



## Disciplina: INF15978 – Engenharia de Software I

Prof.: **Monalessa Perini Barcellos**

(monalessa@inf.ufes.br)

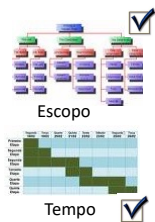
1

## Gerência de Projetos de Software

Antes de iniciar o desenvolvimento de um software, algumas questões são levantadas pelo cliente:



Vou planejar o projeto!



Escopo



2

# Gerência de Projetos de Software

**Gerência de Projetos de Software** envolve, dentre outros, o planejamento e o acompanhamento das **pessoas** envolvidas no projeto, do **produto** sendo desenvolvido e do **processo** seguido para evoluir o software de um conceito preliminar para uma implementação concreta e operacional (PRESSMAN, 2002).

## Ciclo da Gerência de Projetos



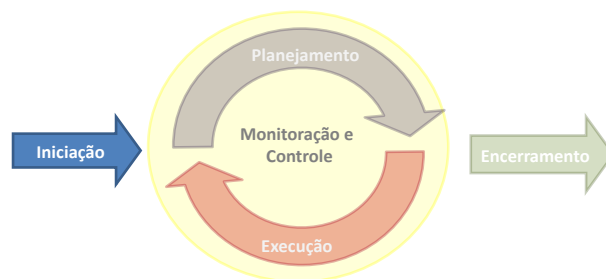
Engenharia de Software

Monalessa Perini Barcellos

3

# Gerência de Projetos de Software

## Ciclo da Gerência de Projetos



Na **Iniciação** é realizada a autorização formal para que o projeto seja iniciado.

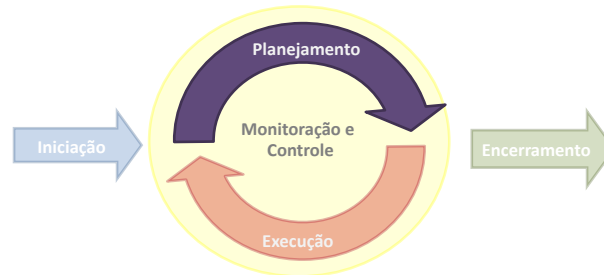
Engenharia de Software

Monalessa Perini Barcellos

4

# Gerência de Projetos de Software

## Ciclo da Gerência de Projetos



No **Planejamento** um plano organizado de como o projeto será conduzido deve ser elaborado.

O planejamento do projeto de software deve tratar fundamentalmente da definição do escopo do software, da definição do processo de software do projeto, da realização de estimativas, da elaboração de um cronograma e da identificação e tratamento dos riscos associados ao projeto.

# Gerência de Projetos de Software

## Ciclo da Gerência de Projetos



Durante a **Execução**, o processo de software planejado para o projeto é executado.

# Gerência de Projetos de Software

## Ciclo da Gerência de Projetos



À medida que o projeto é executado, é realizada sua **Monitoração e Controle**.

É fundamental acompanhar o progresso do trabalho, refinar escopo e estimativas, alterar o processo do projeto e o cronograma, além de monitorar riscos e tomar ações corretivas.

Tipicamente ocorre nos marcos do projeto.

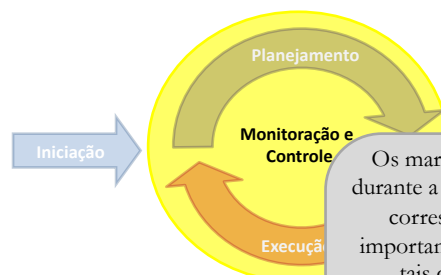
Engenharia de Software

Monalessa Perini Barcellos

7

# Gerência de Projetos de Software

## Ciclo da Gerência de Projetos



À medida que o projeto é executado, é realizada sua **Monitoração e Controle**.

É fundamental acompanhar o progresso do trabalho, refinar escopo e estimativas, alterar o processo do projeto e o cronograma, além de monitorar riscos e tomar ações corretivas.

Tipicamente ocorre nos **marcos** do projeto.

Os marcos de um projeto são estabelecidos durante a definição do processo e normalmente correspondem ao término de atividades importantes do processo de desenvolvimento, tais como Especificação e Análise de Requisitos, Projeto e Implementação. O propósito da definição de um marco é garantir que os interessados tenham uma visão do andamento do projeto e concordem com os rumos a serem tomados.

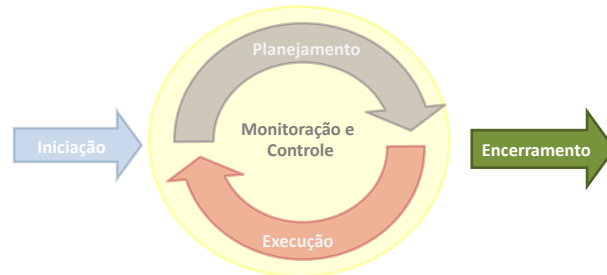
Engenharia de Software

Monalessa Perini Barcellos

8

# Gerência de Projetos de Software

## Ciclo da Gerência de Projetos



No **Encerramento** o projeto é formalmente encerrado.

