Introdução à Engenharia de Software

Profa. Maria Auxiliadora

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Objetivos

- Compreender o que é Engenharia de Software e por que ela é importante.
- Mostrar as distinções e relacionamentos entre sistema e software.
- Mostrar as principais questões sobre engenharia software.
- Compreender questões profissionais e éticas relevantes para os engenheiros de software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Por que?

- Por que tanta demora para entregar o sistema?
- Por que os prazos se atrasam?
- Por que os custos são altos?
- Por que não achar todos os erros antes de entregar?
- Por que é difícil medir o progresso do desenvolvimento de um software?

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Desafio

No mercado atual, não há dúvida de que os profissionais de TI envolvidos com projetos de desenvolvimento de software e soluções corporativas têm um claro desafio:

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Desafio

PRODUZIR soluções mais rápidas, melhores e mais baratas que antes (melhor ainda ser mais rápidas, melhores e mais baratas que a concorrência)

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Desafio

- Desenvolver software é um problema de métodos e técnica, em suma, de atividade humana.

- Gerenciar projetos de TI é gerenciar riscos assumidos e vencidos por seres humanos inteligentes.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Evolução do Software

Período	Evolução
1950-1960	Orientação a batch
	Software totalmente customizado
	Distribuição limitada
1960-1970	Multiusuários
	Tempo Real
	Banco de Dados

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Desafio

Período	Evolução
1980-1990	Sistemas distribuídos
	Inteligência Embutida
	Hardware de baixo custo
1990-2000	Sistemas de desktop poderosos
	Tecnologia orientada a objeto
	Sistemas Especialistas....

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Desafio

Período	Evolução
2000 - atual	As tecnologias orientadas a objetos
	Uso das técnicas de "quarta geração" para o desenvolvimento de software
	Os sistemas especialistas e o software de inteligência artificial.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Aplicação do Software

Básico	Coleção de programas escritos para apoio a outros programas.
Tempo Real	Software que monitora / analisa / controla eventos do mundo real.
Comercial	Processa informações comerciais, reestruturação de dados para tomada de decisões.
Científico e de engenharia	Algoritmos de processamento de números(astronomia,vulcanologia).

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Aplicação do Software

Embutido	usado para controlar produtos e sistemas para os mercados industriais e de consumo.
Computador Pessoal	Processamento de textos, planilhas, computação gráfica.
Inteligência Artificial	faz uso de algoritmos não numéricos para resolver problemas que não sejam favoráveis à computação ou à análise direta.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

O que é Sistema?

é um conjunto de elementos inter-dependentes que realizam operações visando atingir metas especificadas.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Sistema de Computação

é aquele destinado ao suporte ou automação de tarefas através de processamento de informações.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Componentes de Sistemas de Computação

Hardware	Computadores,periféricos e redes.
Software	Os programas e arquivos de dados.
Usuários	Usuários e operadores que realizam as tarefas e procedimentos.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Componentes de Sistemas de Computação (cont.)

Procedimentos	Atividades realizadas pelos usuários e operadores, bem como pelos programas.
Documentação	Manuais e formulários que descrevem as operações do sistema.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Exemplos de Sistemas Computacionais

- Automação Bancária
- Frequência e Folha de Pagamento
- Controle de Tráfego Urbano
- Controle Acadêmico
- Editoração de Jornais e Revistas
- Controle de Elevadores
- Automação de Biblioteca

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

O que é Software?

- Programas de computadores associados a documentação.

- É um conjunto de soluções algorítmicas, codificadas numa linguagem de programação, executado numa máquina real.

