

SCE5764-ENGENHARIA DE SOFTWARE
Módulo 1

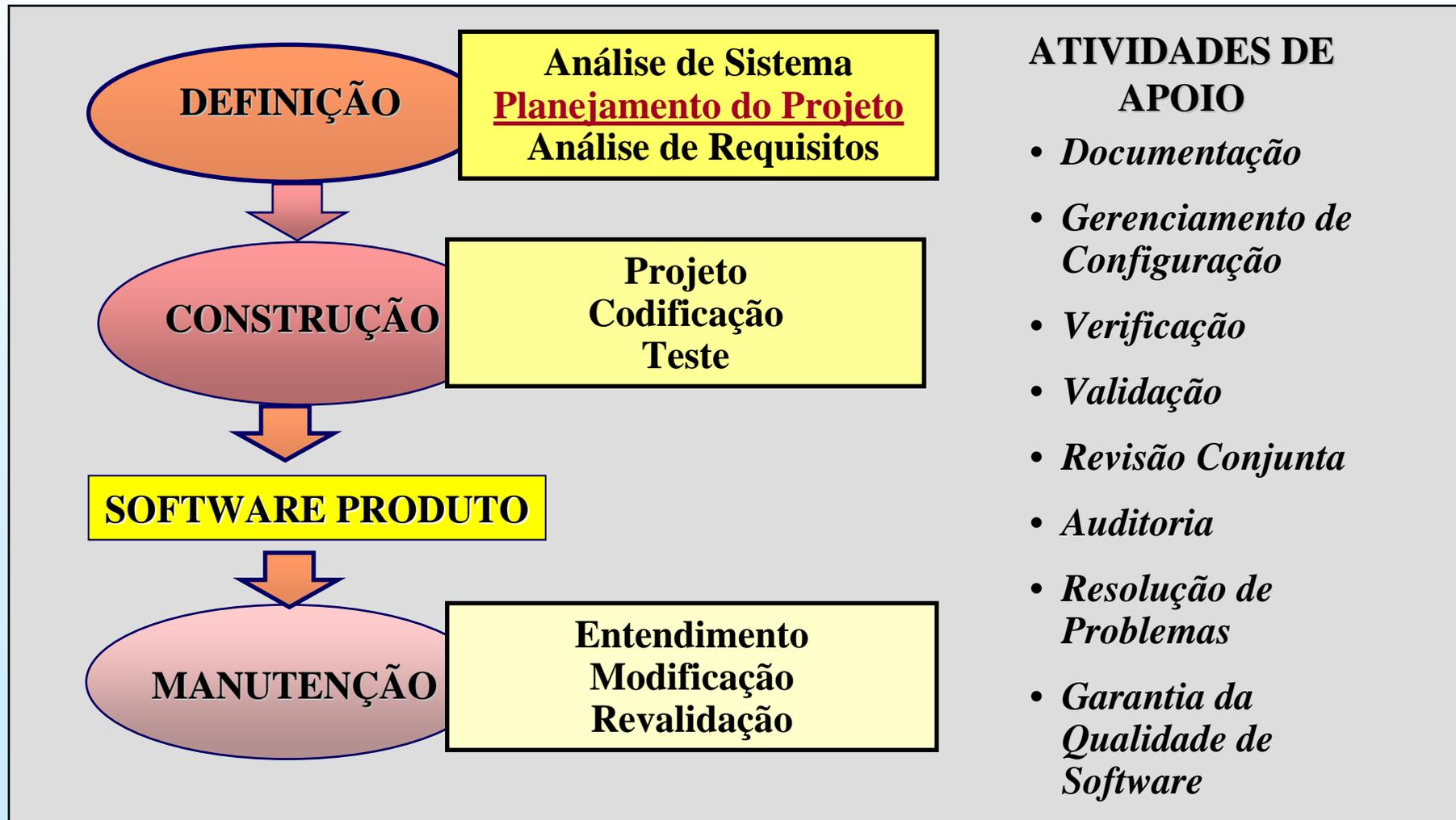
**PLANEJAMENTO DO
PROJETO**

Prof Paulo Masiero

Material: Rosely Sanches e Rosana T. Vaccare Braga

2004

Atividades da Engenharia de Software



Atividades da Engenharia de Software

DEFINIÇÃO

Análise de Sistema
Planejamento do Projeto
Análise de Requisitos

No **Planejamento do Projeto de Software** devem ser derivados:
estimativa do esforço humano
exigido, duração cronológica
e custo

Revalidação

ATIVIDADES DE APOIO

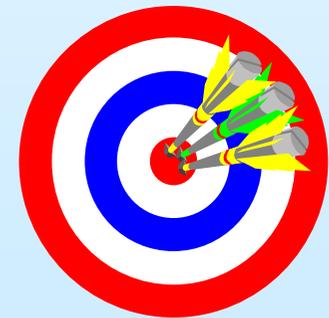
- *Documentação*
- *Gerenciamento de Configuração*
- *Verificação*
- *Validação*
- *Revisão Conjunta*
- *Auditoria*
- *Resolução de Problemas*
- *Garantia da Qualidade de Software*

Por que planejar?

- O desenvolvimento de software possui vários ciclos, que podem ser repetidos diversas vezes, até que se obtenha um produto que satisfaça aos requisitos do cliente
- O cliente precisa saber quanto custará e quando ficará pronto!!
- Há riscos envolvidos ☹
- O planejamento é essencial para:
 - decidir se o projeto continuará ou não
 - servir de base para o gerenciamento de projeto

Objetivos do Planejamento

- ↪ Determinar o alcance do trabalho a ser realizado: função, desempenho, interface e segurança
- ↪ Estimar recursos necessários ao desenvolvimento do software: recursos humanos, de hardware e de software
- ↪ Identificar tarefas a serem efetuadas
- ↪ Elaborar cronogramas
- ↪ Estimar esforço (custo) despendido



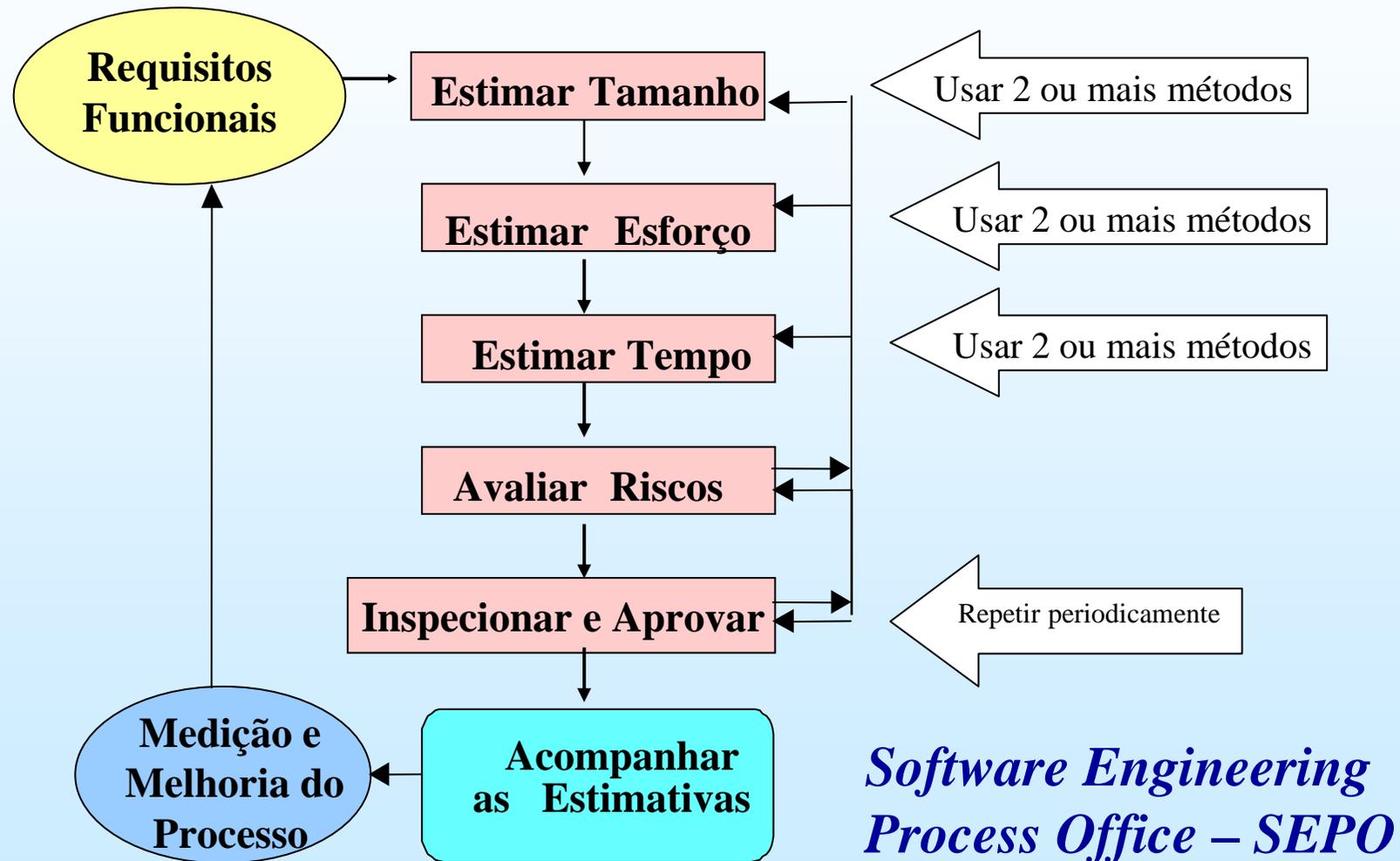
Atividades Fundamentais de Planejamento de Projeto

- ↳ Elaboração de Estimativas
- ↳ Análise de Riscos
- ↳ Elaboração de Cronograma
- ↳ Elaboração do Plano e Aprovação

Atividades Fundamentais de Planejamento de Projeto

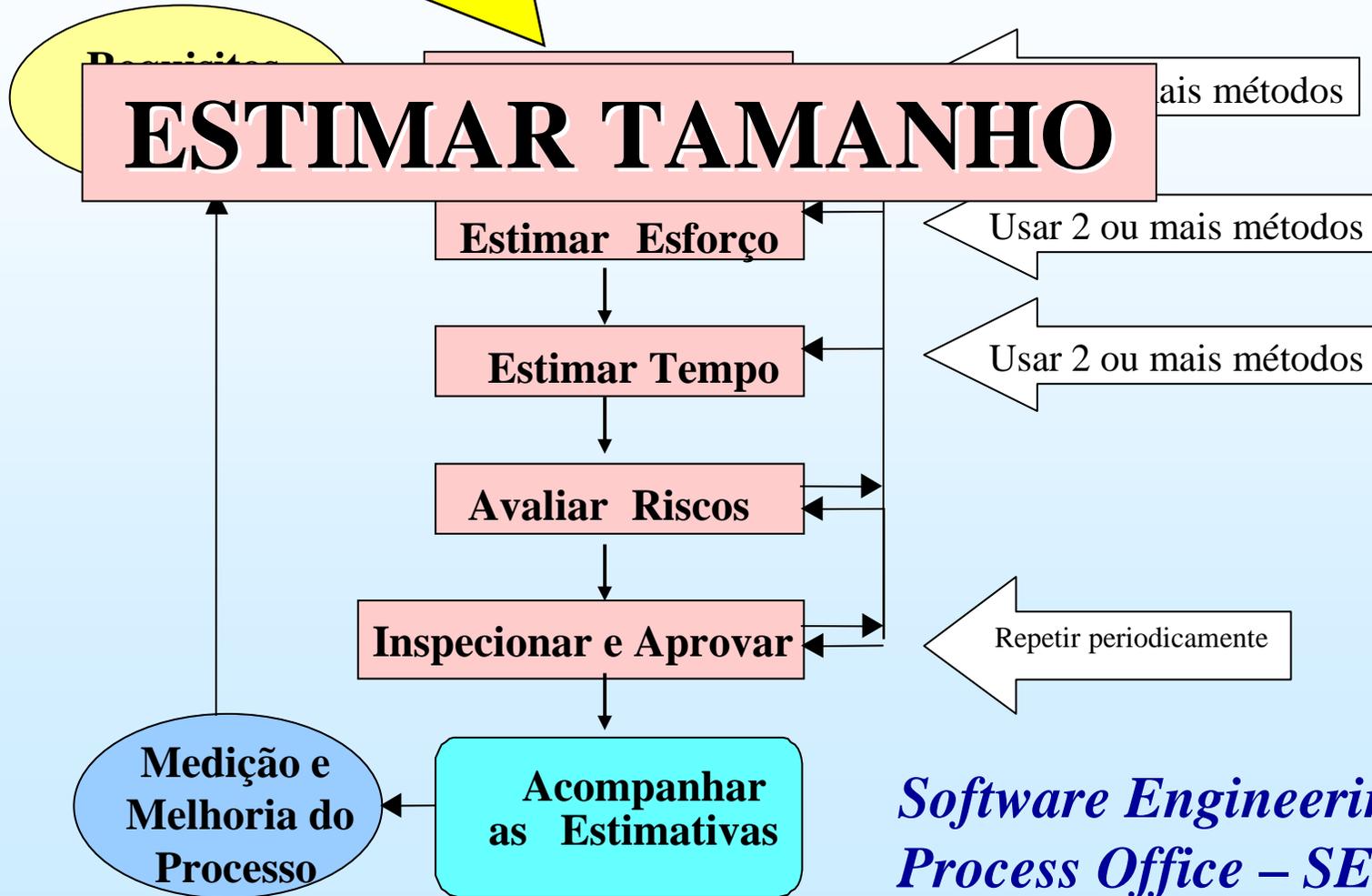
- ↳ Elaboração de Estimativas
- ↳ Análise de Riscos
- ↳ Elaboração de Cronograma
- ↳ Elaboração do Plano e Aprovação

Estimativas de Projeto de Software



**LINHAS DE CÓDIGO
PONTOS POR
FUNÇÃO**

Projeto de Software



*Software Engineering
Process Office – SEPO*

Como Medir o Tamanho do Software?