É realizada uma análise crítica do que deu certo e o que não funcionou, sendo registradas lições aprendidas e oportunidades de melhoria.

Comparações entre valores estimados e realizados, identificação de problemas que ocorreram e causas dos desvios devem ser discutidas com os membros da equipe, visando ao aprendizado organizacional.

Uma técnica bastante empregada neste contexto é a análise *post-mortem*.

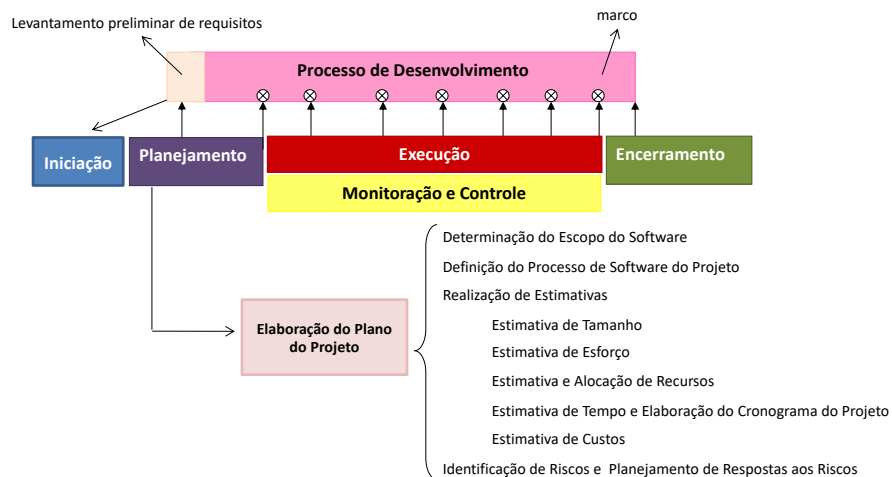
Engenharia de Software

Monalessa Perini Barcellos

9

# Gerência de Projetos de Software

## Processo de Gerência de Projetos integrado ao Processo de Desenvolvimento



Engenharia de Software

Monalessa Perini Barcellos

10

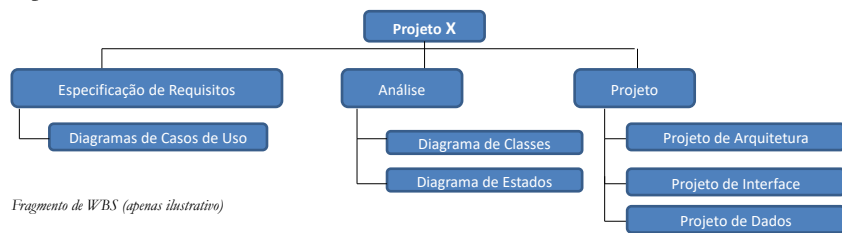
# Gerência de Projetos de Software

## Determinação do Escopo

- Consiste em determinar o que faz e o que não faz parte do projeto.
- Definido considerando duas dimensões: processo e produto.

### Técnicas de apoio à determinação do escopo de um projeto

a) *WBS (Work Breakdown Structure)*: decompõe o trabalho a ser realizado em partes menores, facilitando a realização de estimativas e alocação de recursos.



*Fragmento de WBS (apenas ilustrativo)*

Engenharia de Software

Monalessa Perini Barcellos

11

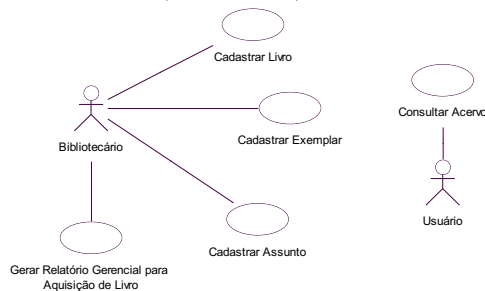
# 9. Gerência de Projetos de Software

## Determinação do Escopo

b) *Diagramas de Casos de Uso*: utilizado para apoiar a especificação de requisitos em projetos de software. Auxilia a determinação do escopo do produto.

Em essência, um diagrama de casos de uso mostra o sistema segundo uma perspectiva externa, na qual atores (usuários ou outros sistemas) aparecem interagindo com as funções do sistema (casos de uso).

