

# Engenharia de Software

## Seção 8 – Estimativas de Software

# Objetivos

O aluno deverá reconhecer os conceitos e técnicas para estimativas de tamanho de Software e de esforço para seu desenvolvimento.



# Coisas que não acontecem...

- ✓ “O software é muito mais complexo do que pensávamos.”
- ✓ “Nossa equipe não interagiu como esperávamos.”
- ✓ “A produtividade nesta linguagem é diferente daquela outra.”
- ✓ “O esforço para desenvolver nessa tecnologia é muito maior do que para desenvolver naquela outra.”

# Planejamento

- ✓ O Planejamento do desenvolvimento é uma atividade que ocorre no início do projeto de desenvolvimento do sistema (ou do software).
- ✓ O Planejamento é fortemente dependente do tamanho do software a ser desenvolvido.



# WBS

O **WBS** (Work Breakdown Structure) depende muito do tamanho do software, pois:

- ✓ O WBS determina a forma como o trabalho é estruturado.
- ✓ O WBS utiliza o esforço estimado para o desenvolvimento, dividindo-o ao longo de etapas e entregáveis.
- ✓ Estimativa de esforço incorreta leva a um WBS incorreto.

# Cronograma

O cronograma organiza, no tempo, a execução das atividades para o desenvolvimento. Ele depende das estimativas de esforço para o desenvolvimento, pois:

- ✓ Os esforços de desenvolvimento influem na duração as tarefas determinadas pelo WBS.
- ✓ A duração das tarefas e o esforço previsto podem determinar qual processo de desenvolvimento será utilizado.
- ✓ A organização das tarefas, com seus esforços e duração, determina qual (ou quais) é o caminho crítico do desenvolvimento.
- ✓ Estimativas incorretas levam a uma visão inadequada do cronograma.

# Orçamento e Aquisições

O orçamento do projeto é diretamente dependente do esforço necessário para o desenvolvimento, pois:

- ✓ Os esforços de desenvolvimento determinam os custos de mão-de-obra.
- ✓ A estimativa de esforços define a permanência de mais ou menos pessoas no projeto, e consequentemente o custo.
- ✓ A estimativa de esforços determina o custo de aquisições de recursos materiais (e.g. computadores) para o desenvolvimento.
- ✓ Estimativas incorretas levam a um orçamento inadequado.

# Plano de Recursos Humanos

O Plano de Recursos Humanos define qual será a equipe de projeto, quantos e quais profissionais serão necessários, e quais serão os treinamentos exigidos. Ele depende das estimativas de esforço para o desenvolvimento, pois:

- ✓ A distribuição de esforços determina quantas pessoas serão necessárias.



# Plano de Recursos Humanos

(continuação)

- ✓ A distribuição das tarefas (decorrente da estimativa de esforços e equipe) define que partes do sistema cada pessoa deverá desenvolver, e com quem deverá se comunicar.
- ✓ Estimativas incorretas levarão a um dimensionamento de equipe inadequado.




# Estimativas

- ✓ O software é usualmente o item mais caro de qualquer sistema baseado em computador.
- ✓ Para sistemas complexos e customizados, erros em estimativas, como vimos, podem ser a diferença entre lucro e prejuízo.
- ✓ A estimativa de custo e esforço do desenvolvimento de software nunca será uma ciência exata, pois há muitos fatores – humanos, técnicos, ambientais, políticos – que o afetam.

# Estimativas (continuação)

- ✓ Para tentar obter estimativas confiáveis de custo e esforço, algumas opções aparecem:
  - ✓ Atrasar a estimativa o máximo possível;
  - ✓ Basear a estimativa em projetos anteriores que já tenham sido completados;
  - ✓ Usar técnicas simples de decomposição do projeto;
  - ✓ Usar um ou mais modelos empíricos.

# Estimativas (continuação)

- ✓ Atrasar a estimativa:
    - ✓ Normalmente não aceitável, pois a estimativa de esforço e custo é necessária no início do projeto;
  - ✓ Basear a estimativa em projetos anteriores:
    - ✓ Só é útil se os projetos anteriores possuem características muito semelhantes às do projeto sendo estimado.
- 

# Estimativas (continuação)

- ✓ Técnicas de Decomposição e modelos empíricos:
  - ✓ São opções viáveis para estimativa de esforço e custo;
  - ✓ Se possível, devem ser aplicadas mais de uma técnica, para efeitos de confirmação.
  - ✓ Cada uma delas é tão boa quanto os dados históricos utilizados.



# Técnicas de Decomposição

- ✓ Técnicas baseadas na divisão do desenvolvimento do software em pedaços menores. Estes pedaços podem ser:
  - ✓ Partes do problema: analisa-se o software:
    - ✓ Estimativas de linhas de código (LOC);
    - ✓ Estimativas de pontos por função (FP);
  - ✓ Partes do processo: analisam-se as etapas do desenvolvimento.

