Gerência e Planejamento de Projeto

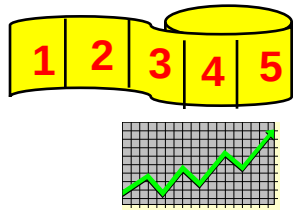
SSC 121 - Engenharia de Software I

Profa. Elisa Yumi Nakagawa

2º semestre de 2012

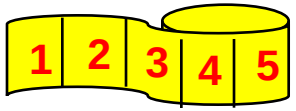
Conteúdo:

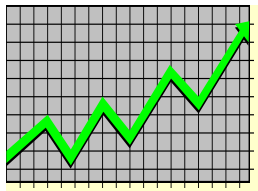
- Parte 1:
  - Gerenciamento & Qualidade
  - Plano de Projeto - aspectos gerais
- Parte 2:
  - Plano de Projeto - Métricas e Estimativas
- Parte 3:
  - Plano de Projeto - Cronograma e Controle



Parte 2 - Objetivos

- Métricas
  - Orientadas ao Tamanho
  - Orientadas à Função
  - de Qualidade
- Estimativas de Projeto
  - Técnicas de Decomposição:
    - Estimativa de LOC e PF
    - Estimativa de Esforço
  - Modelos Empíricos
    - Modelo Estático de Variável Simples
    - COCOMO
  - Ferramentas





Plano de Projeto de Software

I. Introdução
1. Escopo e propósito do documento
2. Objetivos do Projeto

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

III. Riscos do Projeto
1. Análise dos riscos
2. Administração dos riscos

IV. Cronograma
1. Divisão do trabalho (work breakdown)
2. Rede de tarefas
3. 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
2. Relatórios Administrativos

VII. Mecanismos de Controle

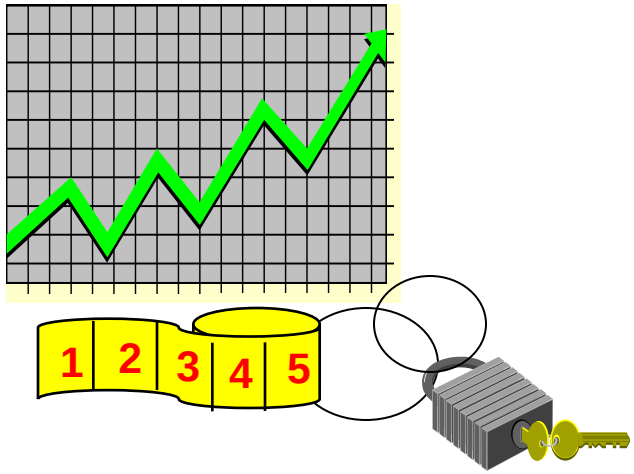
VIII. Apêndices

Plano de Projeto-Estimativas

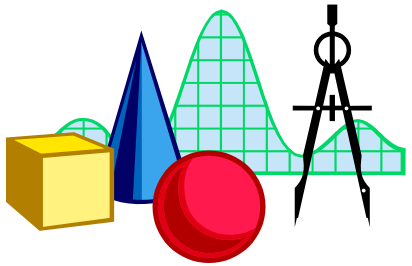
II. ESTIMATIVAS DE PROJETO

MÉTRICAS

TÉCNICAS DE ESTIMATIVAS



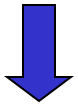
Métricas

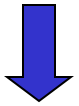


- Razões para se medir o software:
  - Indicar a qualidade do produto
  - Avaliar a produtividade dos que desenvolvem o produto
  - Determinar os benefícios derivados de novos métodos e ferramentas de engenharia de software
  - Formar uma base para as estimativas
  - Ajudar na justificativa de aquisição de novas ferramentas ou de treinamentos adicionais

Métricas

MEDIDAS DO SOFTWARE





MEDIDAS DIRETAS

- Custo
- Esforço
- Linhas de Código
- Velocidade de Execução
- Memória
- Nro de Erros

MEDIDAS INDIRETAS

- Funcionalidade
- Qualidade
- Complexidade
- Eficiência
- Confiabilidade
- Manutenibilidade

