de trabalho.
 - ▶ Pessoa dedicada a história, trabalhando de 6 a 8 horas
- ▶ EX:
 - ▶ Quanto tempo 3 pessoas levariam para terminar uma história?
 - ▶ R: 3 pessoas levariam 4 dias
 - ▶ $3 \times 4 = 12 \text{ PH}$

ESTIMATIVA BASEADA EM PONTOS DE HISTÓRIA (PH)

- ▶ PH é feito por comparação entre histórias:
 - ▶ Separa a lista de histórias por nível de dificuldade e define o esforço de **maneira relativa**
 - ▶ Para o ser humano é mais natural fazer medidas relativas do que absolutas
 - ▶ Utiliza a série de Fibonacci adaptada: 1, 2, 5, 8, 15, 25, 40 ...
 - ▶ Ordem de grandeza natural para o **esforço** e não uma medida exata
 - ▶ Pode utilizar outras medidas como “camiseta”:
 - ▶ Pequeno, médio e grande ...
- 

ESTIMATIVA BASEADA EM PONTOS DE HISTÓRIA

▶ Atribuição de PH segue critérios subjetivos:

▶ Complexidade:

“Essa regra de negócios tem muitos cenários possíveis”

▶ Esforço:

“Essa alteração é simples, mas precisa ser realizada em muitas telas”

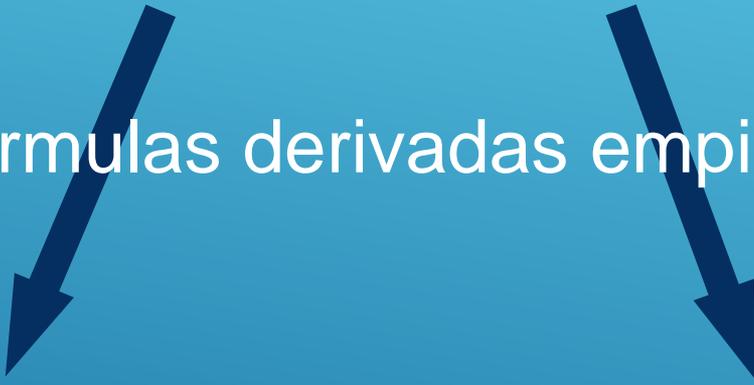
▶ Risco:

“Precisamos utilizar o framework XX mas ninguém tem experiência”

ESTIMATIVAS

MODELOS EMPÍRICOS

Usam fórmulas derivadas empiricamente



**Modelo Estático de
Variável Simples**

COCOMO

ESTIMATIVAS

MODELOS EMPÍRICOS

Modelo Estático de Variável Simples

$$\text{RECURSO} = C_1 X (\text{característica estimada})^{C_2}$$

ESFORÇO $E = 5.2 \times \text{LOC}^{0.91}$ (pessoas-mês)

DURAÇÃO DO PROJETO $D = 4.1 \times \text{LOC}^{0.36}$ (meses)

TAMANHO DA EQUIPE $S = 0.54 \times E^{0.06}$ (pessoas)

LINHAS DE DOCUMENTAÇÃO $\text{DOC} = 49 \times \text{LOC}^{1.01}$

Modelo de Walston e Felix - constantes derivadas de 60 projetos

ESTIMATIVAS

COCOMO

MODELOS EMPÍRICOS

▶ **Modelo 1:** Modelo COCOMO **Básico**

(Boehm)

- ▶ modelo estático de variável simples
- ▶ esforço de desenvolvimento calculado em função do tamanho do software (LOC)

▶ **Modelo 2:** Modelo COCOMO **Intermediário**

- ▶ esforço de desenvolvimento calculado em função do tamanho do software (LOC) e de um conjunto de "direcionadores de custo"

▶ **Modelo 3:** Modelo COCOMO **Avançado**

- ▶ mesmas características do modelo intermediário
- ▶ avaliação do impacto dos "direcionadores de custo" em cada passo do processo de construção

ESTIMATIVAS

MODELOS EMPÍRICOS

COCOMO

São definidos para 3 classes de projetos:

▶ **Orgânico**

- ▶ projetos pequenos
- ▶ equipes pequenas e com baixa experiência
- ▶ requisitos não muito rígidos

▶ **Semi-Separado**

- ▶ projetos com tamanho e complexidade médios
- ▶ equipes com experiências variadas
- ▶ requisitos rígidos e não rígidos

▶ **Embutido**

- ▶ restrições rígidas de hardware, software e operacionais

ESTIMATIVAS

MODELOS EMPÍRICOS

COCOMO

- Modelo COCOMO **Básico**

Esforço $E = A (KLOC)^B$ (pessoas-mês)

Tempo de Desenvolvimento $T = C (E)^D$ (meses)

- Modelo COCOMO **Intermediário**

Esforço $E = A (LOC)^B \times FAE$ (pessoas-mês)

	Básico				Intermediário	
classes	A	B	C	D	A	B
orgânico	2.4	1.05	2.5	0.38	3.2	1.05
semi-separado	3.0	1.12	2.5	0.35	3.0	1.12
embutido	3.6	1.20	2.5	0.32	2.8	1.20

ESTIMATIVAS

MODELOS EMPÍRICOS

COCOMO

FAE - Fator de Ajuste do Esforço

ATRIBUTOS DIRECIONADORES DE CUSTO

- Atributos do Produto: complexidade, confiabilidade exigida, tamanho do banco de dados
- Atributos do Hardware: restrições de desempenho, restrições de memória, etc.
- Atributos Pessoais: capacidade, experiência
- Atributos de projeto: uso de ferramentas, métodos, etc.

Cada atributo é ponderado numa escala de 6 pontos e, através de tabelas publicadas por Boehm, obtem-se o FAE, que varia de 0.9 a 1.14

ESTIMATIVAS

MODELOS EMPÍRICOS

COCOMO

- Exemplo de aplicação do COCOMO

Utilizando-se os dados obtidos através da Estimativa LOC, o Modelo Básico e Semi-separado, tem-se:

$$E = A (KLOC)^B$$

$$\begin{aligned} E &= 3.0 (KLOC)^{1,12} \\ &= 3.0 (33.3)^{1,12} \\ &= 152 \text{ pessoas-mes} \end{aligned}$$

$$T = C (E)^D$$

$$\begin{aligned} T &= 2.5 (E)^{0.35} \\ &= 2.5 (152)^{0.35} \\ &= 14.5 \text{ meses} \end{aligned}$$

Com esses valores é possível determinar um número recomendado de pessoas

$$\begin{aligned} N &= E/T \\ &= 152/14.5 \\ &= 11 \text{ pessoas} \end{aligned}$$

PONTOS-CHAVE

- ▶ Estimativas:
 - ▶ Não constituem uma ciência exata; sempre existem riscos.
 - ▶ Para diminuir os riscos, devem ser baseadas em dados históricos, que são construídos ao longo do tempo através da utilização de métricas.
 - ▶ Estimativas mais precisas devem fazer uso de várias técnicas.