↪ O primeiro **problema** que se depara para elaborar **estimativas** é o dilema da **escolha** da métrica mais adequada para medir o tamanho de aplicações.

- Contagem de Linhas de Código (LOC)

- Contagem de Pontos por Função (PF)

Contagem de Linhas de Código

↳ A forma **familiar** de se medir tamanho de software é por meio da contagem de linhas de código.

- Contagem de Linhas de Código (LOC)

Contagem de Linhas de Código

VANTAGENS:

- Fáceis de serem obtidas
- Vários modelos de estimativa baseados em LOC ou KLOC

DESVANTAGENS:

- LOC depende da linguagem de programação
- Penalizam programas bem projetados, mas pequenos
- Não se adaptam às linguagens não procedimentais
- Difícil de obter em fase de planejamento

Contagem de Pontos por Função

↪ A contagem de Pontos por Função é uma técnica utilizada para medir o tamanho do software pela quantificação da funcionalidade do processamento da aplicação.

- Contagem de Pontos por Função (PF)

Contagem de Pontos por Função

↳ Uma das principais vantagens da contagem de pontos por função é a possibilidade de estimar a dimensão de projetos desde as primeiras fases de análise e projeto de sistemas, quando se dispõe de poucas informações sobre o sistema.

Como Medir o Tamanho do Software?

*Análise de Pontos por Função
IFPUG (International Function Points Users Group)*

*Pontos por Função
NESMA (Netherlands Function Points Users Group)*

- **Contagem de Pontos por Função (PF)**

Estimativa do Tamanho do Software

Contagem de Pontos por Função

↪ Cinco tipos de componentes **lógicos** ou **funções** da aplicação **afetam** de formas distintas o tamanho de um sistema:

- do tipo dados:
- Arquivos Lógicos Internos – **ALI**
- Arquivos de Interface Externa – **AIE**
- do tipo transações:
- Entradas Externas – **EE**
- Saídas Externas – **SE**
- Consultas Externas – **CE**

PF - PASSO 1

Identificar os componentes lógicos

↳ Para se determinar os componentes lógicos, primeiramente deve-se determinar a Fronteira da Aplicação.



EE- Entradas Externas

CE- Consultas Externas

AIE- Arquivo de Interfaces Externas

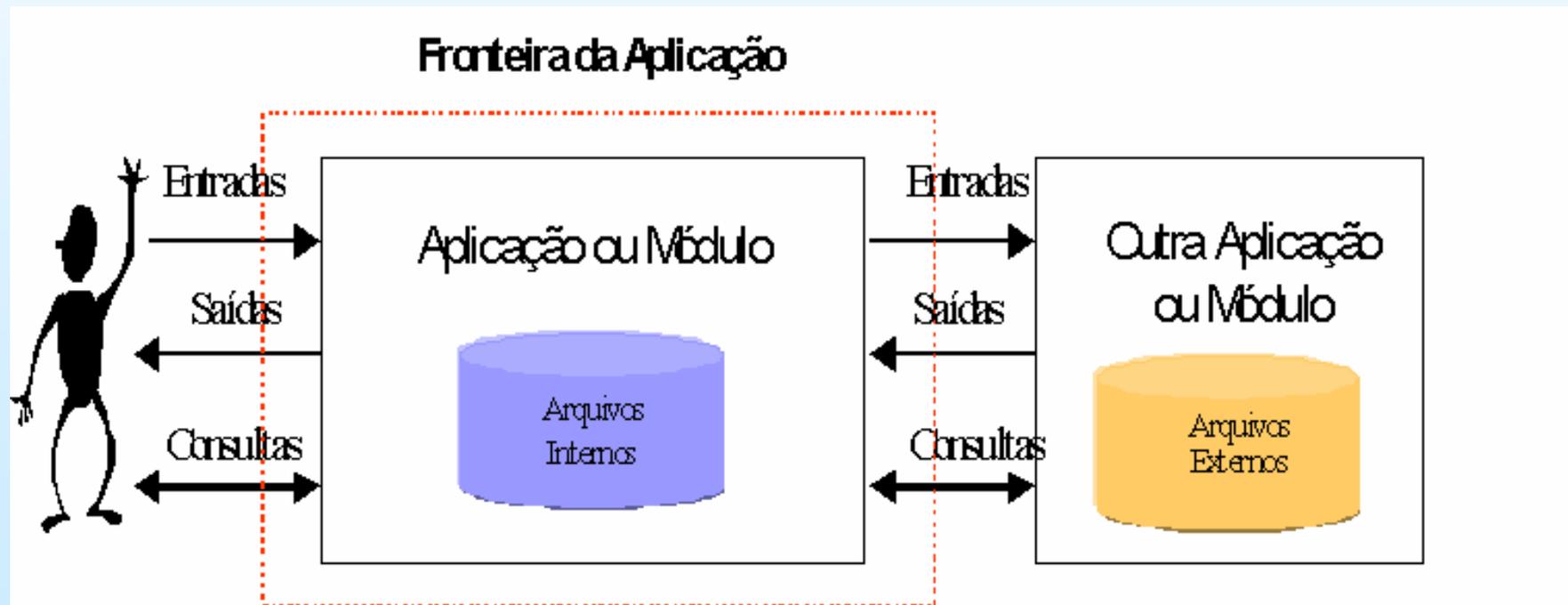
SE- Saídas Externas

ALI- Arquivo Lógicos Internos

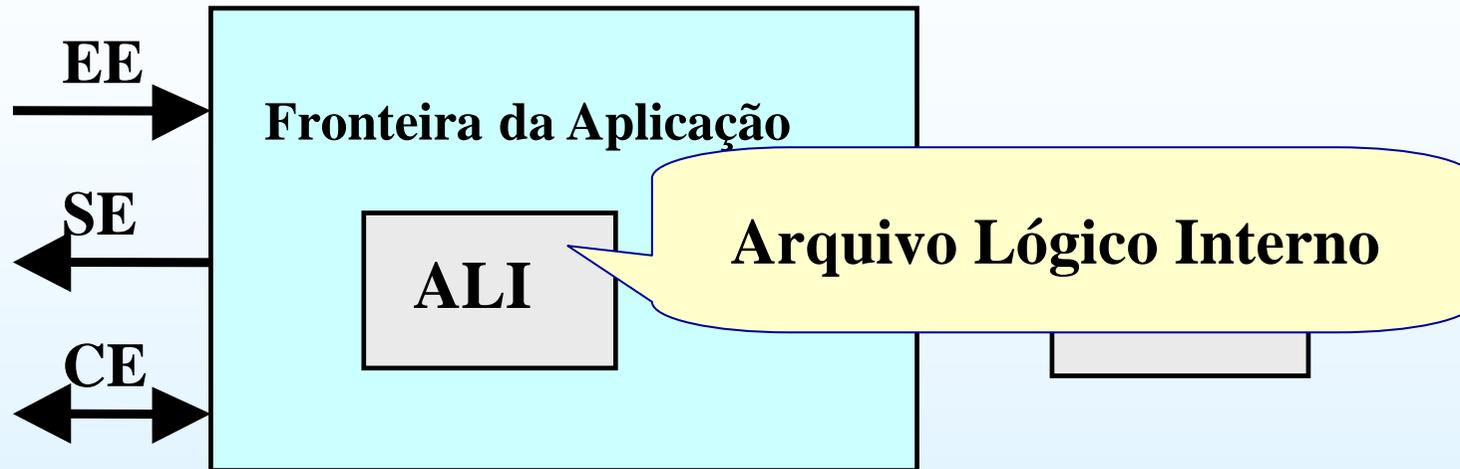
PF - PASSO 1

Identificar os componentes lógicos

- ↪ A **fronteira da aplicação** é a linha que separa o projeto ou aplicação que está sendo contada de outras aplicações ou sistemas da organização.



Arquivos Lógicos Internos - ALI



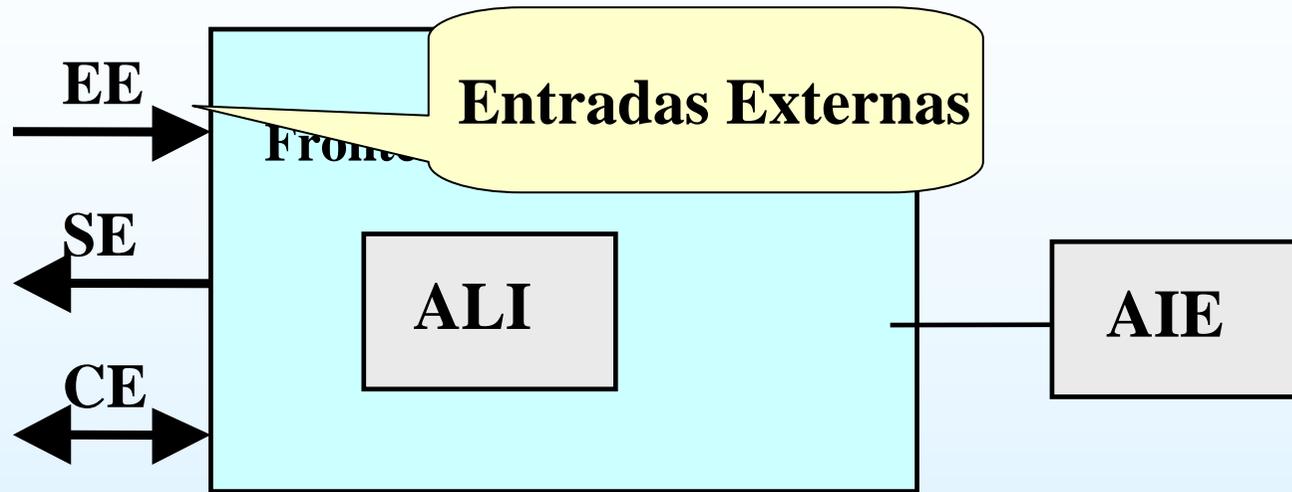
↳ Um *Arquivo Lógico Interno* (ALI) é um grupo de dados logicamente relacionados, ou informações de controle, identificados e modificados pelo usuário e mantidos dentro das fronteiras da aplicação que está sendo contada

Arquivos Interface Externa - AIE



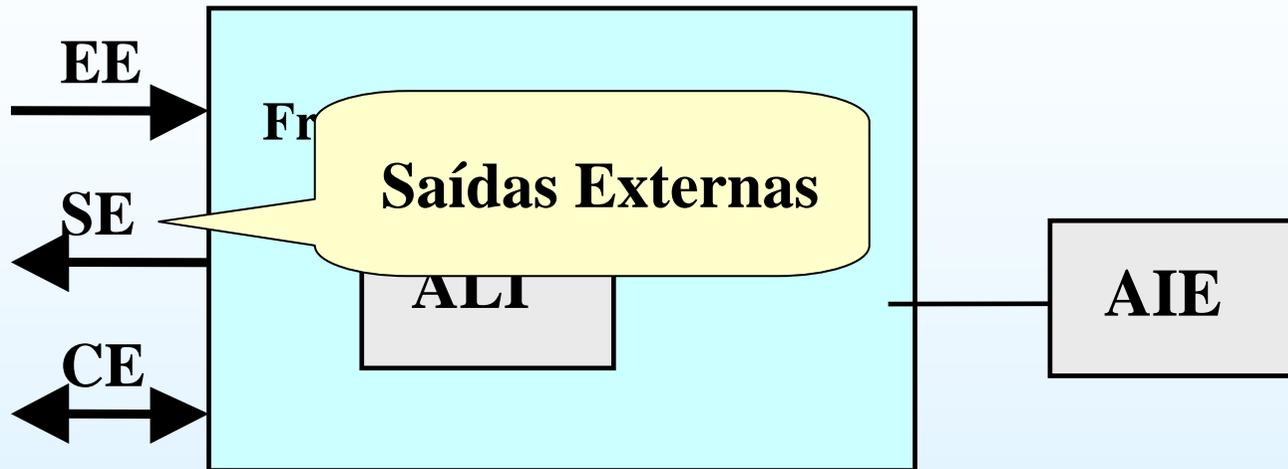
↳ Um Arquivo de Interface Externa (AIE) é um grupo de dados logicamente relacionados, ou informações de controle, utilizados no sistema que está sendo analisado, mas que são mantidos fora da fronteira da aplicação que está sendo contada.

Entrada Externa - EE



↳ Uma **Entrada Externa (EE)** é qualquer função ou transação que leva dados ou informações de controle de fora para dentro da fronteira da aplicação.