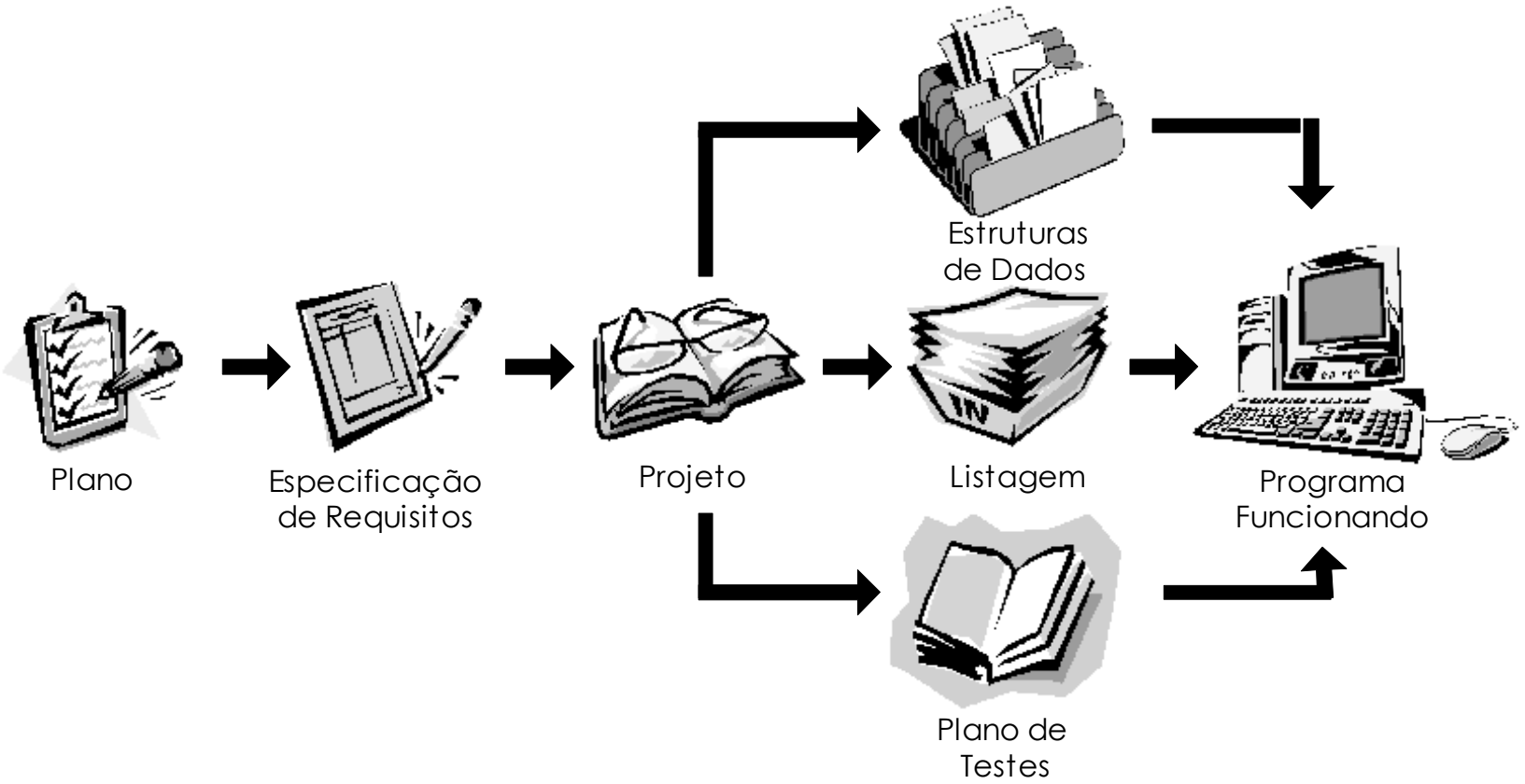
Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Tipos de produtos de software

- **Genéricos** (COTS – Commercial Off-The Shelf) - tipo stand-alone, pacotes de software, como por exemplo, processadores de texto, ferramentas de gerenciamento.
- **Sob encomenda** (personalizado) – desenvolvido para um cliente em particular.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Componentes do Software



Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Características do Software

- Complexidade
- Conformidade
- Mutabilidade

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Características do Software

- **Complexidade**
  - Software é mais complexo do que qualquer outro produto construídos por seres humanos.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Características do Software

- **Conformidade**
  - O software deve ser desenvolvido conforme o ambiente. Não é o ambiente que deve se adaptar ao software.
  - Desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico (industrial).
  - Sucesso é medido pela qualidade e não quantidade.