# Estimativa com base no problema

1. A partir de uma descrição do software a ser desenvolvido, é feita uma decomposição funcional.
2. São aplicadas estimativas (FP ou LOC) para cada função.
3. Métricas de produtividade históricas (FP/pm ou LOC/pm) são aplicadas.
4. As estimativas para cada função são então combinadas para obter a estimativa para todo o projeto.

# Estimativa com base no problema (continuação)

6. Preferencialmente são levantadas três estimativas:
  - a) Otimista: usualmente tem os menores valores de esforço;
  - b) Mais provável.
  - c) Pessimista: com os piores valores.

7. É utilizada a formula abaixo para definir o estorço final estimado:

$$S = (S_{ot} + 4 * S_{mp} + S_{pe})/6$$



# Estimativa por LOC

- ✓ Nesta forma de estimativas, são atribuídas quantidades de linhas de código para cada função identificada do sistema.
- ✓ O sistema precisa ser particionado com alto nível de detalhe, pois uma decomposição com baixa granularidade leva a estimativas superficiais e usualmente inferiores às reais.

# Exemplo

Software simples de registro de horas de trabalho.

Função	Otim	+ Prov	Pess	Final
Interface com Usuário	4000	5000	6000	<b>5000</b>
Cadastro de usuário	1200	2000	2300	<b>1920</b>
Cadastro de projetos	1500	2200	3200	<b>2250</b>
Registro de horas de trabalho	2500	3500	4300	<b>3470</b>
Relatórios	1200	1600	1800	<b>1570</b>
<b>Total</b>	<b>10400</b>	<b>14300</b>	<b>17600</b>	<b>14210</b>

Considerando uma produtividade média de 650 LOC/pm, o esforço para o desenvolvimento será de cerca de 22 pessoas-mês.

# Estimativa por FP

- ✓ Pontos por Função é uma métrica de software que é derivada utilizando-se uma relação empírica baseada em duas medidas:
  - ✓ **Valores do domínio da informação:** é realizada uma contagem direta de existência de determinados elementos no software, ponderada por uma avaliação de complexidade.
  - ✓ **Aplicação de Fatores de Ajuste:** realizada por meio da somatória de medida de aplicabilidade de características adicionais.
- ✓ Estes valores são então aplicados à fórmula:

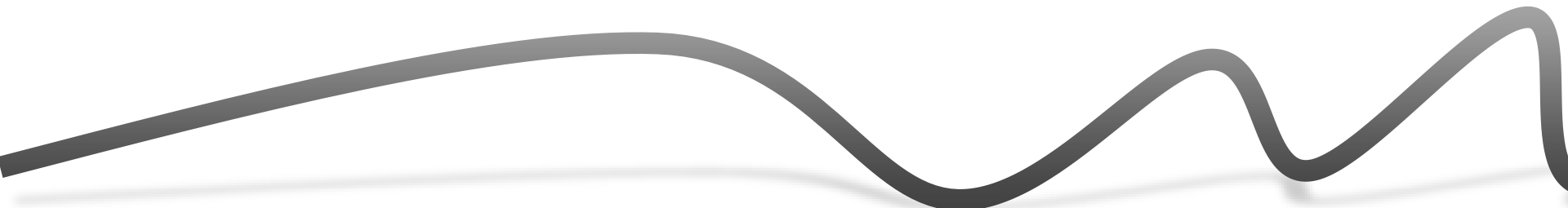
$$\text{FP} = \text{contagem} \times (0,65 + 0,01 \times \sum(F_i))$$

# Domínio da Informação

- ✓ **Número de Entradas Externas** (External Inputs – Els):
  - ✓ Entradas que se originam do usuário
  - ✓ Entradas que se originam de outra aplicação
  - ✓ Fornecem dados ou informações de controle
  - ✓ Não são consultas

# Domínio da Informação (continuação)

- ✓ **Número de Saídas Externas** (External Outputs - EOs):
  - ✓ Derivada de dentro da aplicação
  - ✓ Fornece informação para o usuário
  - ✓ Exemplos: Relatórios, telas, mensagens de erro, etc.
  - ✓ Itens de dados individuais dentro de um relatório não são contados separadamente.



# Domínio da Informação

(continuação)

- ✓ **Número de Consultas Externas** (External Inquiries - EQs):
  - ✓ Entrada online que resulta na geração de alguma resposta imediata do software sob a forma de uma saída online.
- ✓ **Número de Arquivos Lógicos Internos** (Internal Logical Files - ILFs):
  - ✓ Agrupamento lógico de dados que reside dentro das fronteiras da aplicação e é mantido por entradas externas.

# Domínio da Informação

(continuação)

- ✓ **Número de Arquivos de Interface Externa**  
(External Interface Files - EIFs):
- ✓ Agrupamento lógico de dados que reside externamente à aplicação mas fornece dados que podem ser úteis para a aplicação.

# Domínio da Informação

(continuação)

- ✓ Após feitas as contagens, estes valores são multiplicados por pesos, dependendo de sua complexidade.
- ✓ Cada organização determina seus próprios critérios para determinar a complexidade de cada valor.