Saída Externa - SE



↳ Uma **Saída Externa (SE)** é um processo que fornece dados derivados para fora da aplicação que está sendo contada.

Saída Externa - SE

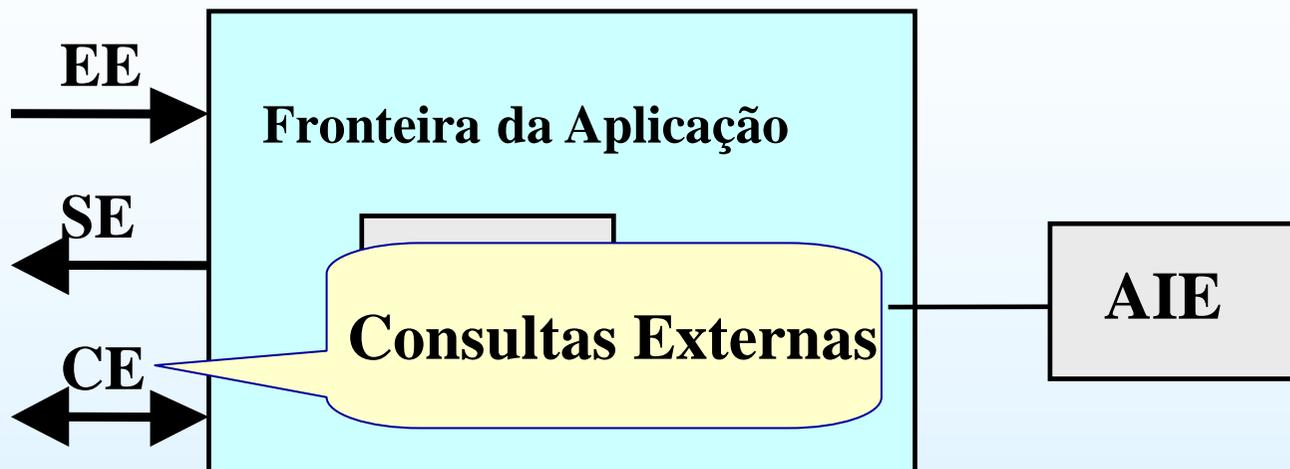


↳ Uma **Saída Externa (SE)** é um processo que fornece dados derivados para fora da aplicação que está sendo contada.

Dado Derivado

Ocorre quando um ou mais dados elementares são combinados para gerar elementos de dados adicionais

Consulta Externa - CE



↳ Uma **Consulta Externa (CE)** é uma transação que combina transações de entrada e de saída, resultando em recuperação de dados de um ALI ou AIE.

Contagem de Pontos de Função segundo o NESMA

↳ O *NESMA* apresenta três tipos de contagem de pontos de função:

- a contagem **indicativa** de ponto de função
- a contagem **estimada** de ponto de função
- a contagem **detalhada** de pontos de função

A ***Contagem Detalhada de Pontos por Função*** é a mesma técnica de ***Análise de Pontos por Função*** do IFPUG - *International Function Points Users Group*

Contagem de Pontos de Função segundo o NESMA

↪ O *NESMA* apresenta três tipos de contagem de pontos de função:

- a contagem indicativa de ponto de função
- a contagem estimada de ponto de função
- a contagem detalhada de pontos de função

A Contagem Detalhada de Pontos de Função é a mesma técnica de Análise de Pontos de Função do IFPUG - International Function Points Users Group

Contagem Estimada de PF

- ↪ A *Contagem Estimada de Pontos de Função* é utilizada na fase inicial da proposta de desenvolvimento, quando não se possuem dados detalhados do processo, apenas informações preliminares sobre os processos e o modelo de dados.
- ↪ Para a *Contagem Estimada de Pontos de Função* são necessárias informações um pouco mais detalhadas sobre a funcionalidade da aplicação, levantadas a partir das exigências do usuário (ou cliente).

Contagem Estimada de PF

↪ A *Contagem Estimada* assume que:

- os arquivos lógicos (ALI e AIE) têm complexidade baixa
- Os processos de entrada (EE), saída (SE) e consulta (CE) têm complexidade média

Contagem Estimada de PF

1º PASSO: Determinar todos os **AIE, ALI, EE, SE, CE**

2º PASSO: Atribuir a complexidade dos AIE e ALI como *Baixa*, e das funções tipo transação EE, SE e CE como *Média*

3º PASSO: Calcular o **total** da contagem dos pontos das funções, segundo a tabela de complexidade

Contagem Estimada de PF

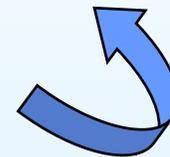
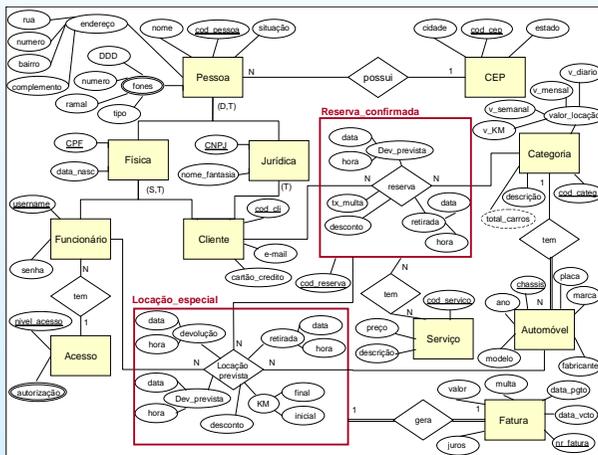
Tabela de Complexidade

	Nível de Complexidade		
Tipo de Função	Baixo	Médio	Alto
ALI	7	10	15
AIE	5	7	10
EE	3	4	6
SE	4	5	7
CE	3	4	6

Contagem Estimada de PF

Exemplo: LOCADORA DE CARROS

↳ Determinar todos os **Arquivos Lógicos Internos**



Esquema Lógico

Pessoa (cod_pessoa, nome, rua, numero, bairro, complemento, cep)

Fone_Pessoa (cod_pessoa, DDD, numero, ramal, tipo)

Pessoa_Física (CPF, cod_pessoa, data_nasc)

Pessoa_Jurídica (CNPJ, cod_pessoa, nome_fantasia)

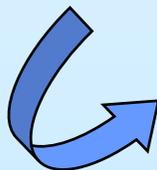
Funcionário (username, senha, nível_acesso, CPF)

Cliente (cod_cli, e-mail, cartão_credito, CNPJ, CPF)

Acesso (nível_acesso, autorização)

Categoria (cod_categ, descrição, v_diário, v_semanal, v_mensal, v_km, total_carros)

Reserva (cod_cli, cod_categ, dt_retirada, hr_retirada, dt_devoluc_prevista, hr_devoluc_prevista, tx_multa, desconto)



Contagem Estimada de PF

Exemplo: LOCADORA DE CARROS

ARQUIVOS LÓGICOS INTERNOS

Funções de Dados ou Transacionais	Tipo de Função	Complexidade (por default)	PF (não ajustados)
Pessoa	ALI	Baixa	7
Fone_Pessoa	ALI	Baixa	7
Pessoa_Física	ALI	Baixa	7
Pessoa_Jurídica	ALI	Baixa	7
Funcionário	ALI	Baixa	7
Cliente	ALI	Baixa	7
Acesso	ALI	Baixa	7
Categoria	ALI	Baixa	7
Reserva	ALI	Baixa	7
Reserva_Confirmada	ALI	Baixa	7
Automovel	ALI	Baixa	7
Locação_Prevista	ALI	Baixa	7
Locação_Especial	ALI	Baixa	7
Serviço	ALI	Baixa	7
Serviço_Reservado	ALI	Baixa	7
Fatura	ALI	Baixa	7

Contagem Estimada de PF

Exemplo: LOCADORA DE CARROS

↳ Determinar todos os Arquivos Lógicos Externos
(não existem)

Contagem Estimada de PF

Exemplo: LOCADORA DE CARROS

↳ Determinar todas as Entradas Externas, Saídas Externas e Consultas Externas



Contagem Estimada de PF

Exemplo: LOCADORA DE CARROS

ENTRADAS EXTERNAS E SAÍDAS EXTERNAS

Funções de Dados ou Transacionais	Tipo de Função	Complexidade (por default)	PF (não ajustados)
Incluir Cliente	EE	Média	4
alterar dados cliente	EE	Média	4
Excluir Cliente	EE	Média	4
Incluir Categoria	EE	Média	4
alterar dados categoria	EE	Média	4
Excluir Categoria	EE	Média	4
Incluir Automóvel	EE	Média	4
Alterar dados automóvel	EE	Média	4
Excluir automóvel	EE	Média	4
Incluir Funcionário	EE	Média	4
Alterar dados funcionário	EE	Média	4
Excluir Funcionário	EE	Média	4

Contagem Estimada de PF

Exemplo: LOCADORA DE CARROS

ENTRADAS EXTERNAS E SAÍDAS EXTERNAS

Funções de Dados ou Transacionais	Tipo de Função	Complexidade (por default)	PF (não ajustados)
Incluir Serviço	EE	Média	4
- Alterar Serviço	EE	Média	4
Excluir Serviço	EE	Média	4
Incluir Nível Acesso	EE	Média	4
Alterar Nível Acesso	EE	Média	4
Excluir Nível Acesso	EE	Média	4
Incluir Reserva	EE	Média	4
- Excluir Reserva	EE	Média	4
Retirar Automóvel	EE	Média	4
Devolução Automóvel	EE	Média	4
Pagamento Fatura	EE	Média	4
Impressão Comprovante Retirada	SE	Média	4
Impressão Comprovante Devolução	SE	Média	4
Listagem Automóveis por período	SE	Média	4
Listagem de reservas efetuadas na data atual	SE	Média	4
Consulta de Ocupação de automóveis	SE	Média	4
Impressão Relatório faturamento por período	SE	Média	4
Impressão das faturas, diariamente	SE	Média	4
Impressão das faturas, em atraso	SE	Média	4
Tamanho funcional Estimado			236 PF

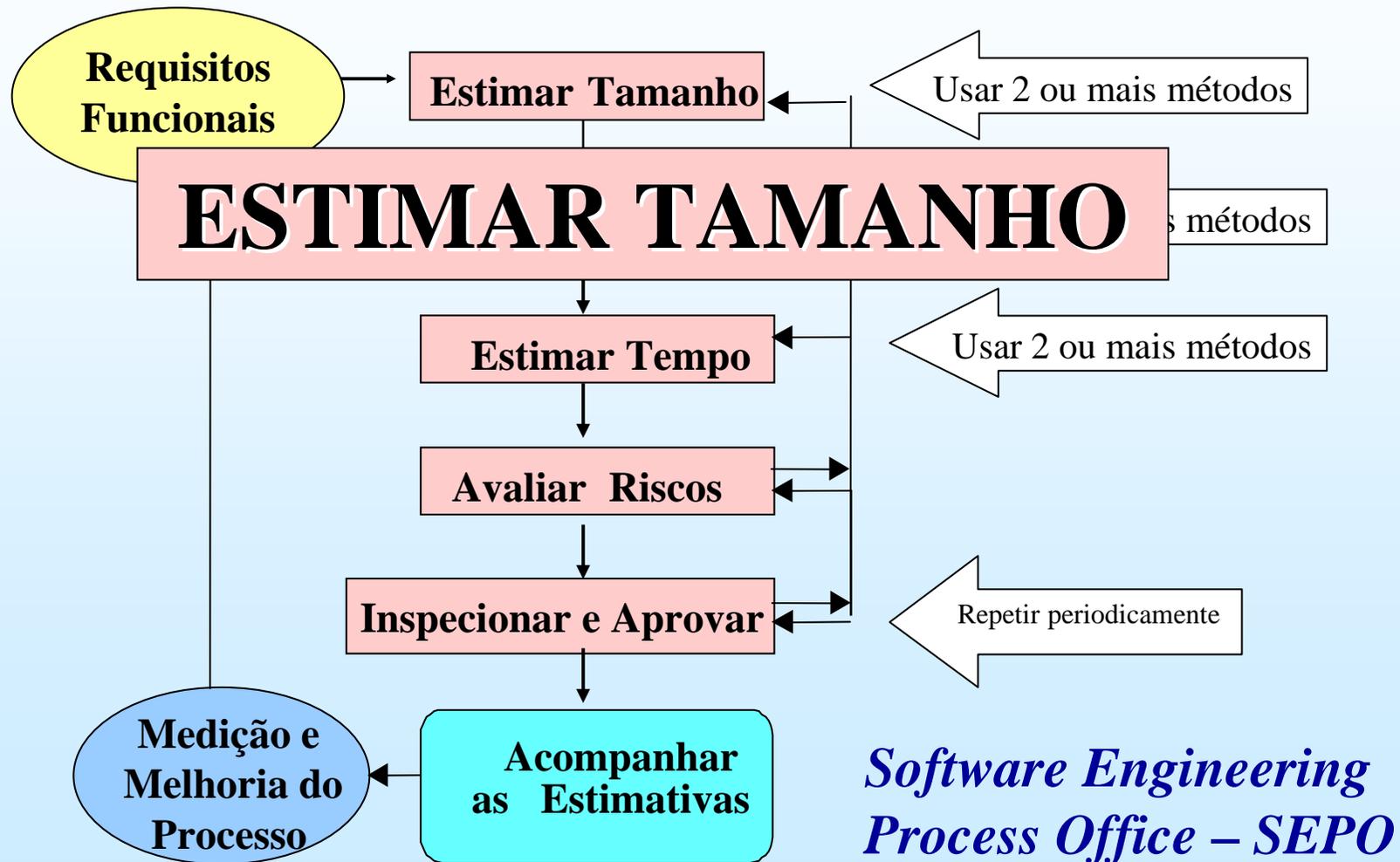
Conversão de Pontos de Função para Linhas de Código

- ↳ **Pontos de função** não ajustados podem ser convertidos na quantidade equivalente de **linhas de código**.
- ↳ A predição do número de instruções-fontes, a partir do **tamanho estimado em pontos de função**, é baseada na observação empírica do número de instruções requerido para implementar um ponto de função.