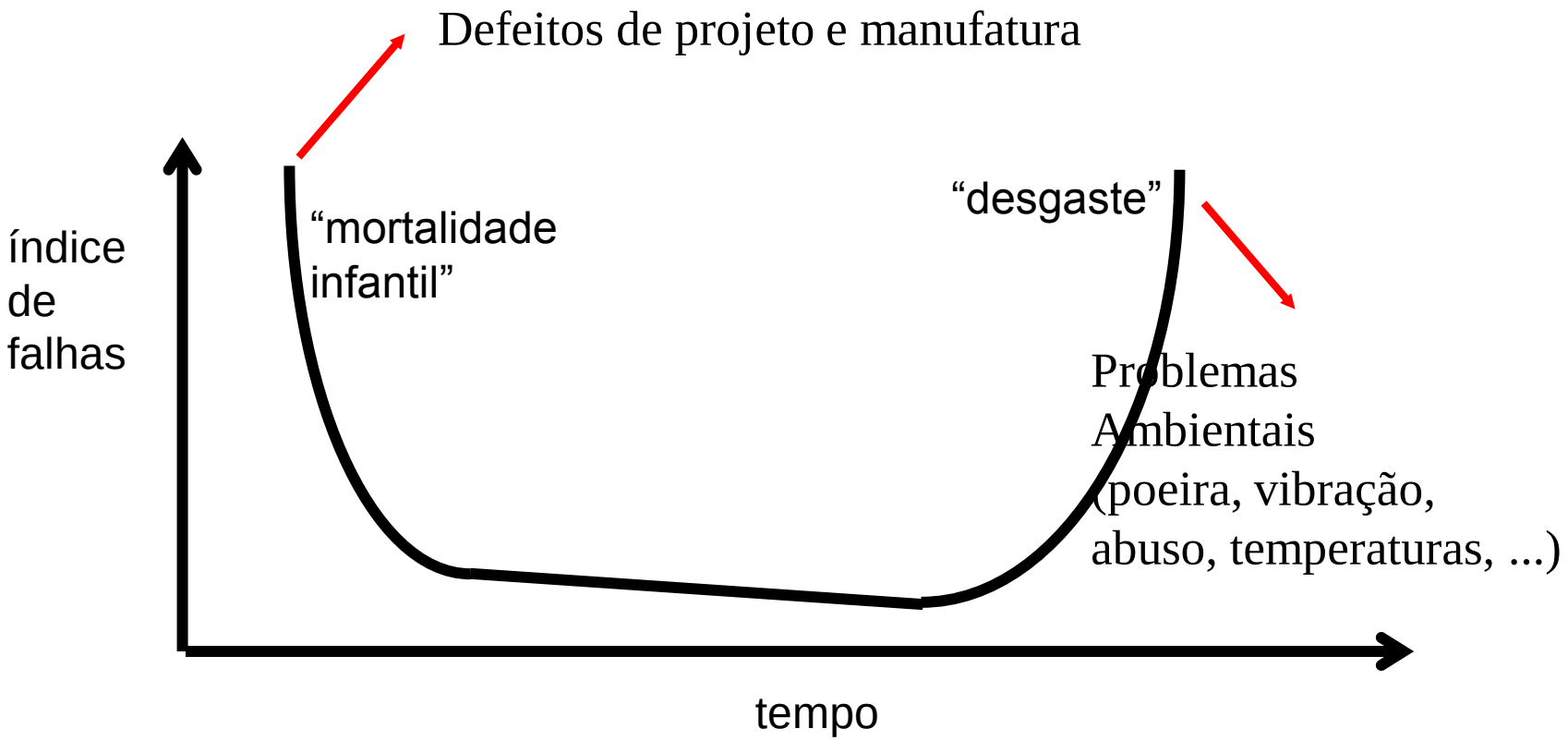
Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Características do Software

- **Mutabilidade**
  - Existe sempre uma pressão para se fazer mudanças em um software.
  - Não se “desgasta”, mas se deteriora devido as mudanças.
  - A maioria é feita sob medida em vez de ser montada a partir de catálogos de componentes existentes (reusabilidade de software).