Valor	Pesos		
	Simples	Média	Complexa
Entradas Externas	3	4	6
Saídas Externas	4	5	7
Consultas Externas	3	4	6
Arquivos Lógicos Internos	2	10	15
Arquivos de Interface Externa	5	7	10



# Fatores de Ajuste

- ✓ Para a determinação dos Fatores de Ajuste ( $F_i$  na fórmula original) são respondidas várias questões.
- ✓ Estas respostas variam de um valor 0 (não aplicável ou não importante) a 5 (absolutamente essencial).
- ✓ Estas respostas são determinadas de forma empírica.
- ✓ As perguntas a serem respondidas são mostradas a seguir.

# Fatores de Ajuste (continuação)

1. O sistema requer backup e recovery confiáveis?
2. Comunicações de dados especializadas são necessárias para transferir informação para ou da aplicação?
3. Existem funções de processamento distribuído?
4. O desempenho é crítico?

# Fatores de Ajuste (continuação)

- 5. O sistema será executado em um ambiente operacional existente, intensamente utilizado?
- 6. O sistema requer entrada de dados online?
- 7. A entrada de dados online exige que a transação de entrada seja construída por meio de várias telas ou operações?



# Fatores de Ajuste (continuação)

- 8. Os ILFs são atualizados online?
- 9. As entradas, saídas, arquivos ou consultas são complexos?
- 10. O processamento interno é complexo?
- 11. O código é projetado para ser reusado?

# Fatores de Ajuste (continuação)

- 12. A conversão e a instalação estão incluídas no projeto (design)?
- 13. O sistema está projetado para instalações múltiplas em diferentes organizações?
- 14. A aplicação está projetada para facilitar mudança e para facilidade de uso pelo usuário?

# Exemplo simples (continuação)

Valor	Contagem / Pesos			Produto
	Simples	Média	Complexa	
Entradas Externas	2/3	1/4	0/6	<b>10</b>
Saídas Externas	2/4	1/5	1/7	<b>20</b>
Consultas Externas	0/3	0/4	0/6	<b>0</b>
Arquivos Lógicos Internos	2/2	1/10	0/15	<b>14</b>
Arquivos de Interface Externa	0/5	0/7	0/10	<b>0</b>
<b>Total</b>	<b>18</b>	<b>19</b>	<b>7</b>	<b>44</b>

✓ Somatória de Fatores de Ajuste: 12

# Exemplo simples (continuação)

- ✓ **FP = contagem x (0,65 + 0,01 x  $\sum(F_i)$ )**
- ✓ **FP = 44 x (0,65 + 0,01 x 12) = 34**
- ✓ Considerando, por exemplo, que:
  - ✓ 1 FP corresponde a 60 LOC
  - ✓ A produtividade é de 12 FP/pm
- ✓ Este sistema seria feito em:
  - ✓ 3 pessoas-mês
  - ✓ Teria pouco mais de 2000 linhas de código
- ✓ Qual estimativa é mais confiável???

# Estimativas de Processo

- ✓ É a técnica de estimativa mais comum.
- ✓ Baseia-se no processo que será usado.
- ✓ Decompõe-se o processo em um conjunto relativamente pequeno de tarefas e é estimado o esforço para cada uma delas.
- ✓ As funcionalidades também são decompostas, então são definidos o esforço por funcionalidade e por tarefa.



# Exemplo Simples

Função	Engenharia		Construção		Total
	Análise	Projeto	Código	Teste	
Cadastro de Usuário	0,25	0,25	0,5	0,25	<b>1,25</b>
Cadastro de Projetos	0,25	0,25	0,5	0,25	<b>1,25</b>
Registro de Apontamento	0,5	0,5	1,0	0,5	<b>2,5</b>
Relatórios	0,25	0,5	1,5	1,0	<b>3,25</b>
<b>Total</b>	<b>1,25</b>	<b>1,5</b>	<b>3,5</b>	<b>2,0</b>	<b>8,25</b>

- ✓ Esforços em pessoas-mês
- ✓ Havendo poucas concordâncias entre as métricas, novas análises devem ser realizadas.

# Modelos Empíricos

- ✓ Modelos empíricos usam fórmulas derivadas por análise de regressão de um grande número de projetos.
- ✓ Leva a uma fórmula como:

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

- ✓ E é o esforço em pessoas-mês.
- ✓ A, B e C são constantes deduzidas empiricamente.
- ✓  $e_v$  é a quantidade da variável de estimativa (LOC ou FP).

# Exemplos

Nome do modelo	Fórmula
Walston-Felix	$E = 5,2 \times (\text{KLOC})^{0,91}$
Bailey-Basili	$E = 5,5 + 0,73 \times (\text{KLOC})^{1,16}$
Bohem simples	$E = 3,2 \times (\text{KLOC})^{1,05}$
Doty para KLOC > 9	$E = 5,288 \times (\text{KLOC})^{1,047}$
Albrecht e Gaffney	$E = -91,4 + 0,355 \times (\text{FP})$
Kemerer	$E = -37 + 0,96 \times (\text{FP})$
Regressão para pequenos projetos	$E = -12,88 + 0,405 \times (\text{FP})$