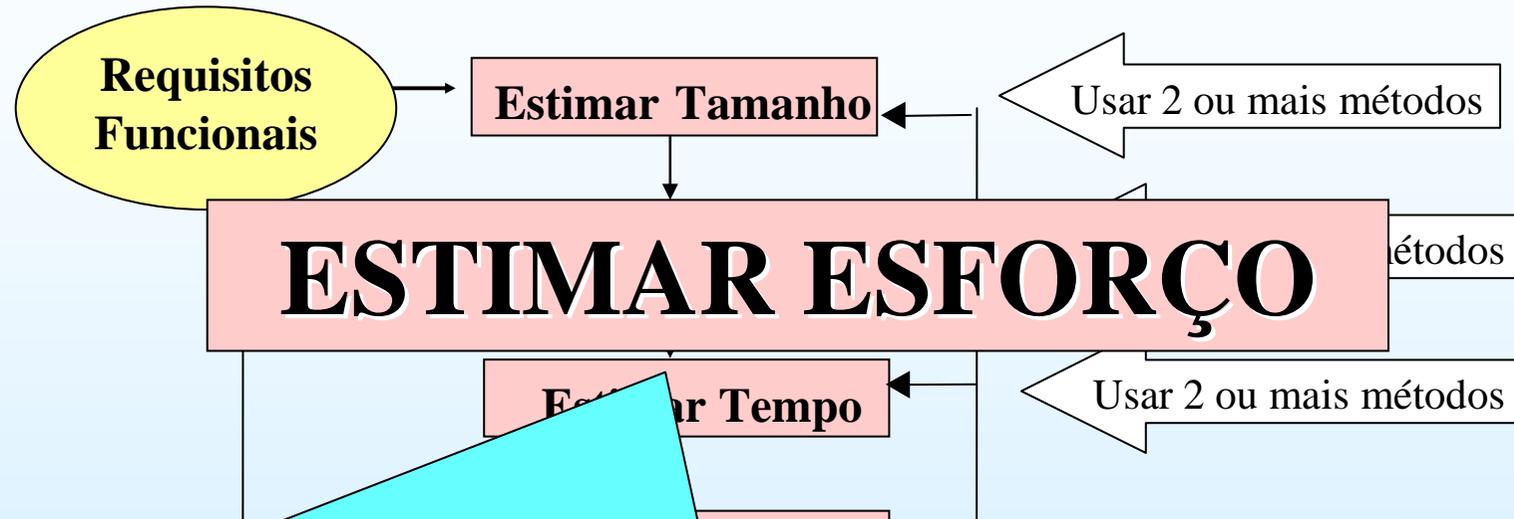
Conversão de Pontos de Função para Linhas de Código

<i>Linguagem</i>	<i>LOC/PF</i>	<i>Linguagem</i>	<i>LOC/PF</i>
ACCESS	38	FoxPro 2.5	34
Ansi SQL	13	HTML 3.0	15
Ansi COBOL 85	91	JAVA	53
C	128	LISP	64
C++	53	Natural 2	46
Clipper	19	Object Pascal	29
COBOL II	107	Oracle	40
dBase IV	36	Turbo C	128
Delphi	29	Turbo Pascal V.5	49
Fortran 95	71	Visual Basic 5	29

Estimativas de Projeto de Software



Estimativas de Projeto de Software



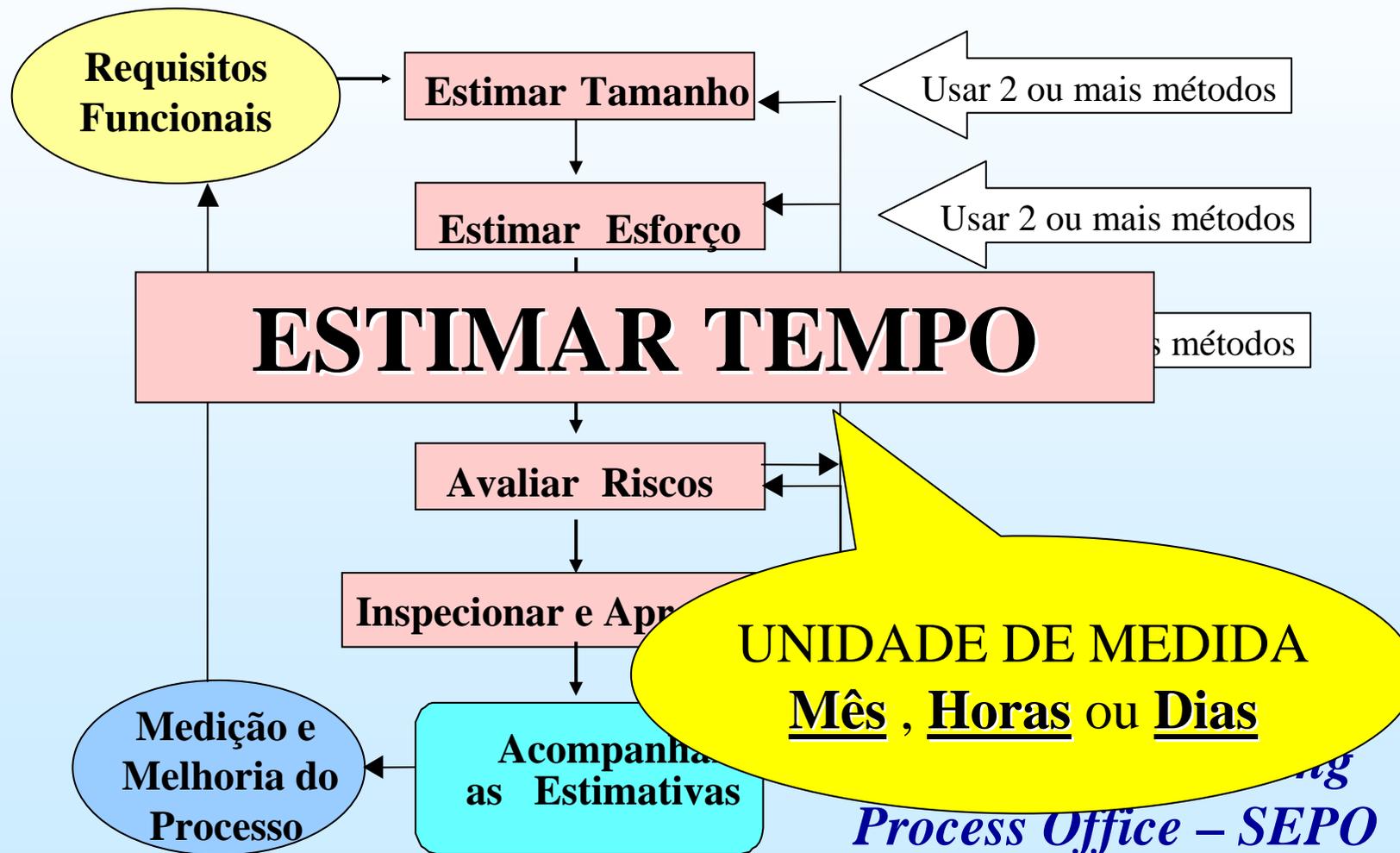
UNIDADE DE MEDIDA

Pessoas.Mês ou Pessoas.Hora

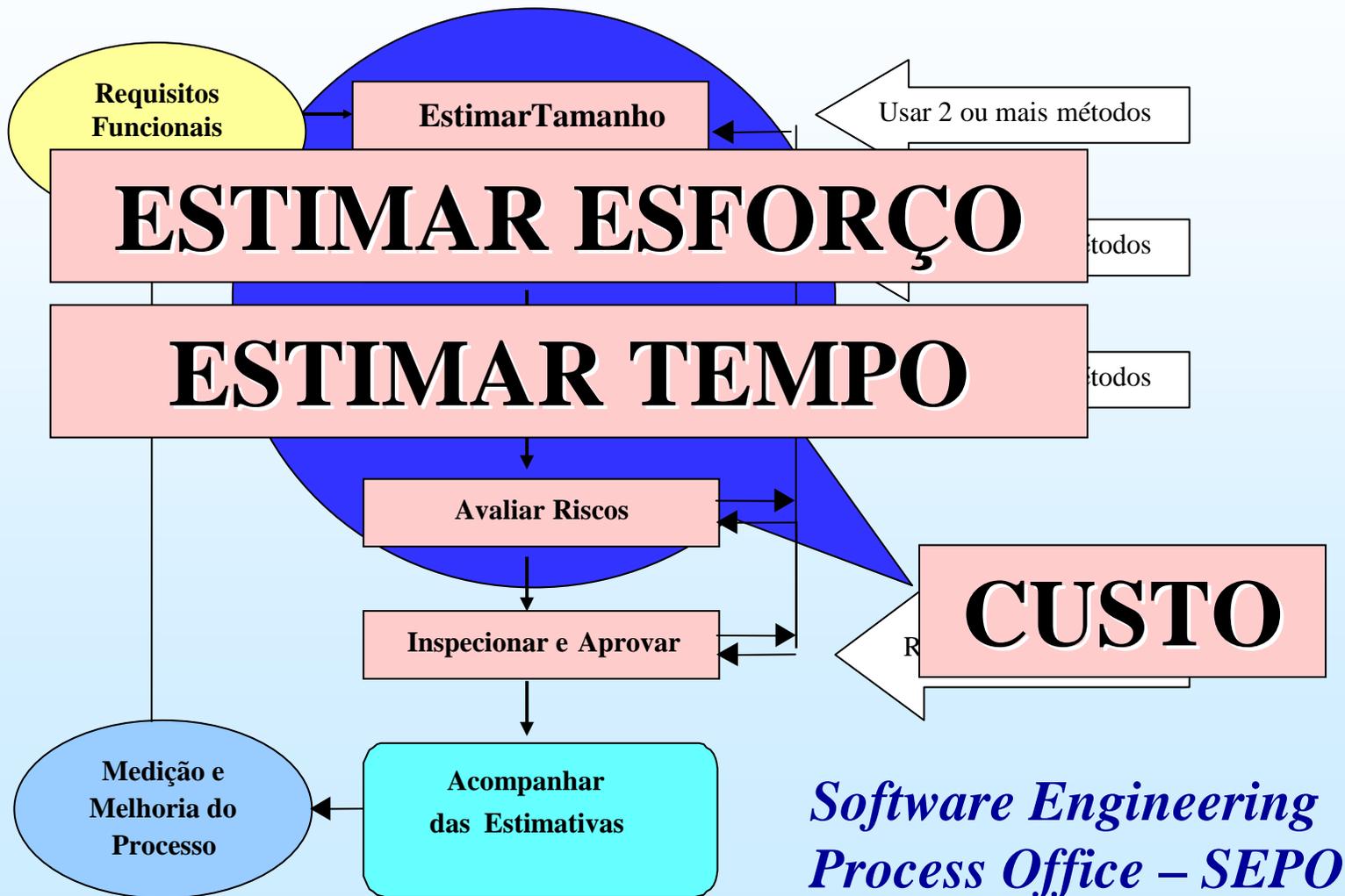
Exemplo:

esforço necessário para desenvolver o projeto = 12 Pessoas.Mês
(1 pessoas durante 12 meses)

Estimativas de Projeto de Software



Estimativas de Projeto de Software



Estimativas de Software

Conhecendo o tamanho

Requisitos
Funcionais

Estimar Tamanho

Usar 2 ou mais métodos

ESTIMAR ESFORÇO

todos

ESTIMAR TEMPO

todos

Avaliar Riscos

CUSTO

modelos empíricos
Modelo COCOMO 81

Software Engineering
Process Office – SEPO

MODELO COCOMO 81

COConstructive COst Model (Modelo de Custo Construtivo)

- ↪ Apresentado em 1981 por Boehm
- ↪ O **COCOMO** é um modelo desenvolvido para estimar esforço, prazo, custo e tamanho da equipe para um projeto de software
- ↪ Todas as referências ao **COCOMO** encontradas na literatura publicada até 1995 são citações desse modelo

MODELO COCOMO 81

↳ O **COCOMO** apresenta uma série de equações derivadas a partir do estudo de uma base de dados de 63 projetos, em sua maior parte na empresa *TRW Systems, Inc*

Aplicações de diferentes domínios

- negócios
- aplicações científicas
- sistemas de controle
- sistemas operacionais

MODELO COCOMO 81

↳ O **COCOMO** apresenta uma série de equações derivadas a partir do estudo de uma base de dados de 63 projetos, em sua maior parte na empresa *TRW Systems, Inc*

Aplicações implementadas em várias linguagens diferentes, cujas dimensões variavam de 2.000 até 1.000.000 de linhas de código (comentários excluídos)

MODELO *COCOMO* 81

↳ Para obter as equações do *COCOMO* foram combinados:

- a experiência
- resultados de outros modelos de estimativa de custo e
- a opinião subjetiva de gerentes de software experientes

MODELO COCOMO 81

↳ O **COCOMO** é apresentado na forma de um conjunto de modelos divididos hierarquicamente em três níveis:

- **Modelo COCOMO Básico**
- **Modelo COCOMO Intermediário**
- **Modelo COCOMO Avançado**

MODELO COCOMO 81

MODELO 1

Modelo COCOMO Básico

- calcula o esforço do desenvolvimento de software em função do tamanho estimado do programa, expresso em linhas de código

MODELO COCOMO 81

MODELO 1

Modelo COCOMO Básico

- Características
 - Esta versão é aplicável à grande maioria dos projetos de software, de pequeno ou médio porte.
 - É limitada por não considerar fatores que interferem no desenvolvimento do projeto, do tipo:
 - restrições de hardware
 - qualificação e experiência do pessoal de desenvolvimento e
 - uso de ferramentas técnicas modernas, entre outros.