Métricas

MÉTRICAS ORIENTADAS AO TAMANHO

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

Exemplos: **LOC - Lines of Code**

KLOC - Thousand Lines of Code

Métricas

MÉTRICAS ORIENTADAS AO TAMANHO LOC/KLOC

projeto	esforço	$	KLOC	pags.docum.	erros	pessoas
projA-01	24	168	12.1	365	29	3
projB-04	62	440	27.2	1224	86	5
projC-03	43	314	20.2	1050	64	6

MÉTRICAS DERIVADAS

PRODUTIVIDADE = KLOC / pessoas-mês

QUALIDADE = erros / KLOC

CUSTO = $ / LOC

DOCUMENTAÇÃO = pags.docum. / KLOC

Métricas

MÉTRICAS ORIENTADAS AO TAMANHO

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

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 Simples	Médio	Complexo	
nro de entradas do usuário		x	3	4	6	
nro de saídas do usuário		x	4	5	7	
nro de consultas do usuário		x	3	4	6	
nro de arquivos		x	7	10	15	
nro de interfaces externas		x	5	7	10	
Contagem-Total	→					

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 incuídas 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

Métricas

DE QUALIDADE

- **corretitude** - grau em que o software executa a função que lhe é exigida
- **manutenibilidade** - grau de facilidade com que o software pode ser corrigido, adaptado ou ampliado
- **integridade** - capacidade que um software tem de suportar *ataques* (acidentais ou intencionais) à sua integridade
- **usabilidade -** tenta quantificar a característica de *user friendliness* do software

Métricas

DE QUALIDADE

- **corretitude** - grau em que o software executa a função que lhe é exigida

ERROS / KLOC é a medida mais comum

os defeitos são registrados pelo usuário depois

que o software foi liberado para uso, e são contados

ao longo de um período de tempo padrão

Métricas

DE QUALIDADE

- **manutenibilidade** - grau de facilidade com que o software pode ser corrigido, adaptado ou ampliado

Tempo médio para mudança

corresponde ao tempo que demora para analisar um pedido de mudança, projetar a modificação adequada, implementar a mudança, testá-la e distribuir para todos os usuários

Métricas

DE QUALIDADE

- **integridade** - capacidade que um software tem de suportar *ataques* (acidentais ou intencionais) à sua integridade

Integridade = $\sum$ (1 - ameaça x (1 - segurança))

ameaça - probabilidade de que um ataque de um tipo específico ocorrerá dentro de determinado tempo

segurança - probabilidade de que o ataque de um tipo específico será repelido

Métricas

DE QUALIDADE

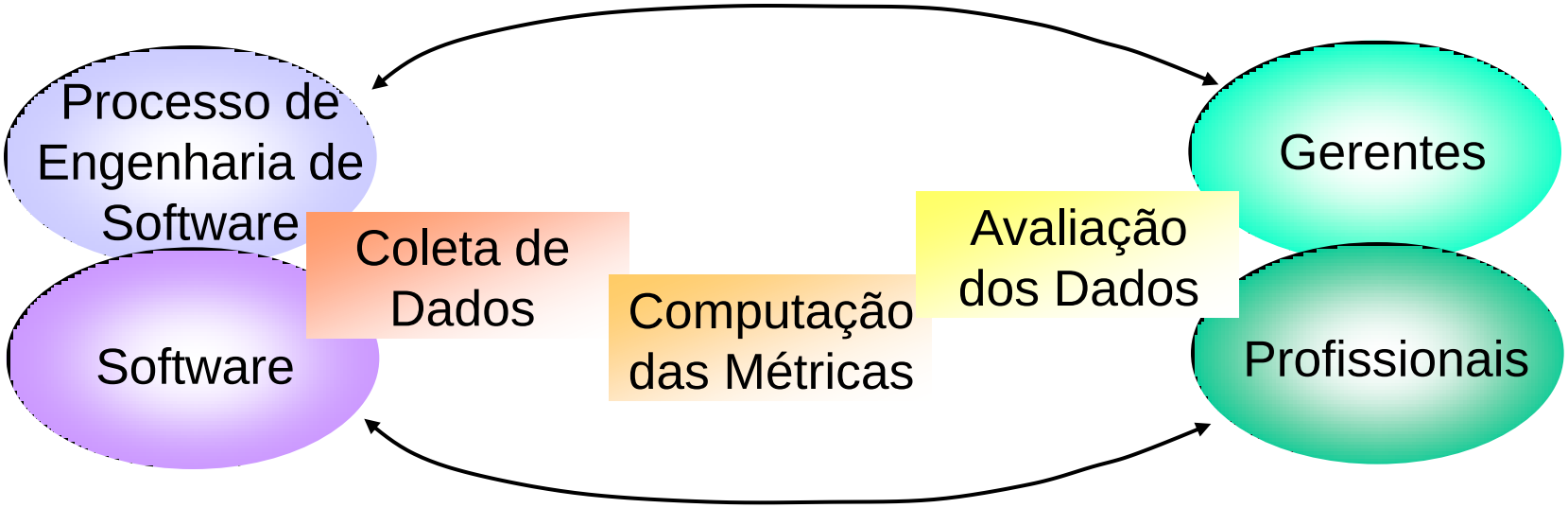
- **usabilidade -** tenta quantificar a característica de *user friendliness* do software