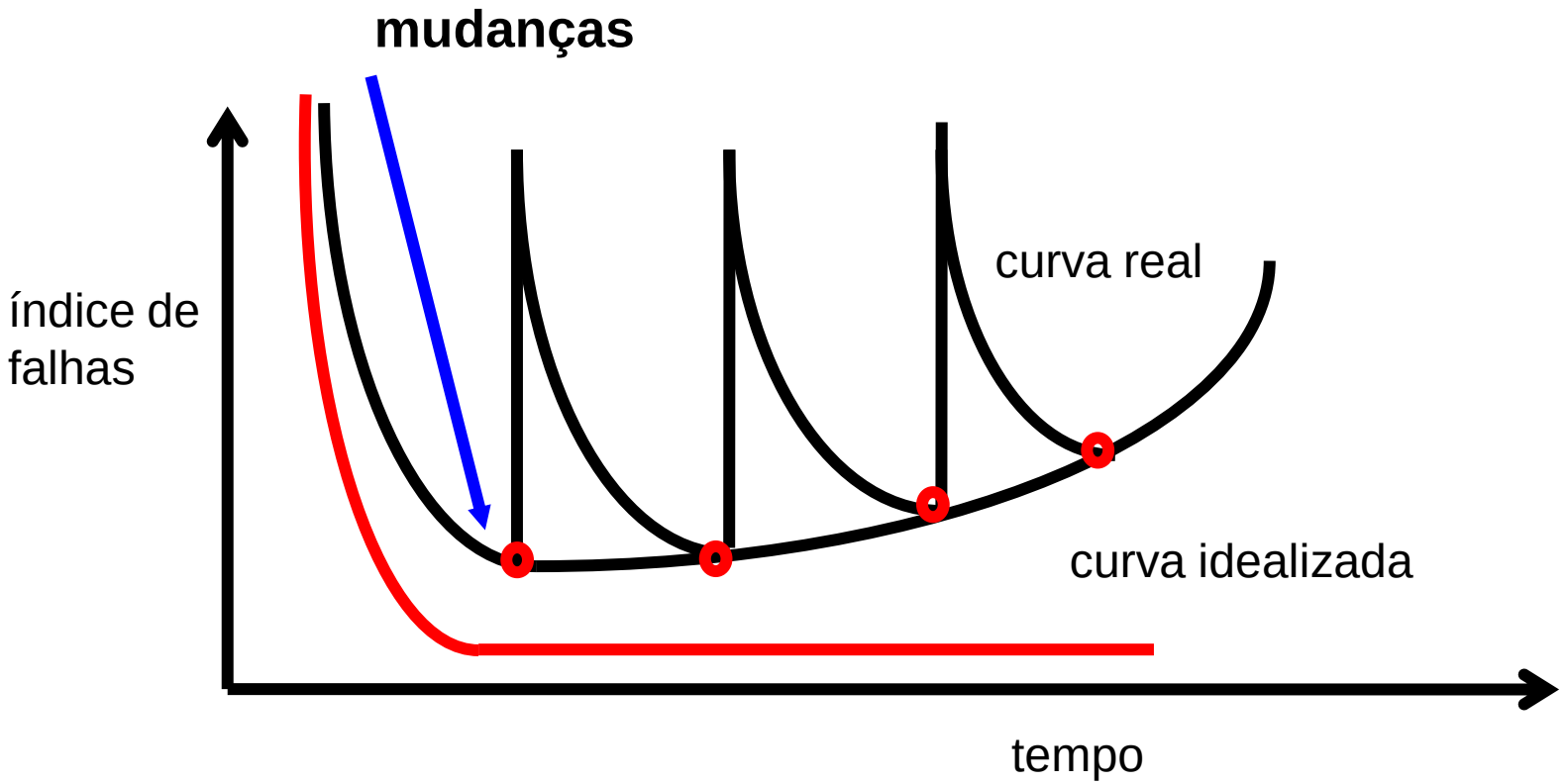
Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Falhas do Hardware



Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Falhas do Software

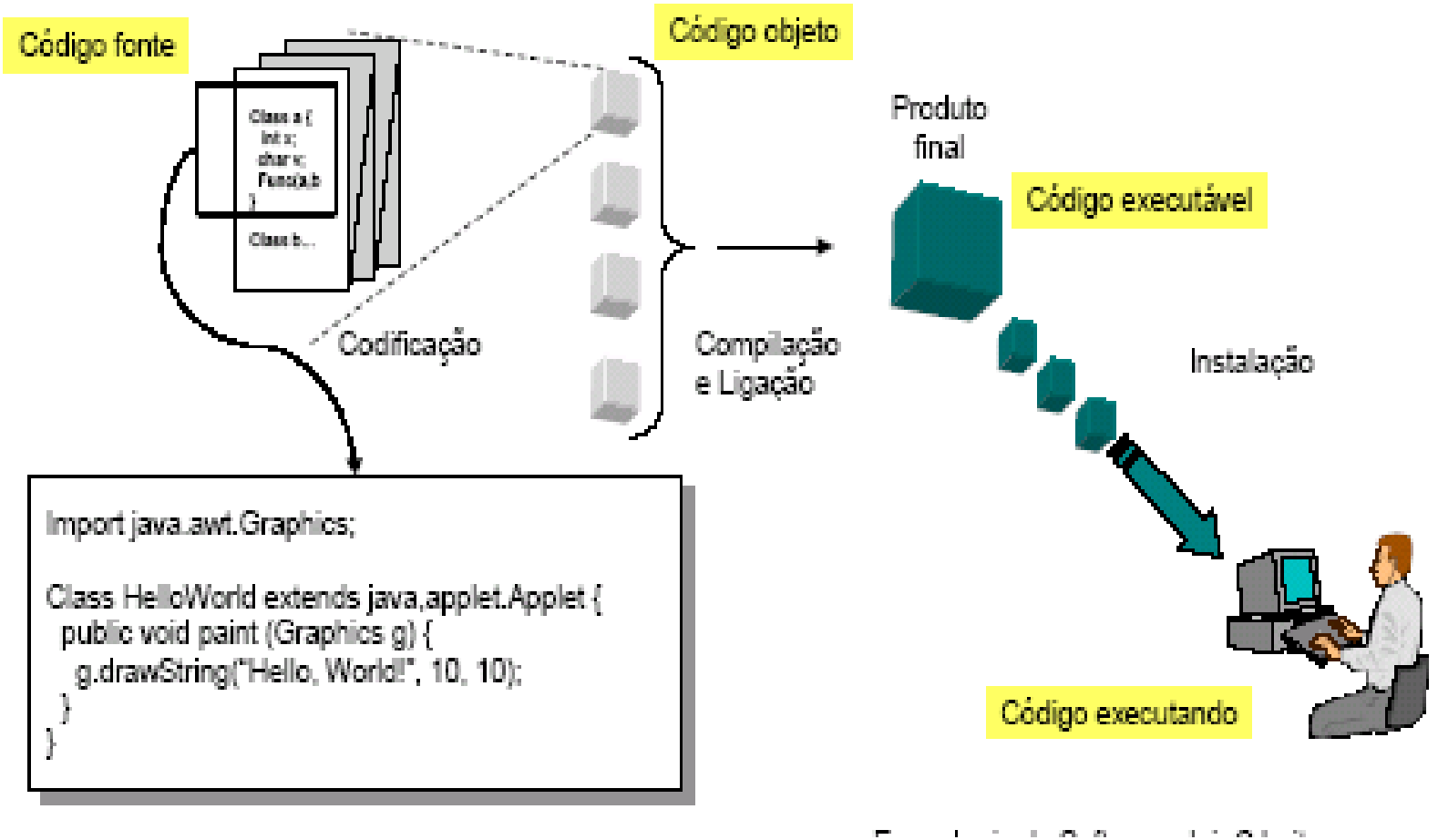


Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Falhas do Hardware/Software

- Quando um componente de hardware se desgasta é substituído por uma “peça de reposição”
- Não existe “peça de reposição” para software
  - Toda falha indica um erro no projeto ou no processo de tradução para o código executável
  - Manutenção do software é mais complexa do que a do hardware

Formas do Software



Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Crise do Software

- Refere-se a um conjunto de problemas encontrados no desenvolvimento de software.

- Problemas não se limitam a softwares que não funcionam adequadamente.

Crise do Software

- Prazos ultrapassados
- Custos acima do previsto
- Não atendimento dos requisitos do usuário

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Crise do Software

- **Elevado custo de manutenção**
  - 1/3 dos projetos são cancelados
  - 2/3 dos projetos extrapolam o orçamento
- **Custo hardware x software**
  - 1970 = 8:2
  - 1991 = 2:8
  - Hoje = 1:9

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Crise do Software

- **As estimativas de prazo e de custo são imprecisas:**
  - Não dedicamos tempo para coletar dados sobre o processo de desenvolvimento de software
  - Estimativas são feitas a olho, com resultados ruins
  - Os prazos arrastam-se por meses

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Crise do Software

- **As estimativas de prazo e de custo são imprecisas (cont.)**
  - Causa insatisfação para o cliente e falta de confiança.
  - Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Crise do Sofware

- **A produtividade das pessoas da área de software não tem acompanhado a demanda por seus serviços**

  – Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Crise do Sofware

- **O software existente é muito difícil de manter:**
  - A tarefa de manutenção devora o orçamento destinado ao software
  - A facilidade de manutenção não foi enfatizada como um critério importante.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Crise do Sofware

Solução para a Crise do Sofware ➔

Engenharia de Software aliada a:

TÉCNICAS E FERRAMENTAS

Causas dos problemas associados à crise de software

- Filosofia do Software.
- Falhas das Pessoas Responsáveis pelo Desenvolvimento de Software.
- Mitos do Software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Filosofia do Software

O software é um elemento de sistema lógico e não físico.

Consequentemente, o sucesso é medido pela qualidade de uma única entidade e não pela qualidade de muitas entidades manufaturadas.