MODELO COCOMO 81

MODELO 2

Modelo COCOMO Intermediário

- calcula o esforço de desenvolvimento de software em função do tamanho do programa e de um conjunto de direcionadores de custo, alternativamente chamados atributos ou fatores de software, que incluem avaliações subjetivas do produto, do hardware, do pessoal e dos atributos do projeto

MODELO COCOMO 81

MODELO

Modelo COCOMO

- calcula o esforço de desenvolvimento de software em função do tamanho do programa e de um conjunto de direcionadores de custo, chamados atributos ou atributos de projeto que incluem avaliações do produto, do hardware, dos atributos do projeto

Característica de desenvolvimento de software que tem efeito aumentativo ou diminutivo na quantidade de esforço de desenvolvimento final do projeto

Exemplos:

- ↳ a experiência da equipe de projeto
- ↳ a confiabilidade requerida do software

MODELO COCOMO 81

MODELO 3

Modelo COCOMO **Avançado**

- incorpora todas as características da versão intermediária, porém em cada passo do processo de engenharia de software.

MODELO COCOMO 81

- ↳ Depois da análise dos **requisitos funcionais do software**, o **tamanho** da aplicação deve ser estimado em milhares de linhas de código (**KLOC**)
- ↳ Determinar o tamanho no início do projeto é uma das limitações do método
- ↳ Uma alternativa viável é a utilização da técnica de contagem de **Pontos de Função**, por ser facilmente efetuada logo no início do projeto

MODELO COCOMO 81

↳ Pontos de função podem ser convertidos em linhas de código

<i>Linguagem</i>	<i>LOC/PF</i>	<i>Linguagem</i>	<i>LOC/PF</i>
ACCESS	38	FoxPro 2.5	34
Ansi SQL	13	HTML 3.0	15
Ansi COBOL 85	91	JAVA	53
C	128	LISP	64
C++	53	Smalltalk	22
Clipper	19	Object Pascal	29
COBOL II	107	Oracle	40
dBase IV	36	Turbo C	128
Delphi	29	Turbo Pascal V.5	49
Fortran 95	71	Visual Basic 5	29

MODELO COCOMO 81

↪ A aplicação do método começa pela **classificação** do produto a ser mensurado, categorizando o software em um de três tipos fundamentais de desenvolvimento identificados por Boehm:

- *Orgânico*
- *Embutido*
- *Semi-destacado*

MODELO *COCOMO* 81

Modelo COCOMO Básico
Modelo COCOMO Intermediário
Modelo COCOMO Avançado

**O MODO
ORGÂNICO**

**O MODO SEMI-
DESTACADO**

**O MODO
EMBUTIDO**

MODELO COCOMO 81

```
graph TD; A[Modelo COCOMO Básico  
Modelo COCOMO Intermediário  
Modelo COCOMO Avançado] --> B[O MODO ORGÂNICO]; A --> C[O MODO SEMI-DESTACADO]; A --> D[O MODO EMBUTIDO];
```

Modelo COCOMO Básico
Modelo COCOMO Intermediário
Modelo COCOMO Avançado

**O MODO
ORGÂNICO**

O MODO SEMI-
DESTACADO

O MODO
EMBUTIDO

Modos de Desenvolvimento de Software

MODO ORGÂNICO

↳ projeto relativamente pequeno (até 50.000 LC)

↳ equipes de software relativamente pequenas

↳ ambiente familiar

↳ maioria das pessoas ligadas ao projeto com grande experiência em trabalhar com sistemas relacionados a organização, e com um entendimento direto de como o sistema contribuirá para os objetivos da organização

Modos de Desenvolvimento de Software

MODO ORGÂNICO

- ↳ processo relativamente descontraído no modo de atender especificações de requisitos e interface
- ↳ ambiente de desenvolvimento relativamente estável com pouca necessidade de inovação
- ↳ inexistência de requisitos de entrega rígidos
- ↳ uso de algoritmos simples

MODELO COCOMO 81

```
graph TD; A[Modelo COCOMO Básico  
Modelo COCOMO Intermediário  
Modelo COCOMO Avançado] --> B[O MODO ORGÂNICO]; A --> C[O MODO SEMI-DESTACADO]; A --> D[O MODO EMBUTIDO];
```

Modelo COCOMO Básico
Modelo COCOMO Intermediário
Modelo COCOMO Avançado

O MODO
ORGÂNICO

O MODO SEMI-
DESTACADO

**O MODO
EMBUTIDO**

Modos de Desenvolvimento de Software

MODO EMBUTIDO

- ↪ também conhecido como *modo restrito*
- ↪ o principal fator que distingue um projeto de software de modo embutido é a necessidade de seguir restrições rigorosas
- ↪ o produto deve operar com (está embutido em) rígido complexo de hardware, software, regulamentos e procedimentos operacionais acoplados
- ↪ são projetos relativamente grandes com muita necessidade de inovação

Modos de Desenvolvimento de Software

MODO EMBUTIDO

- ↳ muito esforço em acomodar alterações e corrigir erros
- ↳ muito esforço para assegurar que o software realmente atende às especificações (alto custo de V&V) e para assegurar que as alterações são feitas corretamente (alto custo de gerenciamento de configuração)
- ↳ Exemplos de projetos do modo embutido são:
 - projeto de sistema de transferência eletrônica de fundos
 - projeto de sistema de controle de tráfego aéreo

MODELO COCOMO 81

Modelo COCOMO Básico
Modelo COCOMO Intermediário
Modelo COCOMO Avançado

O MODO
ORGÂNICO

**O MODO SEMI-
DESTACADO**

O MODO
EMBUTIDO

Modos de Desenvolvimento de Software

MODO SEMI DESTACADO

- ↳ também chamado de *modo difuso*
- ↳ representa um estágio intermediário entre os modos orgânico e embutido
- ↳ Características:
 - todos os membros da equipe tem um nível intermediário de experiência com sistemas relacionados ou
 - a equipe tem uma grande mistura de pessoas experientes e inexperiente ou
 - os membros tem experiência relacionada somente com alguns aspectos do sistema
- ↳ o sistema tem alguns requisitos funcionais e de interface rigorosos e alguns flexíveis

MODELO COCOMO **BÁSICO**

ESTIMATIVA DO ESFORÇO

MODO	EQUAÇÕES DE ESFORÇO
Orgânico	$E = 2.4 \times \text{KLOC}^{1.05}$ (homens-mês)
Semidestacado	$E = 3.0 \times \text{KLOC}^{1.12}$ (homens-mês)
Embutido	$E = 3.6 \times \text{KLOC}^{1.20}$ (homens-mês)

A quantidade **E** é o número de homens-mês estimado para o desenvolvimento do software

MODELO COCOMO INTERMEDIÁRIO

ESTIMATIVA DO ESFORÇO

MODOS

EQUAÇÕES DE ESFORÇO

Orgânico $E_{nom} = 3.2 \times KLOC^{1.05}$ (homens-mês)

Semidestacado $E_{nom} = 3.0 \times KLOC^{1.12}$ (homens-mês)

Embutido $E_{nom} = 2.8 \times KLOC^{1.20}$ (homens-mês)

$$E = FAE * E_{nom}$$

FAE: Fator de Ajuste do Esforço

E: é o número de homens-mês estimado para o desenvolvimento

FAE - Fator de Ajuste de Esforço

FAE: ATRIBUTOS DIRECIONADORES DE CUSTO

- ↳ É uma característica de desenvolvimento de software que tem efeito aumentativo ou diminutivo na quantidade de esforço de desenvolvimento final do projeto
- ↳ Boehm definiu **15 direcionadores de custo** para o *COCOMO* que, segundo ele, provocam impacto significativo na produtividade e nos custos do projeto

F AE - Fator de Ajuste de Esforço

F AE: ATRIBUTOS DIRECIONADORES DE CUSTO

↳ Podem ser agrupados em 4 categorias principais:

- Atributos do Produto
- Atributos Computacionais
- Atributos da Equipe de Desenvolvimento
- Atributos do Projeto

FAE - Fator de Ajuste de Esforço

Atributos			
do Produto	RELY	Confiabilidade requerida pelo software	
	DATA	Tamanho da base de dados	
	CPLX	Complexidade do software	
Computacionais	TIME	Restrições relativas ao tempo de	
	STOR	Restrições quanto ao uso de memória	
	VIRT	Mudanças do ambiente de software	
	TURN	Tempo de resposta	
da Equipe de Desenvolvimento	ACAP	Capacidade dos analistas	
	AEXP	Experiência na aplicação	
	PCAP	Capacidade dos programadores	
	VEXP	Experiência no ambiente de hardware	
	LEXP	Experiência com a linguagem de prog.	
do Projeto	MODP	Técnicas modernas de programação	
	TOOL	Uso de ferramentas de software	
	SCED	Prazo requerido para o desenvolvimento	

FAE - Fator de Ajuste de Esforço

- ↪ Cada um dos atributos deve ser ponderado (em importância e valor) numa **escala** de 6 pontos:
 - **MUITO BAIXO**
 - **BAIXO**
 - **NORMAL**
 - **ALTO**
 - **MUITO ALTO**
 - **EXTRA ALTO**

- ↪ Existe uma **Tabela** que indica em que condições devem ser aplicadas as taxas de 6 pontos

FAE - Fator de Ajuste de Esforço

Atributos		Nível de Influência	
do Produto	RELY	Confiabilidade requerida pelo software	de muito baixo (0,75) a muito alto (1,40)
	DATA	Tamanho da base de dados	de baixo (0,94) a muito alto (1,15)
	CPLX	Complexidade do software	de muito baixo (0,70) a extra alto (1,65)
Computacionais	TIME	Restrições relativas ao tempo de	de nominal (1,00) a extra alto (1,66)
	STOR	Restrições quanto ao uso de memória	de nominal (1,00) a extra alto (1,56)
	VIRT	Mudanças do ambiente de software	de baixo (0,87) a muito alto (1,30)
	TURN	Tempo de resposta	de baixo (0,87) a muito alto (1,15)
da Equipe de Desenvolvimento	ACAP	Capacidade dos analistas	de muito baixo (1,46) a muito alto (0,71)
	AEXP	Experiência na aplicação	de muito baixo (1,29) a muito alto (0,82)
	PCAP	Capacidade dos programadores	de muito baixo (1,42) a muito alto (0,70)
	VEXP	Experiência no ambiente de hardware	de muito baixo (1,21) a alto (0,90)
	LEXP	Experiência com a linguagem de prog.	de muito baixo (1,14) a alto (0,95)
do Projeto	MODP	Técnicas modernas de programação	de muito baixo (1,24) a muito alto (0,83)
	TOOL	Uso de ferramentas de software	de muito baixo (1,24) a muito alto (0,83)
	SCED	Prazo requerido para o desenvolvimento	de muito baixo (1,23) a muito alto (1,10)

FAE - Fator de Ajuste de Esforço

- ↪ Baseando-se na classificação e usando-se a *Tabela de Multiplicadores de Esforço de Desenvolvimento de Software*, um multiplicador de esforço é determinado
- ↪ O produto de todos multiplicadores de esforço torna-se um *FAE*

Tabela de Multiplicadores de Esforço de Desenvolvimento de Software

Atributos do Projeto	muito baixo	baixo	normal	alto	muito alto	extra alto
Capacidade dos Analistas	1.46	1.19	1.00	0.86	0.71	-
Experiência na Aplicação	1.29	1.13	1.00	0.91	0.82	-
Complexidade do Software	0.70	0.85	1.00	1.15	1.30	1.65
Tamanho da Base de Dados	-	0.94	1.00	1.08	1.16	-
Experiência com a Linguagem de Prog.	1.14	1.07	1.00	0.95	-	-
Técnicas Modernas de Programação	1.24	1.10	1.00	0.91	0.82	-
Capacidade dos Programadores	1.42	1.17	1.00	0.86	0.70	-
Confiabilidade requerida pelo Software	0.75	0.88	1.00	1.15	1.40	-
Prazo requerido para o Desenvolvimento	1.23	1.08	1.00	1.04	1.10	-
Restrições quanto ao uso de Memória	-	-	1.00	1.06	1.21	1.56
Restrições relativas ao Tempo de Máquina	-	-	1.00	1.11	1.30	1.66
Uso de Ferramentas de Software	1.24	1.10	1.00	0.91	0.83	-
Tempo de Resposta	-	0.87	1.00	1.07	1.15	-
Experiência no Ambiente de Hardware	1.21	1.10	1.00	0.90	-	-
Mudanças do Ambiente de Software	-	0.87	1.00	1.15	1.30	-