Pode ser medida através de 4 características:

1. habilidade física/intelectual para se aprender o sistema
2. tempo exigido para se tornar moderamente eficiente no uso
3. aumento de produtividade por alguém que seja moderadamente eficiente
4. avaliação subjetiva (questionário)

Métricas

COLETA, COMPUTAÇÃO E AVALIAÇÃO DAS MÉTRICAS



***BASELINE* - DADOS HISTÓRICOS**

Métricas

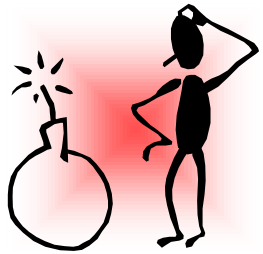
BASELINE - DADOS HISTÓRICOS

- Atributos dos Dados Históricos:
  - Ajudam a reduzir o risco das estimativas
  - Devem ser precisos ou próximos de um valor real
  - Coletados do maior número de projetos possível
  - As medidas devem ser interpretadas da mesma maneira durante todo o projeto
  - As aplicações devem ser similares às do trabalho que se quer estudar

Estimativas

- Constitui um mapa para a bem-sucedida engenharia de software
- Exige:
  - experiência
  - acesso a boas informações históricas
  - coragem para se comprometer com medidas quantitativas quando só existirem dados qualitativos