Falhas das Pessoas Responsáveis pelo Desenvolvimento de Software

- Gerentes sem nenhum *background* em software.
- Os profissionais da área de software têm recebido pouco treinamento formal em novas técnicas para o desenvolvimento de software.
- Resistência a mudanças.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos do Software

Mitos do Administrativos, do Cliente e do Profissional

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Problemas Administrativos

- **Gerentes se encontram sob pressão**
  - manter orçamentos
  - evitar que os prazos sejam ultrapassados
  - melhorar a qualidade

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos Administrativos

- **Mito:** Já temos um manual repleto de padrões e procedimentos para a construção de software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos Administrativos

- **Realidade:**
  - Será que o manual é usado?
  - Os profissionais sabem que ele existe?
  - Ele reflete a prática moderna de desenvolvimento de software?
  - Ele é completo?

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos Administrativos

- **Mito:** Meu pessoal tem ferramentas de desenvolvimento de software de última geração; afinal compramos para eles os mais novos computadores.

Mitos Administrativos

- **Realidade:**
  - —É preciso muito mais do que os mais recentes computadores para se fazer um desenvolvimento de software de alta qualidade.
  - —Ferramentas de Engenharia de Software Auxiliada por Computador - **CASE (Computer-Aided Software Engineering)** são mais importantes do que o hardware.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos Administrativos

- **Mito:**

Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar o atraso.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos Administrativos

- **Realidade:**
  - O desenvolvimento de software não é um processo mecânico igual à manufatura.
  - Acrescentar pessoas em um projeto torna-o ainda mais atrasado. Pessoas podem ser acrescentadas.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos dos Clientes

- **Mito:** Uma declaração geral dos objetivos é suficiente para se começar a escrever programas - podemos preencher os detalhes mais tarde.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos dos Clientes

- **Realidade:**
  - Uma definição inicial ruim é a principal causa de fracassos dos esforços de desenvolvimento de software.
  - É fundamental uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos dos Clientes

- **Mito:** Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos dos Clientes

- **Realidade**:
  - Requisitos podem ser mudados, mas o impacto varia de acordo com o tempo que é introduzido (projeto e custo).
  - Uma mudança, quando solicitada tardiamente num projeto, é mais dispendiosa do que a mesma mudança solicitada nas fases iniciais.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos do Profissional

- **Mito:** Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos do Profissional

- **Realidade**:
  - Os dados da indústria indicam que entre 50 e 70% de todo esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos do Profissional

- **Mito:** Enquanto não tiver o programa "funcionando", eu não terei realmente nenhuma maneira de avaliar sua qualidade.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos do Profissional

- **Realidade**:
  - Mecanismo (Revisão Técnica Formal) de garantia de qualidade de software é aplicado desde o começo do projeto.
  - Revisões de software são um “filtro de qualidade” - descobre erros/defeitos.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos do Profissional

- **Mito:** A única coisa a ser entregue em um projeto bem sucedido é o programa funcionando.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Mitos do Profissional

- **Realidade:**
  - Um programa funcionando é somente uma parte de uma Configuração de Software que inclui todos os itens de informação produzidos durante a construção e manutenção do software.

A DOCUMENTAÇÃO é o alicerce

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Categorias de Tamanho de Softwares

Categoria	Tamanho da Equipe	Duração	Tamanho do Fonte (linhas de código)
Trivial	1	1-4 semanas	500
Pequeno	1	1-6 meses	1000 a 2000
Médio	2-5	1-2 anos	5 mil a 50 mil
Grande	5-20	2-3 anos	50 mil a 100 mil
Muito grande	100-200	4-5 anos	1 milhão
Extremamente grande	2000-5000	5-10 anos	1 a 10 milhões

ex: O Win 95: teve 11 milhões de linhas e 200 programadores /
O Nestscape teve 3 milhões de linhas e 120 programadores

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Solução

- Reconhecer os problemas e suas causas e desmascarar os mitos do software são **os primeiros passos**
- **Métodos e Técnicas** para disciplinar o processo de desenvolvimento do software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

SOLUÇÃO

- Reconhecer os problemas e suas causas e desmascarar os mitos do software são **os primeiros passos**
- **Métodos e Técnicas** para disciplinar o processo de desenvolvimento do software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Aspectos históricos da Engenharia de Software

- 1968 Conferência da OTAN
- Objetivo: resolver a **"Crise do Software"**
- Software é entregue
  - Atrasado
  - Com orçamento estourado
  - Com falhas residuais

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Aspectos históricos da Engenharia de Software

- Eles não encontraram uma solução, mas definiram uma meta: **Engenharia de Software.**

- Custo do hardware decrescente e custo do software em ascensão.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

O que é Engenharia de Software?