Tabela de Multiplicadores de Esforço de Desenvolvimento de Software - EXEMPLO

Atributos do Projeto	 muito baixo	baixo	normal	alto	 muito alto	extra alto
Capacidade dos Analistas	1.46	1.19	1.00	0.86	0.71	-
Experiência na Aplicação	1.29	1.13	1.00	0.91	0.82	-
Complexidade do Software	0.70	0.85	1.00	1.15	1.30	1.65
Tamanho da Base de Dados	-	0.94	1.00	1.08	1.16	-
Experiência com a Linguagem de Prog.	1.14	1.07	1.00	0.95	-	-
Técnicas Modernas de Programação	1.24	1.10	1.00	0.91	0.82	-
Capacidade dos Programadores	1.42	1.17	1.00	0.86	0.70	-
Confiabilidade requerida pelo Software	0.75	0.88	1.00	1.15	1.40	-
Prazo requerido para o Desenvolvimento	1.23	1.08	1.00	1.04	1.10	-
Restrições quanto ao uso de Memória	-	-	1.00	1.06	1.21	1.56
Restrições relativas ao Tempo de Máquina	-	-	1.00	1.11	1.30	1.66
Uso de Ferramentas de Software	1.24	1.10	1.00	0.91	0.83	-
Tempo de Resposta	-	0.87	1.00	1.07	1.15	-
Experiência no Ambiente de Hardware	1.21	1.10	1.00	0.90	-	-
Mudanças do Ambiente de Software	-	0.87	1.00	1.15	1.30	-

Tabela de Multiplicadores de Esforço de Desenvolvimento de Software

Atributos do Projeto		muito baixo	baixo	normal	alto	muito alto	extra alto
Capacidade dos Analistas	1.46	1.19	1.00	0.86	0.71	-	
Experiência na Aplicação	1.29	1.13	1.00	0.91	0.82	-	
Complexidade do Software	0.70	0.85	1.00	1.15	1.30	1.65	
Tamanho da Base de Dados	-	0.94	1.00	1.08	1.16	-	
Experiência com a Linguagem de Prog.	1.14	1.07	1.00	0.95	-	-	
Técnicas Modernas de Prog					0.82	-	
Capacidade dos Programad					0.70	-	
Confiabilidade requerida pel					1.40	-	
Prazo requerido para o Des					1.10	-	
Restrições quanto ao uso de					1.21	1.56	
Restrições relativas ao Tempo de Máquina	-	-	1.00	1.11	1.30	1.66	
Uso de Ferramentas de Software	1.24	1.10	1.00	0.90	0.82	-	
Tempo de Resposta	-	0.87	1.00	1.07	1.15	-	
Experiência no Ambiente de Hardware	1.21	1.10	1.00	0.90	-	-	
Mudanças do Ambiente de Software	-	0.87	1.00	1.15	1.30	-	

**O produto de todos
multiplicadores de esforço
torna-se um FAE**

$$E = FAE * E_{nom}$$

MODELOS COCOMO *BÁSICO* e *INTERMEDIÁRIO*

ESTIMATIVA DO PRAZO (em função do esforço E)

modo	equações de tempo
Orgânico	$P = 2.5 \times E^{0.38}$ (meses)
Semidestacado	$P = 2.5 \times E^{0.35}$ (meses)
Embutido	$P = 2.5 \times E^{0.32}$ (meses)

A quantidade **P** é o número de meses estimado para o desenvolvimento do software

MODELO COCOMO 81

- ↪ Os Pontos Fortes do modelo de estimativa são o **embasamento** em experimentações, em derivações matemáticas e em tabelas de dados
- ↪ Há **critérios** bem definidos para a determinação do nível de influência do ambiente profissional e da capacidade produtiva dos profissionais envolvidos com o projeto

Conversão de Pontos de Função para Linhas de Código



<i>Linguagem</i>	<i>LOC/PF</i>	<i>Linguagem</i>	<i>LOC/PF</i>
ACCESS	38	FoxPro 2.5	34
Ansi SQL	13	HTML 3.0	15
Ansi COBOL 85	91	JAVA	53
C	128	LISP	64
C++	53	Smalltalk	22
Clipper	19	Object Pascal	29
COBOL II	107	Oracle	40
dBase IV	36	Turbo C	128
Delphi	29	Turbo Pascal V.5	49
Fortran 95	71	Visual Basic 5	29

TAMANHO = 38 x 236(PF) = 8968 LOC = 8,968KLOC ≈ 9KLOC

Estimativa de Esforço e Prazo

Exemplo: LOCADORA DE CARROS

Usando modelo COCOMO intermediário

- ↪ O tamanho nominal do sistema é 9.0 KLOC
- ↪ O modo de desenvolvimento do projeto foi considerado orgânico

$$ESFORÇO: E = 3.2 \times KLOC^{1.05} * FAE$$

$$PRAZO: P = 2.5 \times E^{0.38}$$

Estimativa de Esforço e Prazo

Exemplo: LOCADORA DE CARROS

Direcionadores

↙ O gerente avaliou os 15 direcionadores de custo e chegou ao seguinte resultado:

- *Complexidade do Software*: Alta (1.15)
- *Restrições quanto ao uso de Memória*: Alto (1.06)
- *Experiência com a Linguagem de Prog*: Baixa (1.14)
- *Capacidades dos Programadores*: Baixa (1.17)
- Os outros atributos foram considerados **nominais**

Estimativa de Esforço e Prazo

Exemplo: LOCADORA DE CARROS

ESFORÇO

$$E = 3.2 \times \text{KLOC}^{1.05} * \mathbf{FAE}$$

$$\mathbf{FAE} = (1.15 * 1.06 * 1.14 * 1.17) = \mathbf{1.63}$$

$$E = 3.2 * (9.0)^{1.05} * \mathbf{1.63}$$

$$E = 3.2 * 10.05 * 1.63$$

$$\mathbf{E = 52.42 \text{ pessoas-mês}}$$

Estimativa de Esforço e Prazo

Exemplo: LOCADORA DE CARROS

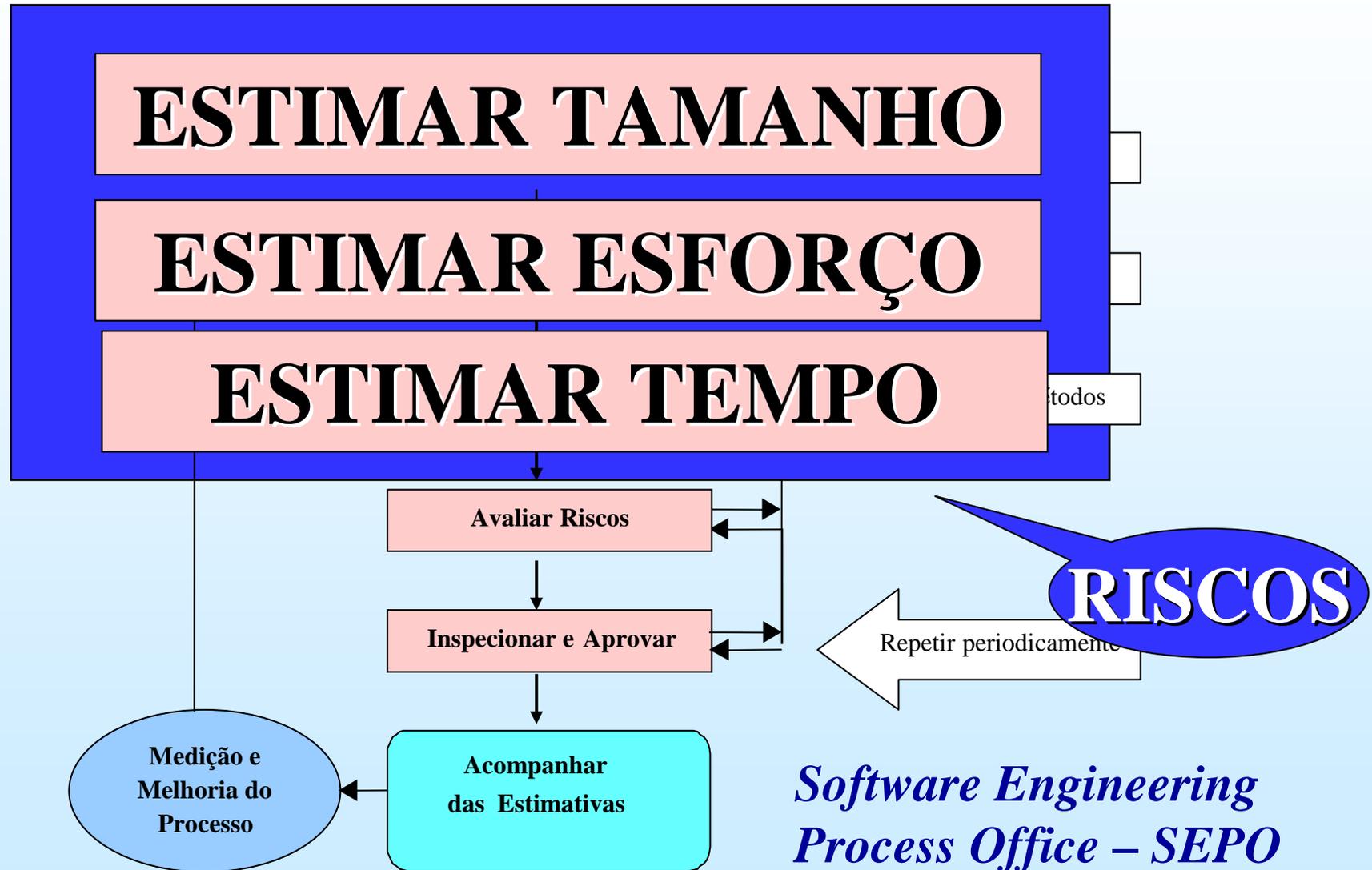
PRAZO

$$P = 2.5 \times E^{0.38}$$

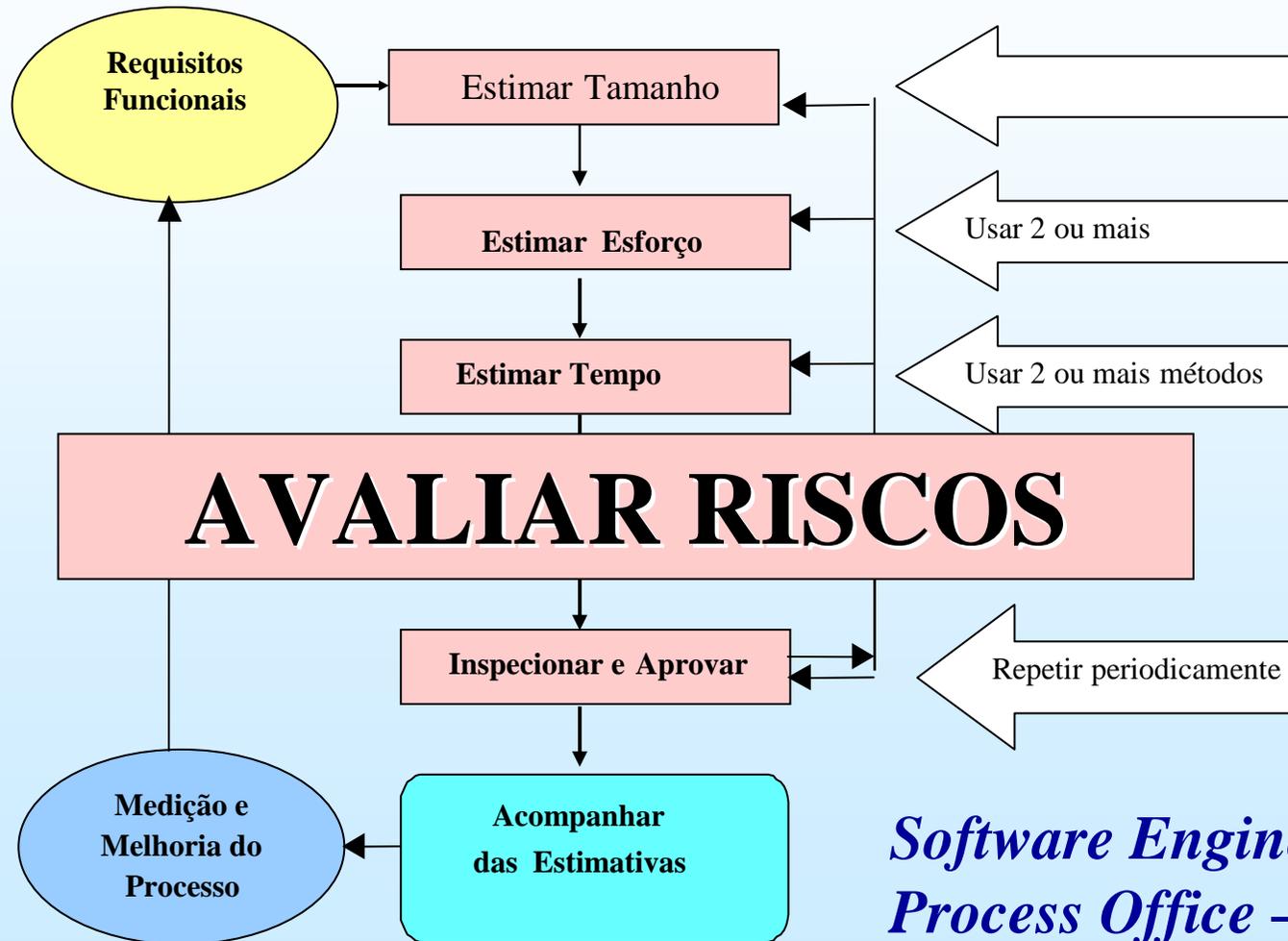
$$P = 2.5 * 52.42^{0.38}$$

$$P = 11.26 \text{ meses}$$

Estimativas de Projeto de Software



Estimativas de Projeto de Software



*Software Engineering
Process Office – SEPO*

Atividades Fundamentais de Planejamento de Projeto

- ↳ Elaboração de Estimativas
- ↳ Análise de Riscos
- ↳ Elaboração de Cronograma
- ↳ Elaboração do Plano e Aprovação

Gerenciamento de Riscos

↳ **Risco** é um problema em potencial – pode ou não acontecer

↳ É importante:

- Identificá-lo
- Avaliar sua probabilidade de ocorrência
- Estimar seu impacto
- Estabelecer um plano de contingência para o caso dele efetivamente ocorrer

↳ Não são tarefas fáceis!!!