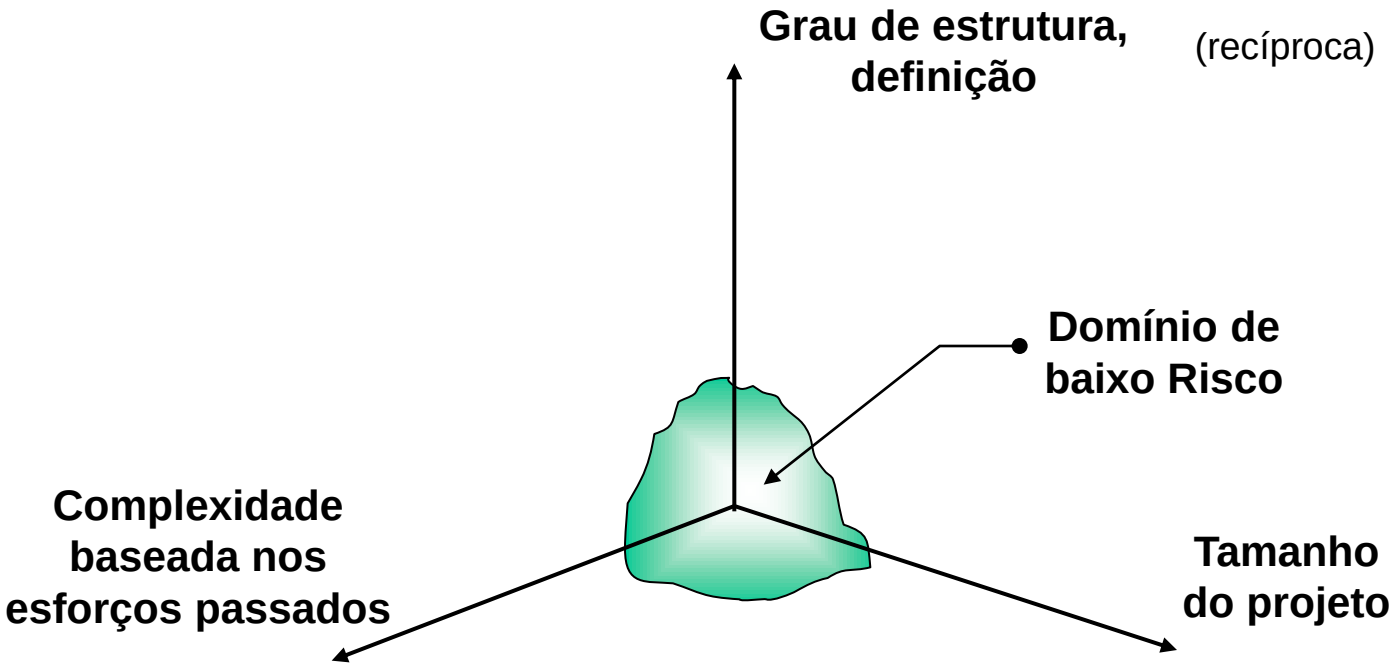
Estimativas



- Estimativas possuem Riscos inerentes
  - os Riscos são medidos pelo grau de incerteza das estimativas estabelecidas para recursos, prazos e custo
  - escopo mal compreendido ou requisitos sujeitos a mudanças $\Rightarrow$ Riscos elevados
  - clientes e gerente devem estar cientes de que variação de requisitos $\Rightarrow$ instabilidade de custo e prazo

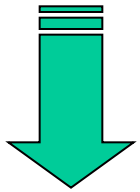
Estimativas

- Fatores que aumentam o risco:
  - complexidade
  - tamanho do projeto
  - grau de estruturação

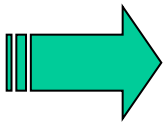


Estimativas

- Fator que reduz o risco:



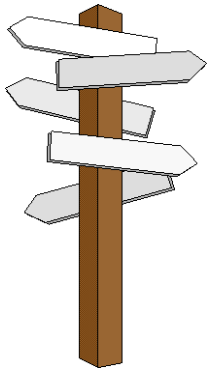
DADOS HISTÓRICOS



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

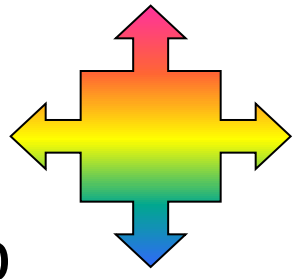
Estimativas

- Não é uma ciência exata
- As Estimativas são afetadas por muitas variáveis:
  - humanas
  - técnicas
  - ambientais
  - políticas



Estimativas

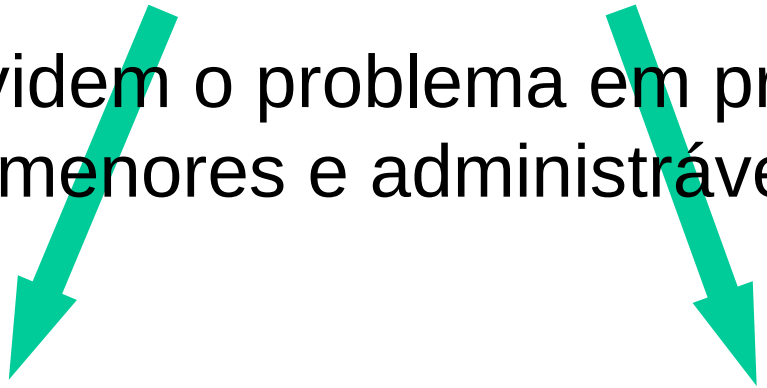
- As opções para se ter Estimativas com graus aceitáveis de Risco:
  - retardar as estimativas do projeto
  - usar técnicas de decomposição (dividir o problema complexo em pequenos problemas)
  - desenvolver um modelo empírico
  - adquirir ferramentas de estimativas