*Diagrama de casos de uso preliminar de um sistema hipotético para uma biblioteca.*



Engenharia de Software

Monalessa Perini Barcellos

12

# Gerência de Projetos de Software

## Estimativas

- Antes mesmo de serem iniciadas as atividades técnicas de um projeto, o gerente e a equipe de desenvolvimento devem estimar o trabalho a ser realizado, os recursos necessários, a duração e, por fim, o custo do projeto.
- Apesar das estimativas serem um pouco de arte e um pouco de ciência, essa importante atividade não deve ser conduzida desordenadamente.
- As estimativas podem ser consideradas a fundação para todas as outras atividades de planejamento de projeto.
- Para alcançar boas estimativas de prazo, esforço e custo, existem algumas opções:
  - Postergar as estimativas até o mais tarde possível no projeto.
  - Usar técnicas de decomposição.
  - Usar um ou mais modelos empíricos para estimativas de custo e esforço.
  - Basear as estimativas em projetos similares que já tenham sido concluídos.

Engenharia de Software

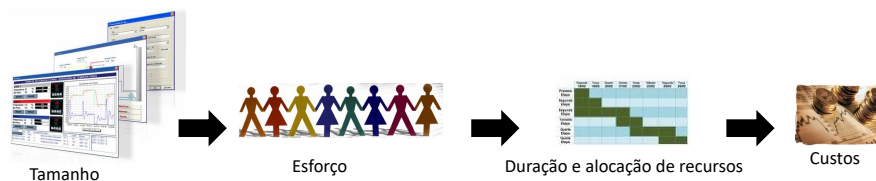
Monalessa Perini Barcellos

13

# Gerência de Projetos de Software

## Estimativas

- Há estimativas para diversos aspectos de um projeto: tamanho, esforço, recursos, tempo e custos.
- Geralmente, a realização de estimativas começa pelas estimativas de tamanho.
- A partir delas, estima-se o esforço necessário e, em seguida, alocam-se os recursos necessários, elabora-se o cronograma do projeto (estimativa de duração) e, por fim, estima-se o custo do projeto.



Engenharia de Software

Monalessa Perini Barcellos

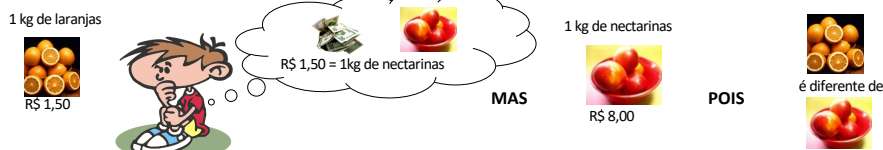
14

# Gerência de Projetos de Software

## Estimativas e Medição de Software

- Para utilizar dados de projetos anteriores como base para a realização de estimativas, é necessário que estes sejam adequados.
- Estabelecer perfis de projetos e coletar algumas medidas nesse contexto pode ser bastante importante para apoiar a realização de estimativas, pois para que dados de projetos anteriores sejam utilizados como base, as características desses projetos devem ser as mesmas (ou próximas) das características do projeto para qual as estimativas são realizadas\*.
- *Exemplo:* se uma organização tem indicadores para *produtividade* (tamanho/esforço) e *custo* (R\$/tamanho) para diversos perfis de projetos diferentes, é possível, a partir de uma estimativa de tamanho, chegar a estimativas de esforço e custo.

**\*Importante: é preciso comparar coisas iguais ou similares.**



Engenharia de Software

Monalissa Perini Barcellos

15

# Gerência de Projetos de Software

## Estimativa de Tamanho

- Estimativas de tamanho tipicamente são utilizadas como base para estimativas de esforço, prazo e custos.
- Também podem ser utilizadas em contratos por preço unitário, onde o preço é estabelecido para uma **unidade de tamanho**.
- Unidades de tamanho de software:
  - ✓ LOC (Lines Of Code)
  - ✓ Pontos de Função
  - ✓ Pontos de Caso de Uso



Engenharia de Software

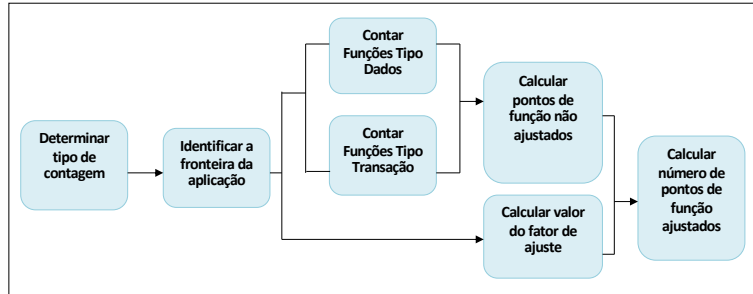
Monalissa Perini Barcellos

16



# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função



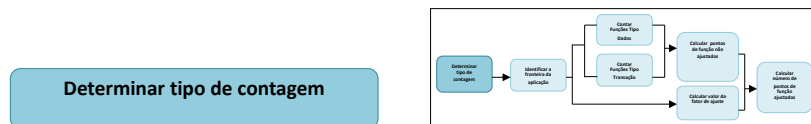
Engenharia de Software

Monalessa Perini Barcellos

17

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função



Tipos de contagem:

- *Projeto de Desenvolvimento*: mede a funcionalidade fornecida aos usuários finais do software para a primeira instalação da aplicação.
- *Projeto de Manutenção*: mede as modificações realizadas para aplicações existentes.
- *Aplicação*: mede uma aplicação instalada.