Plano de Projeto-Riscos

III. RISCOS DO PROJETO

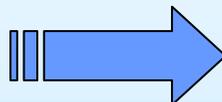
1. Análise dos riscos



Passos para analisar os riscos:

- identificação
- avaliação
- disposição por ordem de prioridade

2. Administração dos riscos



Passos para atacar os riscos:

- estratégias de administração
- resolução
- monitoração

“O fundamental é que os Riscos assumidos sejam os Riscos certos”

Plano de Projeto-Riscos

Identificação dos Riscos

de Projeto

identificam problemas orçamentários, de cronograma, de pessoal, de recursos, de clientes, de requisitos e o impacto no projeto do software

Técnicos

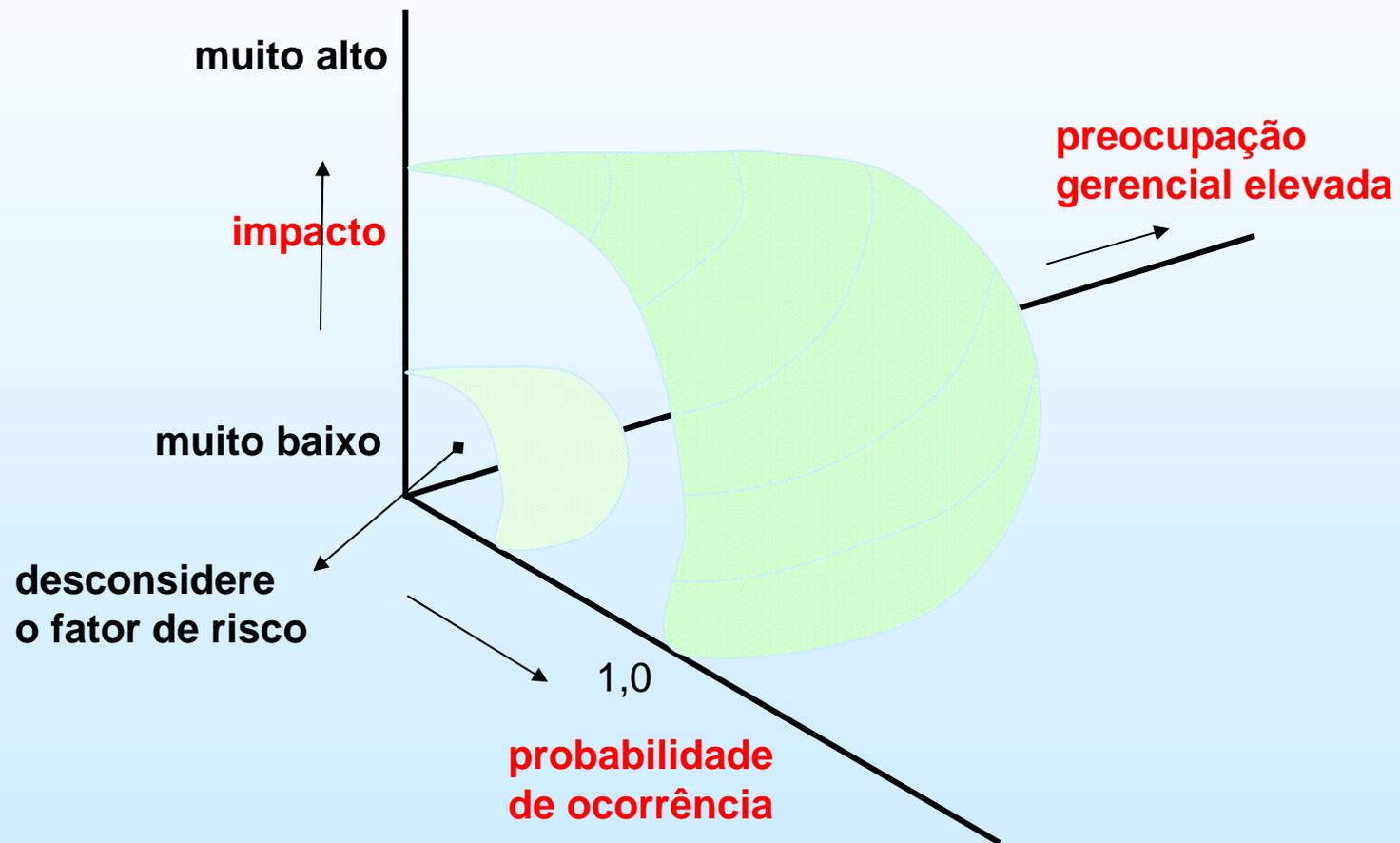
identificam potenciais problemas de projeto, implementação, interface, verificação e manutenção

do Negócio

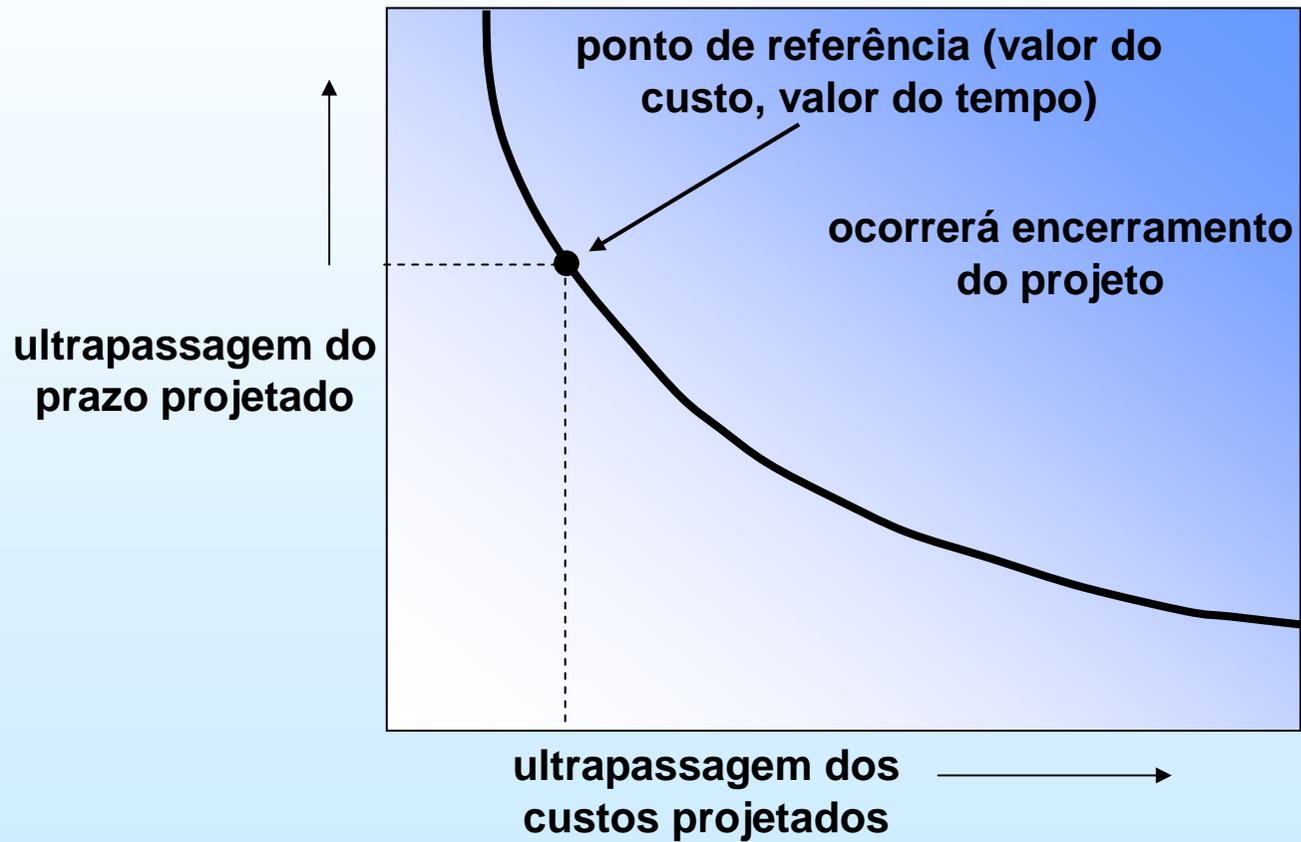
podem destruir até os melhores projetos: construir um produto que ninguém quer; ou que não se encaixe mais na estratégia da empresa; perder o apoio da administração, ou o compromisso orçamentário

“Se você não atacar ativamente os riscos técnicos e de projeto, eles lhe atacarão ativamente.” Gilb

Risco e Preocupação Gerencial



Nível de Referência de Risco



Atenuação, Monitoração e Administração do Risco

- ↪ **Atenuação** — como podemos evitar o risco?
- ↪ **Monitoração** — que fatores podem ser rastreados para ajudar-nos a prevenir a ocorrência do risco?
- ↪ **Administração** — que planos de contingência temos para o caso do risco se tornar efetivo?

Exemplos para pensar: Cliente não sabe o que quer, Alta Rotatividade de Pessoal

Cuidado com a administração de riscos!

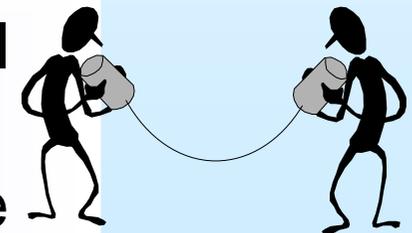
Mito: “Se sairmos fora do cronograma, adicionamos mais programadores e recuperamos o atraso”.



Isso faz o cronograma atrasar ainda mais!

Motivo: a comunicação é absolutamente essencial para o desenvolvimento do software.

Todo novo caminho de comunicação exige esforço adicional e portanto, tempo adicional.



Exemplo 1: riscos relacionados ao cliente

Questões a serem respondidas

- Você já trabalhou com esse cliente no passado?
- O cliente tem uma idéia sólida dos requisitos?
- O cliente deseja participar das revisões?
- O cliente é tecnicamente sofisticado?
- O cliente entende o processo de engenharia de software?

Exemplo 2: Riscos Tecnológicos

Questões a serem respondidas

- A tecnologia é nova para sua empresa?
- Algum hardware novo ou não testado está envolvido?
- Será necessária uma interface com o usuário especializada?
- Você está usando novos métodos de engenharia de software?
- Você está usando métodos de desenvolvimento de software não convencionais, como métodos formais, abordagens de IA, redes neurais?
- Existem restrições significativas de desempenho?

Riscos: os 10 mais (Bohem)

- ↪ Imprevistos de pessoal
- ↪ Cronogramas e orçamentos não realísticos
- ↪ Desenvolvimento das funções erradas
- ↪ Desenvolvimento da interface com o usuário errada
- ↪ Requisitos sofisticados, sem necessidade
- ↪ Fluxo contínuo de mudanças nos requisitos
- ↪ Imprevistos em serviços terceirizados
- ↪ Imprevistos em componentes terceirizados
- ↪ Imprevistos de desempenho em tempo real
- ↪ Capacidade de ciência de computação excedida

Exposição ao risco (ER)

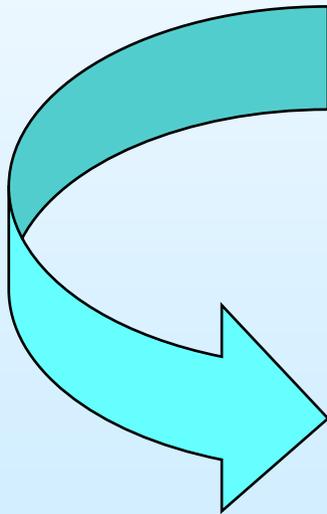
↳ $ER = P \times C$

↳ Onde P é a probabilidade de ocorrência de um risco e C é o custo para o projeto no caso do risco ocorrer

↳ A exposição pode ser calculada para cada risco da tabela de riscos, com isso pode-se ajustar a estimativa final de custo de um projeto.