Estimativas

TÉCNICAS DE DECOMPOSIÇÃO

Subdividem o problema em problemas menores e administráveis



Estimativa de LOC e PF

Estimativa de Esforço

Estimativas

TÉCNICAS DE DECOMPOSIÇÃO — Estimativa de LOC e PF

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 ou PF *otimista*, *mais provável*, *pessimista***

	LOC			
Funções	otimista(a)	mais provável(b)	pessimista(c)	Esperado
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	

Estimativas

TÉCNICAS DE DECOMPOSIÇÃO — Estimativa de LOC e PF

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

	LOC			
Funções	otimista(a)	mais provável(b)	pessimista(c)	Esperado
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 ou PF ESTIMADO

Estimativas

TÉCNICAS DE DECOMPOSIÇÃO — Estimativa de LOC e PF

Determinação do Custo e do Esforço:

ABORDAGEM 1

- De Projetos Passados (Dados Históricos) obtém-se:

  Produtividade Média = 596 LOC/pessoas-mês

  Custo Médio = 19,7 $/LOC

- Da última Tabela obteve-se LOC ESTIMADO = 33360

  ESFORÇO = LOC ESTIMADO / Produtividade Média

  ESFORÇO = 33360 / 596 = 56 pessoas-mes

  CUSTO = LOC ESTIMADO x Custo Médio

  CUSTO = 33360 x 19,7 = 656680 $

Estimativas

TÉCNICAS DE DECOMPOSIÇÃO — Estimativa de LOC e PF

Determinação do Custo e do Esforço:

ABORDAGEM 2

- De Projetos Passados (Dados Históricos) obtém-se:

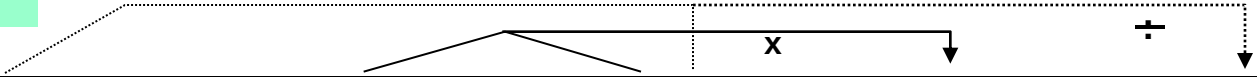
  Diferentes valores de produtividade, de acordo com a complexidade de cada subfunção.

  O **CUSTO** e o **ESFORÇO** são calculados separadamente para cada subfunção.

Estimativas

TÉCNICAS DE DECOMPOSIÇÃO — Estimativa de LOC e PF

ABORDAGEM 2



Funções	LOC/pessoas-mes	$/LOC	LOC Estimado	$	pessoas-mes
função1	315	14	2340	32760	7.4
função2	220	20	5380	107600	24.4
função3	220	20	6800	136000	30.9
função4	240	18	3350	60300	13.9
função5	200	22	4850	108900	24.7
função6	140	28	2140	59920	15.2
função7	300	18	8400	151200	28.0
CUSTO ESTIMADO DO PROJETO				656680	
ESFORÇO ESTIMADO DO PROJETO					144.5

Estimativas

TÉCNICAS DE DECOMPOSIÇÃO

Estimativa de Esforço

1) Decompor o software em funções menores que possam ser estudadas individualmente

2) Usando Dados Históricos ou intuição, estimar para cada subfunção o ESFORÇO (pessoas-mês) necessário em cada etapa da construção do software

	Etapas da Construção				
Funções	Análise Requisitos	Projeto	Codificação	Teste	Totais
função1	1.0	2.0	0.5	3.5	7.0
função2	2.0	10.0	4.5	9.5	26.0
função3	2.5	12.0	6.0	11.0	31.5
função4	2.0	6.0	3.0	4.0	15.0
função5	1.5	11.0	4.0	10.5	27.0
função6	1.5	6.0	3.5	5.0	16.0
função7	4.0	14.0	5.0	7.0	30.0