Engenharia de Software

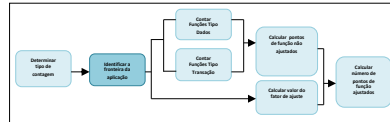
Monalessa Perini Barcellos

18

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Identificar a fronteira da aplicação



Consiste em identificar as fronteiras do software que está sendo medido.

A partir daí torna-se possível definir quais funcionalidades serão incluídas no processo de contagem dos pontos de função.

Engenharia de Software

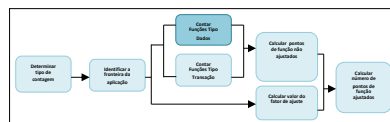
Monalissa Perini Barcellos

19

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Contar funções Tipo Dados



**Funções Tipo Dados** representam as funcionalidades relativas aos **dados** do software. São elas:

- *Arquivo Lógico Interno (ALI)* : grupo logicamente relacionado de dados ou informações de controle, identificável pelo usuário, mantido dentro da fronteira da aplicação que está sendo controlada. Por exemplo: as tabelas ou classes do sistema.
- *Arquivo de Interface Externa (AIE)*: grupo logicamente relacionado de dados ou informações de controle, referenciado pela aplicação, identificável pelo usuário, mantido fora da fronteira da aplicação que está sendo controlada. Por exemplo: as tabelas ou classes acessadas em um outro sistema.

A diferença básica entre um ALI e um AIE é que o último **não** é mantido pela aplicação que está sendo contada.

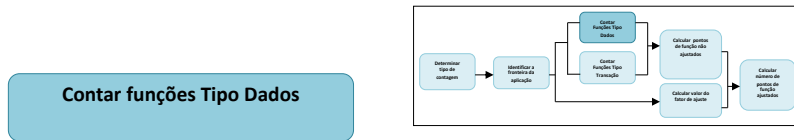
Engenharia de Software

Monalissa Perini Barcellos

20

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função



Cada Arquivo Lógico Interno e cada Arquivo de Interface Externa possui dois tipos de **elementos** que devem ser contados para cada função identificada:

- *Tipos de Elementos de Dados (TED)*: campo único, reconhecido pelo usuário, não recursivo. Por exemplo: campos das tabelas, atributos das classes.
- *Tipos de Elementos de Registros (TER)*: subgrupo de dados, reconhecido pelo usuário. Por exemplo: generalização/especialização de classes.

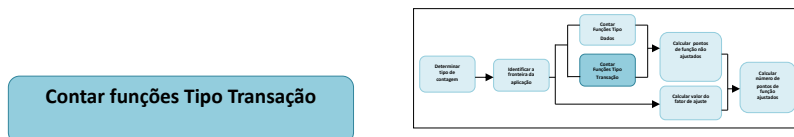
Engenharia de Software

Monalessa Perini Barcellos

21

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função



**Funções Tipo Transação** representam as funcionalidades de **processamento dos dados**. São elas:

- *Entrada Externa (EE)*: processo elementar da aplicação que processa dados ou informações de controle que vêm de fora da fronteira da aplicação que está sendo controlada. Exemplos: validações, fórmulas e cálculos matemáticos cujos parâmetros vêm de fora da fronteira da aplicação.
- *Saída Externa (SE)*: processo elementar da aplicação que gera dados ou informações de controle que são enviados para fora da fronteira da aplicação que está sendo controlada. Exemplos: relatórios e gráficos.
- *Consulta Externa (CE)*: processo elementar da aplicação que representa uma combinação de entrada (solicitação de informação) e saída (recuperação de informação). Exemplos: consultas implícitas, verificação de senhas e recuperação de dados com base em parâmetros.

Engenharia de Software

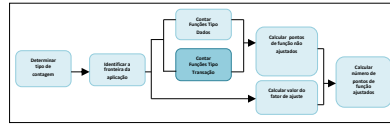
Monalessa Perini Barcellos

22

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

### Contar funções Tipos Transação



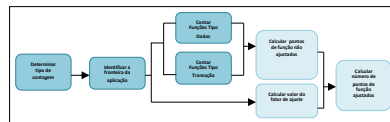
Cada Entrada Externa, Saída Externa e Consulta Externa possui dois tipos de **elementos** que devem ser contados para cada função identificada:

- *Tipos de Elementos de Dados (TED)*: campo único, reconhecido pelo usuário, não recursivo. Por exemplo: campos das tabela, atributos das classes.
- *Tipos de Arquivos Referenciados ou Arquivos Referenciados (TAR)*: arquivos lógicos utilizados para processar a entrada e/ou saída. É o total de ALI e AIE utilizados pela transação.

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Resultado até o momento: tabela abaixo preenchida.

