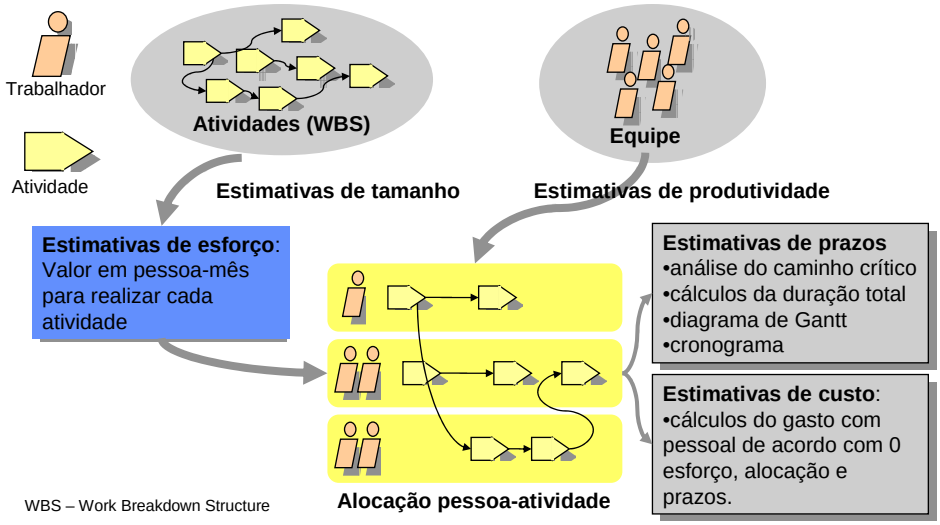
Estimativas de software

- Fazer boas estimativas é uma das mais desafiadoras e importantes atividades da engenharia de software.
- Estimativas de custos
  - ferramentas (H/S) e infra-estrutura
  - pessoal (salários e encargos mensais)
  - dependente dos prazos
- Estimativas de prazos
  - dependente das atividades
  - dependente de pessoal
- Estimativas de esforço
  - Medida que associa atividades a serem realizadas com o pessoal necessário em um certo período de tempo.

Estimativas de software



WBS – Work Breakdown Structure

Estimativas e Métricas

- Estimativas são baseadas em métricas históricas e empíricas
- Métricas históricas
  - Obtidas a partir de experiências anteriores da equipe
- Métricas empíricas
  - Dados estatísticos de diferentes equipes

Métricas

- Planejamento, Gerenciamento e Avaliação são realizados com base em **métricas**
- A medição possibilita
  - Avaliar a qualidade dos produtos
  - Avaliar a produtividade da equipe
  - Avaliar métodos e ferramentas
  - Realizar estimativas no planejamento
- Métricas do processo
  - Métricas de produtividade
- Métricas do produto
  - Métricas da qualidade e métricas técnicas

Métricas para Planejamento e Gerenciamento

- Objetivos
  - Dimensão dos produtos
    - Modelos, protótipos, documentos e software
  - Esforço de produção
    - Pessoas necessárias num período de tempo
  - Produtividade
    - Quantidade produzida por esforço
  - Defeitos
    - Número de erros encontrados
  - Custo de produção
    - Valor do esforço de produção e correção de erros

Principais Métricas

Métricas	**Objetivos**
Linhas-de-Código (LOC)	Dimensão do produto
Pontos-por-função (FP)	Dimensão do produto
LOC/FP	Dimensão do produto
Pessoa-Mês (PM)	Esforço humano
Pessoa-Mês/LOC	Produtividade linear
Defeitos/LOC	Qualidade
Custo/LOC	Custo

Tamanho = LOC e Ponto-por-Função

- Métricas relacionadas a tamanho do código
  - Linhas de código fonte (LOC ou SLOC)
  - Mede todo o esforço necessário para entregar um código correto, sem erros.
  - Depende da linguagem
- Métricas relacionadas a funções
  - Determinadas pela funcionalidade do sistema.
  - Independente de linguagem
- Em qualquer abordagem é necessário:
  - ter experiência
  - e utilizar dados históricos

Esforço humano

- Determina o número de pessoas para realizar uma atividade num período de tempo
- Exemplo:
  - Construir um programa com 100 KLOC
  - Produtividade linear dos programadores:
    - 2,9 Pessoa-mês/KLOC
  - O esforço depende do tamanho do software

Tamanho da equipe	Esforço	Prazo total
1	290 P-M	290 meses
10	290 P-M	29 meses
100	290 P-M	2,9 mês

Atenção: a produtividade diminui em equipes maiores, principalmente devido à necessidade de comunicação e integração.