Estimativas

TÉCNICAS DE DECOMPOSIÇÃO

Estimativa de Esforço

3) Aplicar a TAXA de Trabalho a cada uma das Etapas de Construção do software

4) Calcular o CUSTO e o ESFORÇO para cada função e cada etapa

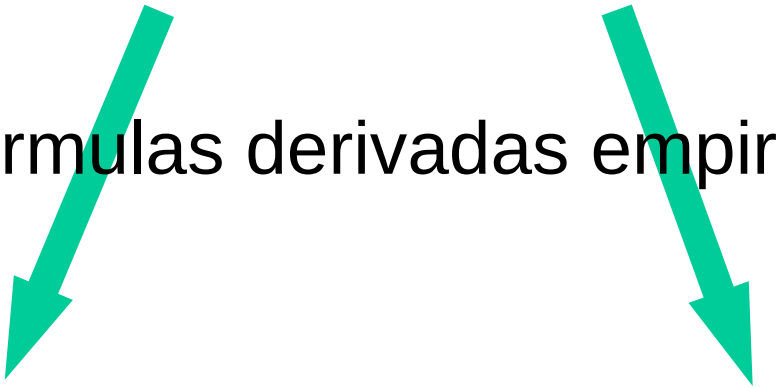
	Etapas da Construção				
Funções	Análise Requisitos	Projeto	Codificação	Teste	Totais
função1	1.0	2.0	0.5	3.5	7.0
função2	2.0	10.0	4.5	9.5	26.0
função3	2.5	12.0	6.0	11.0	31.5
função4	2.0	6.0	3.0	4.0	15.0
função5	1.5	11.0	4.0	10.5	27.0
função6	1.5	6.0			16.0
função7	4.0	14.0	5.0	7.0	30.0
Total	14.5	61.0			152.5
Taxa ($/pessoa-mês)	5.200	4.800	4.250	4.500	
CUSTO	75.400	292.800	112.625	227.250	708.075

ESFORÇO ESTIMADO

CUSTO ESTIMADO

Estimativas

MODELOS EMPÍRICOS



Usam fórmulas derivadas empiricamente

Modelo Estático de Variável Simples

COCOMO

Estimativas

MODELOS EMPÍRICOS

Modelo Estático de Variável Simples

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

ESFORÇO **E = 5.2 x LOC $^{0.91}$ (pessoas-mês)**

DURAÇÃO DO PROJETO **D = 4.1 x LOC $^{0.36}$ (meses)**

TAMANHO DA EQUIPE **S = 0.54 x E $^{0.06}$ (pessoas)**

LINHAS DE DOCUMENTAÇÃO **DOC = 49 x LOC $^{1.01}$**

Modelo de Walston e Felix - constantes derivadas de 60 projetos

Estimativas

MODELOS EMPÍRICOS

COCOMO

(Boehm)

- **Modelo 1:** Modelo COCOMO **Básico**
  - 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 $E = A\,(LOC)^{B}$ **x 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:

$E = A\ (KLOC)^B$

$E = 3.0\ (KLOC)^{1,12}$

$= 3.0\ (33.3)^{1,12}$

$= 152$ pessoas-mes

$T = C\ (E)^D$

$T = 2.5\ (E)^{0.35}$

$= 2.5\ (152)^{0.35}$

$= 14.5$ meses

Com esses valores é possível determinar um número recomendado de pessoas

$N = E/T$

$= 152/14.5$

$= 11$ pessoas

Estimativas

FERRAMENTAS AUTOMATIZADAS

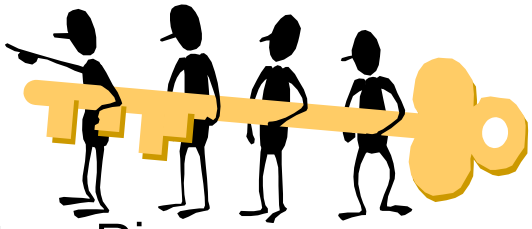
As Técnicas de Decomposição e os Modelos Empíricos de Estimativas podem ser implementados em software.

Esses softwares exigem os seguintes tipos de dados:

- estimativas quantitativas do tamanho ou funcionalidade do software (LOC ou PF)
- características qualitativas do projeto (complexidade, confiabilidade exigida, etc.)
- descrição do pessoal de desenvolvimento e/ou ambiente de trabalho (experiência, motivação, etc.)

Pontos-Chaves

- Quanto às Métricas:
  - Sem medir, não há maneira de determinar se existe melhoria
  - A medição resulta em mudança cultural
  - Ao criar uma *baseline* (banco de dados contendo medições do processo e do produto), engenheiros e gerentes podem ter uma melhor visão do processo e do produto



- Quanto às Estimativas:
  - Não constituiem 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

Tabelas de Conversão

- http://www.qsm.com/FPGearing.html