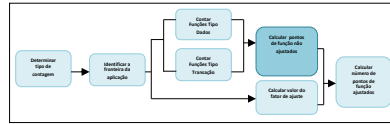


Arquivos Lógicos Internos	Tipos de Elementos de Dados	Tipos de Elementos de Registros
Arquivos de Interface Externa	Tipos de Elementos de Dados	Tipos de Elementos de Registros
Entradas Externas	Tipos de Elementos de Dados	Arquivos Referenciados
Consultas Externas	Tipos de Elementos de Dados	Arquivos Referenciados
Saídas Externas	Tipos de Elementos de Dados	Arquivos Referenciados

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Calcular pontos de função não ajustados



Os **pontos de função não ajustados** refletem especificamente as **funcionalidades** do software fornecidas ao usuário.

Para calculá-los é preciso identificar a complexidade e a contribuição, em pontos por função, de cada uma das funções e elementos contados.

São utilizadas tabelas de complexidade específicas para cada tipo de função.

Complexidade de Arquivos Lógicos Internos e Arquivos de Interface Externa

		Tipos de Elementos de Dados		
		1 a 19	20 a 50	≥ 51
Tipos de Elementos de Registros	1	BAIXA	BAIXA	MÉDIA
	2 a 5	BAIXA	MÉDIA	ALTA
	≥ 6	MÉDIA	ALTA	ALTA

Engenharia de Software

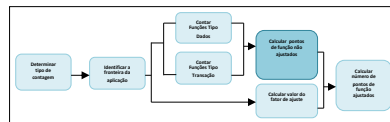
Monalessa Perini Barcellos

25

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Calcular pontos de função não ajustados



O número de pontos de função não ajustados (PFNA) é obtido através do preenchimento da seguinte tabela:

Função	Items Contados por Complexidade	Contribuição	Total por Complexidade	Total de PFNA da Função
ALI	— Baixa	x 7		
	— Média	x 10		
	— Alta	x 15		
AIE	— Baixa	x 5		
	— Média	x 7		
	— Alta	x 10		
EE	— Baixa	x 3		
	— Média	x 4		
	— Alta	x 6		
CE	— Baixa	x 3		
	— Média	x 4		
	— Alta	x 6		
SE	— Baixa	x 4		
	— Média	x 5		
	— Alta	x 7		

Total de Pontos de Função Não Ajustados: \_\_\_\_

Engenharia de Software

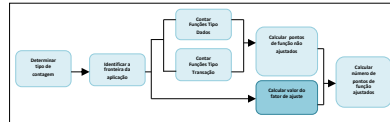
Monalessa Perini Barcellos

26

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Calcular valor do fator de ajuste



O número de pontos de função não ajustados não considera as características específicas do software (por exemplo: se é *stand alone* ou cliente servidor) – **requisitos não funcionais**.

O ajuste considera 14 características : Comunicação de Dados, Processamento Distribuído, Performance, Configuração Altamente Utilizada, Taxa de Transações, Entrada de Dados On-Line, Eficiência do Usuário Final, Atualização *On-Line*, Processamento Complexo, Reutilização, Facilidade de Operação, Facilidade de Instalação, Múltiplos Locais e Modificações Facilitadas.

Para cada característica deve ser atribuído um nível de influência de 0 (nenhuma) a 5 (grande).

O valor do fator de ajuste (VFA) é dado por  $VFA = (GIT * 0,01) + 0,65$ , onde GIT é o grau de influência total (soma de todos os valores dos níveis de influência).

Engenharia de Software

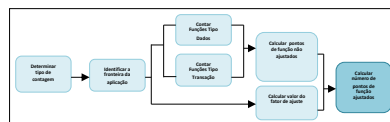
Monalessa Perini Barcellos

27

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Calcular número de pontos de função ajustados



Após calculado o valor do fator de ajuste, os pontos de função não ajustados serão ajustados, multiplicando-se o valor dos pontos de função não ajustados (PFNA) pelo valor do fator de ajuste (VFA). Assim:

$$PFA = PFNA \times VFA$$

Engenharia de Software

Monalessa Perini Barcellos

28

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

### Considerações importantes

O maior problema da APF é que os dados necessários para essa análise são bastante imprecisos no início de um projeto. No entanto, gerentes de projeto são, muitas vezes, obrigados a produzir estimativas antes de um estudo mais aprofundado.

Assim, pode ser pouco produtivo utilizar o método da APF integralmente para realizar as primeiras estimativas.

A APF é, antes de mais nada, um método de medição e, portanto, seu uso, da forma como proposta, para a realização de estimativas de tamanho é uma adaptação.

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Tendo em vista isso, foram propostas algumas variações mais simples do método voltadas para a realização de estimativas e que apresentam resultados satisfatórios, dentre elas a Contagem Indicativa, o uso de Complexidades Médias e a Contagem Estimativa. Todas consideram apenas PFs não ajustados

Na **Contagem Indicativa**, é necessária apenas a identificação das funções de dados (ALIs e AIEs). O número de PFs não ajustados é dado por  $PFNA = 35 * nALI + 15 * nAIE$ .