- **Fritz Bauer – 1969 (primeira definição)**

"O estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais"

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

O que é Engenharia de Software?

- **IEEE, 1993**

"A aplicação de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção do software. O estudo de abordagens e princípios a fim de obter economicamente softwares confiáveis e que executem de forma eficiente nas máquinas reais"

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

O que é Engenharia de Software?

Programador (técnicas)	Engenheiro (técnicas)
1. Paradigma de tentativa e erro	1. Paradigma adaptado ao escopo do sistema
2. Estrutura de Dados	2. Análise e Projeto
3. Linguagens de Programação	3. Ferramentas CASE e SGBD´s

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

O que é Engenharia de Software?

- É uma disciplina que integra **métodos, ferramentas** e **procedimentos** para o desenvolvimento de software de computador.
- Possibilitar ao gerente o controle do processo de desenvolvimento.
- Oferecer ao profissional uma base para a construção de software de alta qualidade.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Engenharia de Software - Método

- Proporcionam os detalhes de **"como fazer"** para construir o software.
- Envolvem um amplo conjunto de tarefas.
- Um **método** de ES é uma aproximação estruturada para o desenvolvimento de software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Engenharia de Software - Método

- Todos os **métodos** pretendem
  - Criar modelos do sistema que possam ser representados graficamente;
  - Usar estes **métodos** como especificação.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Engenharia de Software - Método

- **Década de 70** (DeMarco e Jackson):
  - Orientado à função.
- **80s-90s** (Booch e Rumbaugh):
  - Métodos orientados a objeto.
- Atualmente os diferentes métodos estão integrados numa aproximação unificada baseada em **Unified Modeling Language (UML).**

Engenharia de Software - Método

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Engenharia de Software - Método

- **Métodos** devem incluir os seguintes componentes:
  - Descrição gráficas
  - Regras
  - Recomendações
  - Diretrizes de processo

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Engenharia de Software - Método

- **Descrição gráficas.**

  Descrições dos modelos do sistema que deverão ser desenvolvidos e da notação usada para os definir.

Ex. Modelos de objetos, fluxos de dados etc.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Engenharia de Software - Método

- **Regras**

  Restrições que se aplicam a modelos de sistema.

  Ex.Cada entidade deve ter um único nome.

- **Recomendações**

  Conselho em prática de projeto.

  Ex.Nenhum objeto deve ter mais que sete subobjetos.

Engenharia de Software - Método

- **Diretrizes de processo**
  - Descrição das atividades que podem ser seguidas.
  - Atributos de objetos devem ser documentados.

Engenharia de Software - Ferramentas

- Fornecem suporte automatizado ou semi- automatizado aos métodos.
- Existem atualmente ferramentas para sustentar cada um dos métodos.
- Quando as ferramentas são integradas é estabelecido um sistema de suporte ao desenvolvimento de software chamado *CASE.*

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

O que é CASE
(Computer-Aided Software Engineering)