Fator que Reduz o Risco das Estimativas

DADOS HISTÓRICOS



- Estimativas podem ser feitas com maior segurança
- Prazos podem ser estabelecidos para se evitar dificuldades passadas
- Riscos globais podem ser reduzidos

Atividades Fundamentais de Planejamento de Projeto

- ↳ Elaboração de Estimativas
- ↳ Análise de Riscos
- ↳ Elaboração de Cronograma
- ↳ Elaboração do Plano e Aprovação

Elaboração do Cronograma

TAREFAS:

1. Identificar e selecionar os recursos para o projeto
2. Inter-relacionar as atividades e definir precedências
3. Calcular o caminho crítico
4. Alocar recursos nas atividades
5. Preparar cronograma do projeto

Elaboração do Cronograma

TAREFAS:

1. Identificar e selecionar os recursos para o projeto
2. Inter-relacionar as atividades e definir precedências
3. Calcular o caminho crítico
4. Alocar recursos nas atividades
5. Preparar cronograma do projeto

Identificar e Selecionar os Recursos para o Projeto

- ↪ A identificação e seleção de recursos para o projeto é usualmente conduzida em paralelo com a elaboração de estimativas de tempo, devido à dependência intrínseca entre duração e quantidade de recursos.
- ↪ Para se calcular a duração mais precisa do projeto, é necessário que se conheçam todos os recursos alocados nas atividades e a produtividade de cada um deles.

Identificar e Selecionar os Recursos para o Projeto

↳ Devem ser identificados e selecionados:

- todos os recursos humanos (quantos e quais profissionais),
- todos os materiais de consumo e equipamentos (quantos, quando e quais os tipos de equipamentos) e
- todos os recursos financeiros (quanto e quando) necessários à execução do projeto.

Elaboração do Cronograma

TAREFAS:

1. Identificar e selecionar os recursos para o projeto
2. Inter-relacionar as atividades e definir precedências
3. Calcular o caminho crítico
4. Alocar recursos nas atividades
5. Preparar cronograma do projeto

Inter-relacionar as Atividades e Definir Precedências

↳ O objetivo dessa tarefa é identificar atividades interdependentes para que o cronograma do projeto seja elaborado.

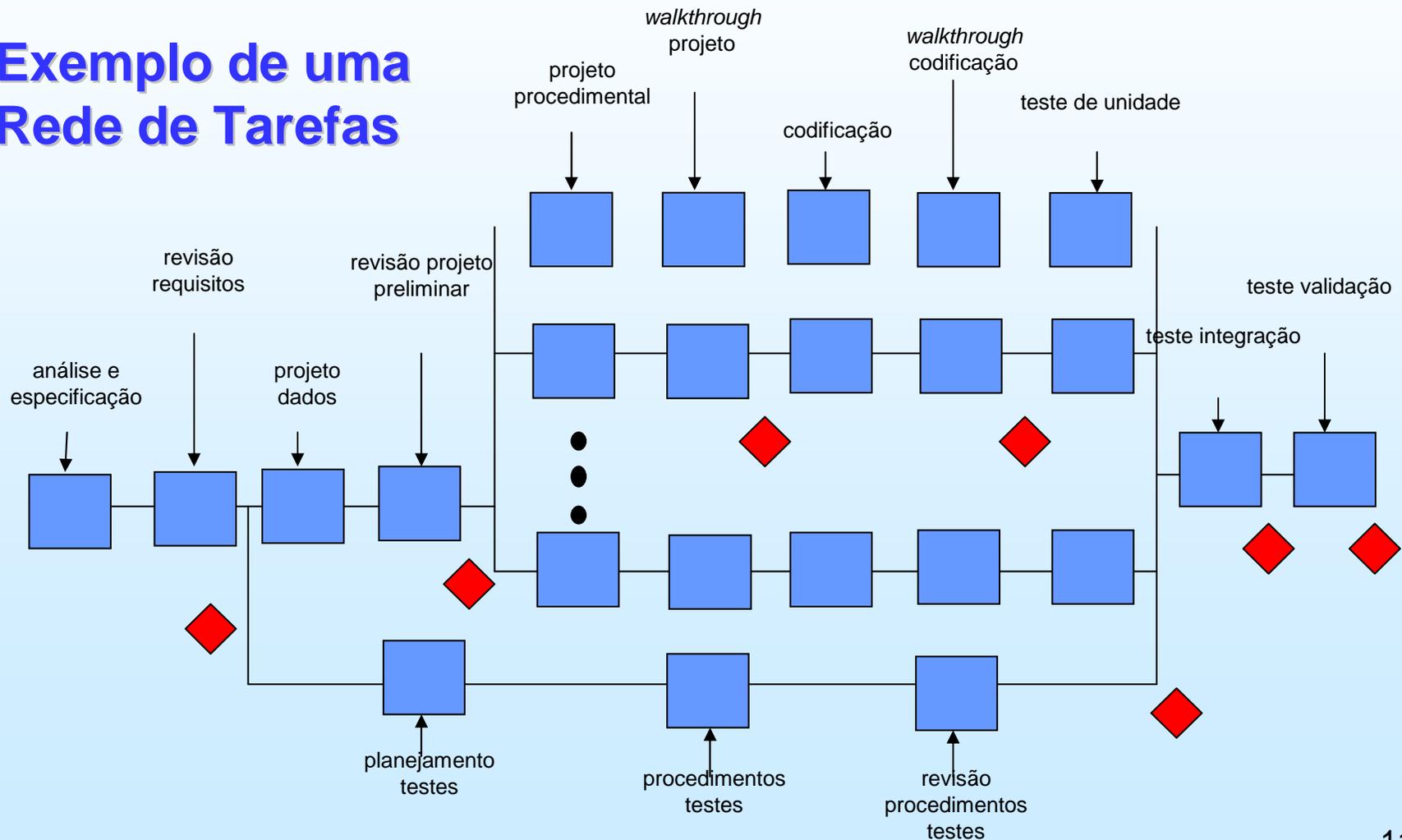
Inter-relacionar as Atividades e Definir Precedências

- ↳ Existem várias técnicas gráficas para representar os interrelacionamentos entre as atividades e definir as precedências
- ↳ A mais consagrada:
 - **a rede de PERT**

Rede PERT

(*P*rogram *E*valuation and *R*eview *T*echnique)

Exemplo de uma Rede de Tarefas



Elaboração do Cronograma

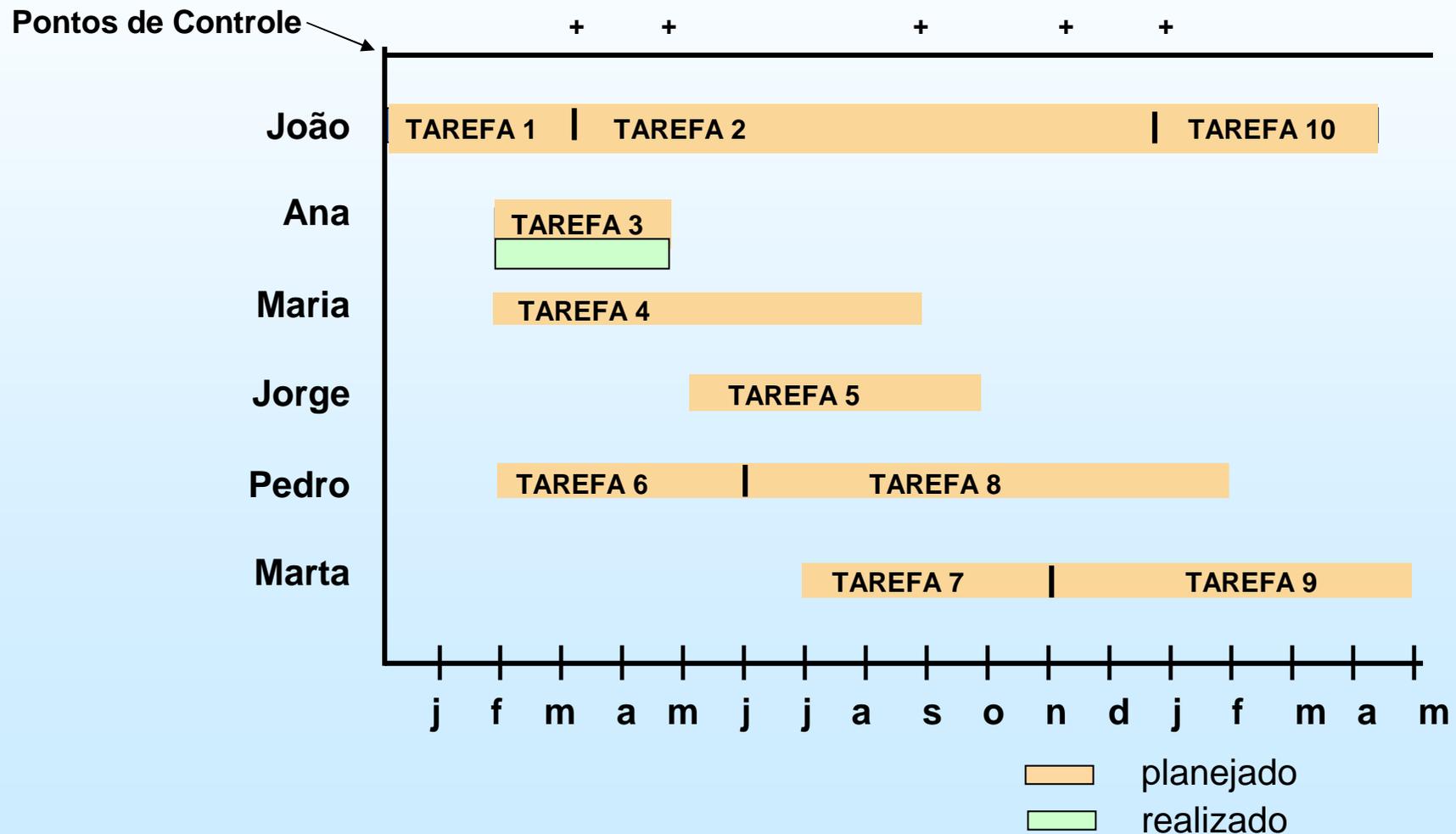
TAREFAS:

1. Identificar e selecionar os recursos para o projeto
2. Inter-relacionar as atividades e definir precedências
3. Calcular o caminho crítico
4. Alocar recursos nas atividades
5. Preparar cronograma do projeto

Preparar Cronograma do Projeto

- ↪ Essa tarefa tem como objetivo apresentar graficamente as datas de início e término de cada atividade, uma vez que os recursos, durações e as interdependências já estão estabelecidas.
- ↪ O cronograma do projeto pode ser apresentado de diferentes formas:
 - Tabelas com listas de atividades
 - Gráficos de *Gantt*,
 - Gráficos de marcas ou etapas, etc

Exemplo de Gráfico de Gantt



Atividades Fundamentais de Planejamento de Projeto

- ↳ Elaboração de Estimativas
- ↳ Análise de Riscos
- ↳ Elaboração de Cronograma
- ↳ Elaboração do Plano e Aprovação

Elaboração do Plano do Projeto

↳ Essa tarefa consiste no preenchimento de todas as seções do plano de projeto.

Esboço do Plano de Projeto de Software

I. Introdução.

1. Escopo e propósito do documento.
2. Objetivos do projeto.
 - a. Objetivos.
 - b. Funções principais.
 - c. Questões de desempenho.
 - d. Restrições técnicas e administrativas.

II. Estimativas de projeto.

1. Dados históricos usados nas estimativas.
2. Técnicas de estimativa.
3. Estimativas.

III. Riscos do projeto.

1. Análise dos riscos.
 - a. Identificação.
 - b. Estimativa dos riscos.
 - c. Avaliação.
2. Administração dos riscos.
 - a. Opções para evitar os riscos.
 - b. Procedimentos de monitoração dos riscos.

IV. Cronograma.

1. *Work breakdown* - divisão de trabalho no projeto.
2. Rede de tarefas.
3. Gráfico de timeline (gráfico de Gantt).
4. Tabela de recursos.

V. Recursos do projeto.

1. Pessoal.
2. Hardware e software.
3. Recursos especiais.

VI. Organização do pessoal.

1. Estrutura de equipe (se for o caso).
2. Relatórios administrativos.

VII. Mecanismos de tracking (rastreamento) e controle.

VIII. Apêndices.

Leituras adicionais

- ↪ 2a. edição do livro de **Shari Pfleeger**
 - Cap. 3 – Planning and Managing the Project
- ↪ 5a. edição do livro de **Pressman**
 - Cap. 3 – Conceitos de Gestão de Projetos
 - Cap. 4 – Métricas de Processo e Projeto de Software
 - Cap. 5 – Planejamento de Projeto de Software
 - Cap. 6 – Análise e Gestão de Risco
- ↪ 6a. edição do livro de **Sommerville**
 - Cap. 4 – Gerenciamento de Projetos
 - Cap. 23 – Estimativa de Custo de Software
- ↪ <http://www.rspa.com/docs/Projectplan.html>
(plano de projeto de software detalhado)