Em **Complexidades Médias** e **Contagem Estimativa**, é necessário identificar todas as funções de dados (ALIs e AIEs) e transacionais (EEs, CE e SEs), mas não é necessário usar as tabelas de complexidade.

# Gerência de Projetos de Software

- Para **Complexidades Médias** o número de PFs não ajustados é dado por:

$$nPF = 7,4 * nALI + 5,5 * nAIE + 4,3 * nEE + 5,4 * nSE + 3,8 * nCE \quad (ISBSG \text{ Benchmark})$$

- Para **Contagem Estimativa**, o número de PFs não ajustados dado por:

$$nPF = 7 * nALI + 5 * nAIE + 4 * nEE + 5 * nSE + 4 * nCE \quad (NESMA)$$

Estudos realizados pela NESMA com mais de 100 projetos apontam que a Contagem Estimativa tem menor dispersão em relação à contagem detalhada do que a Contagem Indicativa.

Assim, uma boa opção é iniciar as estimativas com uma técnica simplificada, tal como a Contagem Estimativa da NESMA, e, à medida que um maior entendimento dos requisitos é obtido, passar à contagem detalhada.

Engenharia de Software

Monalessa Perini Barcellos

31

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

**Exemplo:** Pequeno sistema hipotético desenvolvido para uma academia de ginástica com o objetivo de cadastrar os alunos matriculados e emitir um relatório gerencial que apresente o número de alunos matriculados totalizados por mês (inclui o código do aluno, nome do aluno, mês da matrícula, totalizador de alunos matriculados por mês e totalizador de alunos matriculados no ano).

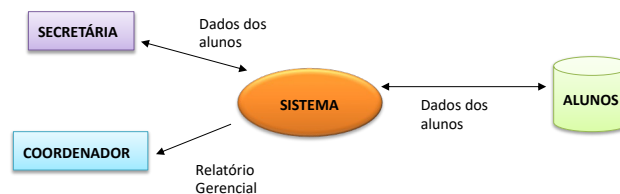


Diagrama de representação do sistema hipotético.  
O arquivo (tabela) Alunos possui 10 atributos.

Engenharia de Software

Monalessa Perini Barcellos

32

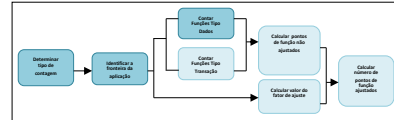


# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

*Exemplo:*

**Determinar o Tipo de Contagem**  
Projeto de Desenvolvimento



**Identificar a Fronteira da Aplicação**

Apresentado no diagrama. Não há interação com outros sistemas.

**Contar Funções Tipo Dados**

Arquivos Lógicos Internos	Tipos de Elementos de Dados	Tipos de Elementos de Registros
1 (Alunos)	10 (atributos de Alunos)	1
Arquivos de Interface Externa	Tipos de Elementos de Dados	Tipos de Elementos de Registros
0 (não há interação com outros sistemas)	0	0

Engenharia de Software

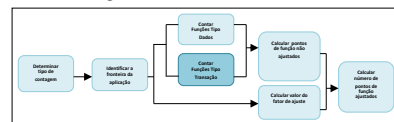
Monalissa Perini Barcellos

33

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

*Exemplo:*



**Contar Funções Tipo Transação**

Entradas Externas	Tipos de Elementos de Dados	Arquivos Referenciados
1 (Inclusão)	10 (atributos de Alunos)	1 (Alunos)
1 (Alteração)	10 (atributos de Alunos)	1 (Alunos)
1 (Exclusão)	1 (código do aluno)	1 (Alunos)
Consultas Externas	Tipos de Elementos de Dados	Arquivos Referenciados
1 (Consulta aos dados cadastrais)	10 (atributos de Alunos)	1 (Alunos)
Saídas Externas	Tipos de Elementos de Dados	Arquivos Referenciados
1 (Relatório Gerencial)	5 (informações apresentadas no relatório)	1 (Alunos)

Engenharia de Software

Monalissa Perini Barcellos

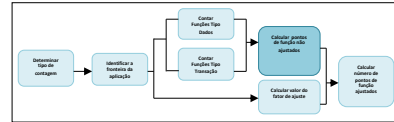
34

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Exemplo:

Calcular pontos de função não ajustados



Função	Itens Contados por Complexidade	Contribuição	Total por Complexidade	Total de PFNA da Função
ALI	1 Baixa	x 7	7	7
	0 Média	x 10	0	
	0 Alta	x 15	0	
AIE	0 Baixa	x 5	0	0
	0 Média	x 7	0	
	0 Alta	x 10	0	
EE	3 Baixa	x 3	9	9
	0 Média	x 4	0	
	0 Alta	x 6	0	
CE	1 Baixa	x 3	3	3
	0 Média	x 4	0	
	0 Alta	x 6	0	
SE	1 Baixa	x 4	4	4
	0 Média	x 5	0	
	0 Alta	x 7	0	