Métricas e linguagens

- O número de linhas de códigos e a produtividade variam de acordo com a linguagem utilizada

Linguagem de programação	LOC por FP
C++	53
Cobol	107
Delphi 5	18
HTML 4	14
Visual basic 6	24
SQL	13
Java	46

Métricas e Sistemas

- Produtividade (Pessoa-mês/KLOC) varia de acordo com o tipo de sistema a ser desenvolvido
- COCOMO – Constructive Cost Model – modelo para estimativa de custos de B. Boehm

Tipo de Sistema	Produtividade
Default (usando COCOMO II)	2,94
Sistema embutido	2,58
Comércio eletrônico	3,60
Sistema Web	3,30
Sistema Militar	2,77

Abordagens para estimativas de esforço

- Julgamento de especialista – Técnica Delphi
  - Uso da experiência de desenvolvedores
  - Uso métricas históricas
- Estimativas por Analogia
  - Uso de métricas históricas de projetos anteriores similares
- Métodos algoritmos
  - COCOMO e COCOMO II

Estimativas top-down e bottom-up

- Podem ser utilizadas com qualquer abordagem
- Top-down
  - Usada quando não se tem uma arquitetura do software
  - Leva em consideração atividades globais – documentação, gerenciamento, testes integrados
  - Falha nas estimativas de atividades relacionadas a detalhes técnicos.
- Bottom-up
  - Divisão do software em unidades menores arquitetura do software
  - Elaboração da Estrutura de Divisão do Trabalho (WBS)
  - Estimativa para unidades menores são mais precisas
  - Pode subestimar esforços para a integração das unidades
    - “O todo não é apenas a soma das partes”.

Julgamento por especialistas

- Arquitetura do software e divisão do trabalho
- Estimativas de tempo para:
  - problemas antigos (A) e novos (N)
  - problemas fáceis (F), moderados (M), difíceis (D)
- Mais de um especialista (normalmente três)

TABELA 3.6 Crítico quanto ao tempo

	Dificuldade					
Tipo de software	AF	AM	AD	NF	NM	ND
Controle	21	27	30	33	40	49
Entrada/saída	17	24	27	28	35	43
Pré/pós processador	16	23	26	28	34	42
Algoritmo	15	20	22	25	30	35
Gerenciamento de dados	24	31	35	37	46	57
Crítico quanto ao tempo	75	75	75	75	75	75

Métodos algoritmicos

- O custo é estimado matematicamente como uma função:
  - do produto, do projeto e do processo
  - Esforço = $A \times Tamanho^B \times M$
    - A é uma constante que depende da organização que desenvolve
    - B ajusta o valor, aplicando penalidades, em função do tamanho do projeto
    - M é um multiplicador associado a atributos de pessoas, produto e processo
- O tamanho é a variável fundamental
- Modelos diferentes apresentam variações para os valores de A, B e M

O modelo COCOMO

- Modelo empírico baseado na experiência de projetos existentes
- Versão inicial em 1981 (COCOMO-81) e várias versões até o COCOMO 2

Project complexity	Formula	Description
Simple	$PM = 2.4\ (KDSI)^{1.05} \times M$	Well-understood applications developed by small teams.
Moderate	$PM = 3.0\ (KDSI)^{1.12} \times M$	More complex projects where team members may have limited experience of related systems.
Embedded	$PM = 3.6\ (KDSI)^{1.20} \times M$	Complex projects where the software is part of a strongly coupled complex of hardware, software, regulations and operational procedures.

COCOMO-81, fonte: Ian Sommerville

COCOMO 2

- COCOMO 2 é um modelo de três níveis
  - Nível inicial de prototipação
    - Estimativas baseadas em pontos de objetos e fórmula simples
  - Nível inicial de projeto
    - Estimativas baseadas em pontos de função (FP) que são traduzidas para linhas-de-código (LOC)
  - Nível pós-arquitetura
    - Utiliza linhas-de-código e atributos de produtividade
- Permite melhoria das estimativas de acordo com o progesso do desenvolvimento

Nível inicial de prototipação

- Permite estimativas de prototipação com reuso
- Uso de ferramentas CASE e linguagens de quarta geração
- Fórmula:
  - PM = (NOP × (1 - %reuso/100)) / PROD
    - PM – esforço em pessoa-mês
    - NOP – número de pontos de objetos
    - PROD – produtividade
- Valores de produtividade:

Developer's experience and capability	Very low	Low	Nominal	High	Very high
ICASE maturity and capability	Very low	Low	Nominal	High	Very high
PROD (NOP/month)	4	7	13	25	50

Nível inicial de projeto

- Estimativas neste nível são feitas quando os requisitos estão definidos
- Fórmula:
  - PM = A × Tamanho^B × M + PM_m onde
  - M = PERS × RCPX × RUSE × PDIF × PREX × FCIL × SCED
  - PM_m = (ASLOC × (AT/100)) / ATPROD
  - A = 2.5 – podendo ser ajustado
  - B varia de 1.1 a 1.24 dependendo de
    - Originalidade, flexibilidade, riscos e maturidade.

Multiplicadores

- Valores de 1 (baixo) a 6 (alto) para
  - RCPX – confiabilidade e complexidade do produto
  - RUSE – reuso requerido
  - PDIF – dificuldade de plataforma
  - PREX – experiencia do pessoal
  - PERS – capacidade do pesoal
  - SCED – prazo requerido
  - FCIL – recursos de suporte
- PM é p esforço requerido para a geração automática de código

Nível pós-arquitetura

- Mesma formula anterior
- Valor de tamanho ajustado – mais preciso
  - ESLOC = ASLOC × (AA + SU +0.4DM + 0.3CM +0.3IM)/100