Total de Pontos de Função Não Ajustados: 23

Engenharia de Software

Monalissa Perini Barcellos

35

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Exemplo:

Calcular valor do fator de ajuste

Características Gerais do Sistema	Nível de Influência	Justificativa
Comunicação de Dados	0	O sistema opera em micro <i>stand-alone</i> , portanto, não possui comunicação de dados.
Processamento Distribuído	0	O sistema opera em micro <i>stand-alone</i> .
Performance	1	Requisitos de <i>performance</i> foram estabelecidos, mas nenhuma ação especial foi necessária.
Configuração altamente utilizada	0	Não há restrições operacionais.
Volume de Transações	0	Nenhum período de pico de transações esperado.
Entrada de dados <i>on-line</i>	5	Sistema <i>on-line</i> .
Eficiência do usuário final	3	Sistema desenvolvido com interface gráfica.
Atualização <i>on-line</i>	3	Sistema <i>on-line</i> , sem proteção para perda de dados.
Processamento complexo	0	O sistema não executa processamento matemático ou de segurança.
Reusabilidade	1	O sistema foi desenvolvido levando-se em conta reuso de rotinas.
Facilidade de instalação	4	Utilização de ferramenta automática para implantação do sistema.
Facilidade de Operação	2	Sistema <i>on-line</i> .
Múltiplos locais	0	Nenhuma solicitação do usuário para implantar a aplicação em mais de um local.
Modificação facilitada	0	Nenhuma solicitação do usuário para projetar a aplicação visando minimizar ou facilitar mudanças.

Grau de Influência Total = 19

$VFA = (GIT * 0,01) + 0,65 = 0,84$

Engenharia de Software

Monalissa Perini Barcellos

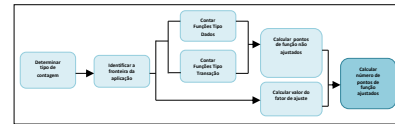
36

# Gerência de Projetos de Software

## Processo de Contagem de Pontos de Função

Exemplo:

Calcular pontos de função ajustados



$$PEA = PFNA \times VEA$$

$$PEA = 23 \times 0,84$$

$$PEA = 19,32$$

O sistema hipotético considerado no exemplo possui 19 pontos de função.

Engenharia de Software

Monalessa Perini Barcellos

37

# Gerência de Projetos de Software

## Estimativa de Esforço

- Para a realização de estimativas de tempo e custo, é fundamental estimar, antes, o esforço necessário para completar o projeto ou cada uma de suas atividades.
- Estimativas de esforço podem ser obtidas :
  - a) diretamente pelo julgamento de especialistas;
  - b) usando técnicas de decomposição;
  - c) computadas a partir de dados de tamanho ou de dados históricos;
  - d) por métodos empíricos.



Engenharia de Software

Monalessa Perini Barcellos

38

# Gerência de Projetos de Software

## Estimativa de Esforço

- Para adotar a decomposição (b), a *WBS* pode ser utilizada como base (o esforço é determinado para cada resultado práticos e pacote de trabalho em abordagens *top down* ou *bottom up*).
- Para usar estimativas de tamanho (c), pode ser considerado um *fator de produtividade*, que indica quanto em unidades de esforço é necessário para completar um projeto (ou módulo), descrito em unidades de tamanho.

*Exemplo:* quantos em homens-hora (unidade de esforço) são necessários para desenvolver 1000 LOCs (KLOC) ou 1 PF (unidades de tamanho).

Importante: esses fatores de produtividade devem levar em conta características dos projetos e da organização. Assim, pode ser útil ter vários fatores de produtividade, considerando perfis de projetos específicos.

Engenharia de Software

Monalessa Perini Barcellos

39

# Gerência de Projetos de Software

## Estimativa de Esforço

- Pode ser obtida a partir de métodos empíricos (d).

*Exemplo:*  $E = 5,5 + 0,73 * (KLOC)^{1,16}$  (Bailey-Basili)  
(obtenção do esforço necessário em pessoas-mês para desenvolver um projeto com tamanho dado em KLOC)

“Calibração” de métodos empíricos:

Experimentar o modelo usando resultados de projetos já finalizados, comparar os valores obtidos com os dados reais e analisar a eficácia do modelo. Se a concordância dos resultados não for boa, as constantes do modelo devem ser recalculadas usando dados organizacionais.

- Realizadas as estimativas de esforço, os recursos humanos adequados devem ser alocados ao projeto.

*Nota:* às vezes convém realizar a alocação dos recursos somente após uma versão preliminar do cronograma estar definida ou de forma paralela à sua definição.

Engenharia de Software

Monalessa Perini Barcellos

40

# Gerência de Projetos de Software

## Estimativa de Tempo e Elaboração do Cronograma

- De posse das estimativas de esforço e realizando em paralelo a alocação de recursos, é possível estimar a duração de cada atividade e, por conseguinte, do projeto como um todo.
- Se a estimativa de esforço tiver sido realizada para o projeto como um todo, então ela deverá ser distribuída pelas atividades do projeto (abordagem *top down*).
- Dados históricos de projetos já concluídos na organização são uma boa base para se fazer essa distribuição.
- Diretrizes de distribuição do esforço (e, consequentemente, do tempo), tais como:

Planejamento	Especificação e Análise de Requisitos	Projeto	Implementação	Teste e Entrega
Até 3%	De 10 a 25%	De 20 a 25%	De 15 a 20%	De 30 a 40%



Engenharia de Software

Monalessa Perini Barcellos

41

# Gerência de Projetos de Software

## Estimativa de Tempo e Elaboração do Cronograma

Para a elaboração do cronograma, é necessário:

- Definir a rede de atividades do projeto, considerando as dependências entre as atividades inicialmente estabelecidas no processo definido para o projeto e outras possíveis dependências (de recurso, por exemplo);
- Estabelecer qual é o caminho crítico do projeto, isto é, qual o conjunto de atividades que determina a duração do projeto. Um atraso em uma dessas atividades provocará atraso no projeto como um todo.
- Alocar os recursos (humanos, de hardware e de software) às atividades;
- Definir a duração das atividades e suas datas de início e fim;
- Elaborar um Gráfico de Tempo (ou Gráfico de Gantt).

Sendo assim, devem constar no cronograma:

- ✓ Sequência de atividades do projeto (processo do projeto);
- ✓ Recursos alocados às atividades (recursos humanos, software, hardware);
- ✓ Duração e datas de início e de fim das atividades;
- ✓ Identificação do caminho crítico.

Engenharia de Software

Monalessa Perini Barcellos

42

# Gerência de Projetos de Software

## Estimativa de Custos

- De posse das demais estimativas, é possível estimar os custos do projeto.
- De maneira geral, os seguintes itens devem ser considerados nas estimativas de custos:
  - ✓ Custos relativos ao esforço (recursos humanos) empregado pelos membros da equipe no projeto;
  - ✓ Custos de hardware e software (incluindo manutenção);
  - ✓ Outros custos relacionados ao projeto, tais como custos de viagens e treinamentos realizados no âmbito do projeto;
  - ✓ Despesas gerais, incluindo gastos com água, luz, telefone, pessoal de apoio administrativo, pessoal de suporte etc.



Engenharia de Software

Monalessa Perini Barcellos

43

# Gerência de Projetos de Software

## Estimativa de Custos

Exemplo:

Recurso	Quantidade	Valor Unitário	Quantidade de Horas	Valor Total
<b>Recursos Humanos</b>				
Gerente de Projetos	1	R\$ 100,00	80	R\$ 8.000,00
Engenheiro de Sistemas	1	R\$ 80,00	180	R\$ 14.400,00
Designer de Jogos	1	R\$ 70,00	208	R\$ 14.560,00
Programador	2	R\$ 70,00	362	R\$ 50.680,00
Tester	2	R\$ 70,00	64	R\$ 8.960,00
<b>Total com Recursos Humanos</b>				<b>R\$ 96.600,00</b>
<b>Recursos de Software</b>				
Axure RP	1	R\$ 200,00	—	R\$ 200,00
Cheetah 3D	1	R\$ 500,00	—	R\$ 500,00
<b>Total com Recursos de Software</b>				<b>R\$ 700,00</b>
<b>Recursos de Hardware</b>				
Óculos realidade virtual	2	R\$ 2.000,00	—	R\$ 4.000,00
<b>Total com Recursos de Hardware</b>				<b>R\$ 4.000,00</b>
<b>Outras Despesas</b>				
Passagens Vitória-Rio-Vitória	2	R\$ 1.000,00	—	R\$ 2.000,00
Diárias	8	R\$ 250,00	—	R\$ 2.000,00
<b>Total com Outras Despesas</b>				<b>R\$ 4.000,00</b>
<b>Custos Totais</b>				<b>R\$ 105.300,00</b>

Engenharia de Software

Monalessa Perini Barcellos

44

# Gerência de Projetos de Software

## O Plano do Projeto

- Todas as atividades realizadas no contexto da gerência de projeto devem ser documentadas em um Plano de Projeto, que contém todas as informações produzidas ao longo do planejamento do projeto (não apenas as referentes a escopo, tempo, custos e equipe).
- Cada organização deve estabelecer um modelo ou padrão para a elaboração desse documento, de modo que todos os projetos da organização contenham as informações consideradas relevantes.



Engenharia de Software

Monalessa Perini Barcellos

45

Universidade Federal do Espírito Santo

Centro Tecnológico

Departamento de Informática



**Disciplina: INF15978 – Engenharia de Software I**

**Prof.: Monalessa Perini Barcellos**

(monalessa@inf.ufes.br)

46