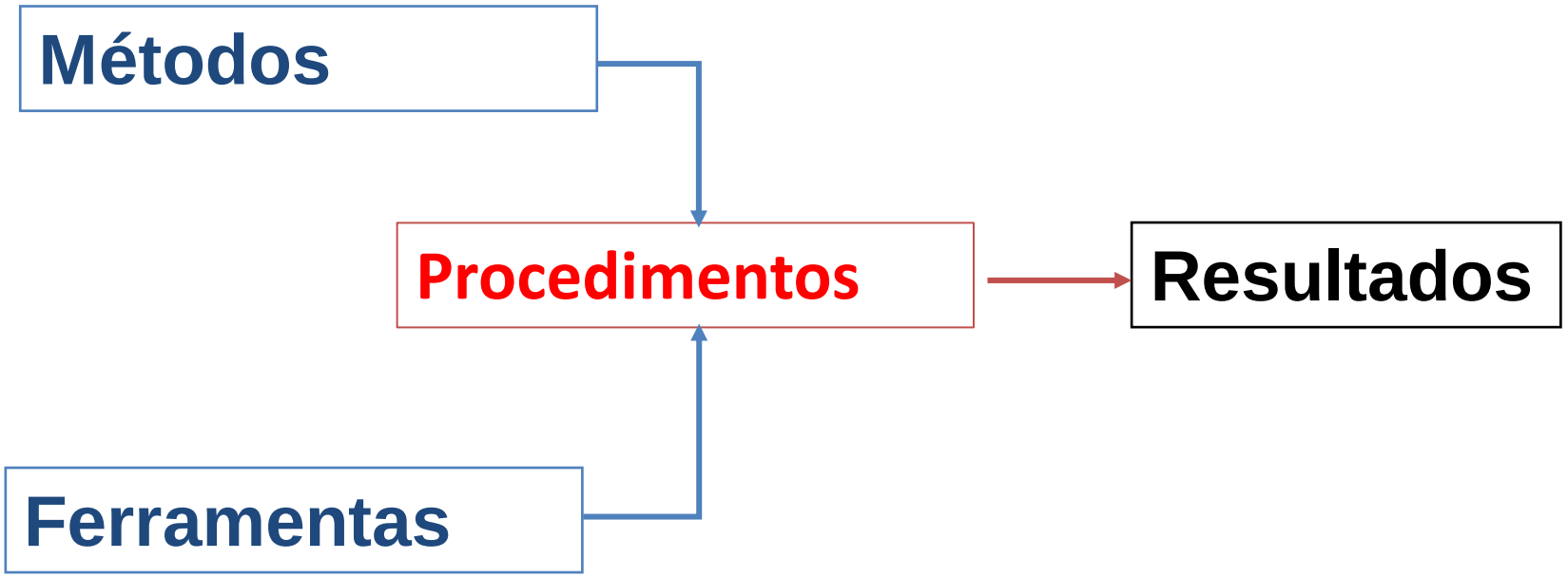
- **Upper-CASE** - Ferramenta para dar apoio às fases iniciais do processo de software.
- **Lower-CASE** - Ferramenta para dar apoio à implementação a aos testes.
  - Ex.(Poseidon para UML , ArgoUML ..)

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Engenharia de Software - Procedimentos

- Constituem o **elo de ligação** que mantém juntos os **métodos** e as **ferramentas** para desenvolvimento do software.
- Sequência em que os métodos serão aplicados.
- Controles que ajudam assegurar a qualidade e coordenar as alterações.
- Marcos de referência que possibilitam administrar o progresso do software.

Engenharia de Software - Procedimentos



Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Princípios da Engenharia de Software

- Todo engenheiro de software deve desenvolver com:
  - Rigor e Formalidade
  - Separação de interesses
  - Modularidade
  - Abstração
  - Antecipação de mudanças
  - Generalidade
  - Possibilidades de evolução

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Princípios da Engenharia de Software

- **Rigor e Formalidade**

  O rigor é a abordagem que produz produtos mais confiáveis pelo controle das variáveis envolvidas. Formalidade é o requisito de que o processo seja dirigido e avaliado por leis matemáticas.

Princípios da Engenharia de Software

- **Separação de interesses**

  Separar conceitos permite-nos trabalhar com aspectos individuais e diferentes de um mesmo problema. Esta separação facilita o entendimento, focando a atenção em certas características mais significativas.

Princípios da Engenharia de Software

- **Modularidade**

Consiste na divisão de sistemas complexos em partes menores e mais simples (módulos) com características desejáveis (coesão e acoplamento).

Princípios da Engenharia de Software

- **Modularidade (cont.)**
  - **Decomposição** é o ato de dividir um problema original em subproblemas recursivamente.
  - **Composição** é o ato de juntar os elementos componentes de um problema até chegar ao sistema completo. Ajuda na manutenção do sistema.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Princípios da Engenharia de Software

- **Antecipação de mudanças**

  Sistemas de softwares são desenvolvidos enquanto seus requisitos ainda não estão totalmente claros. Quando o sistema é finalmente liberado, novos requisitos podem ser descobertos e velhos requisitos atualizados através do "feedback" do usuário.

Princípios da Engenharia de Software

- **Generalidade / Especialidade**

  Soluções genéricas tendem a ser mais caras em termos de recursos e em tempo de desenvolvimento, ao contrário das soluções específicas. No processo de produção de software estas questões devem ser cuidadosamente analisadas.

Princípios da Engenharia de Software

- **Incrementabilidade**

  Caracteriza o processo em modo passo a passo, incrementalmente. O objetivo desejado é atingido por aproximações sucessivas. Útil quando os requisitos iniciais não foram todos obtidos antes do início do desenvolvimento da aplicação.

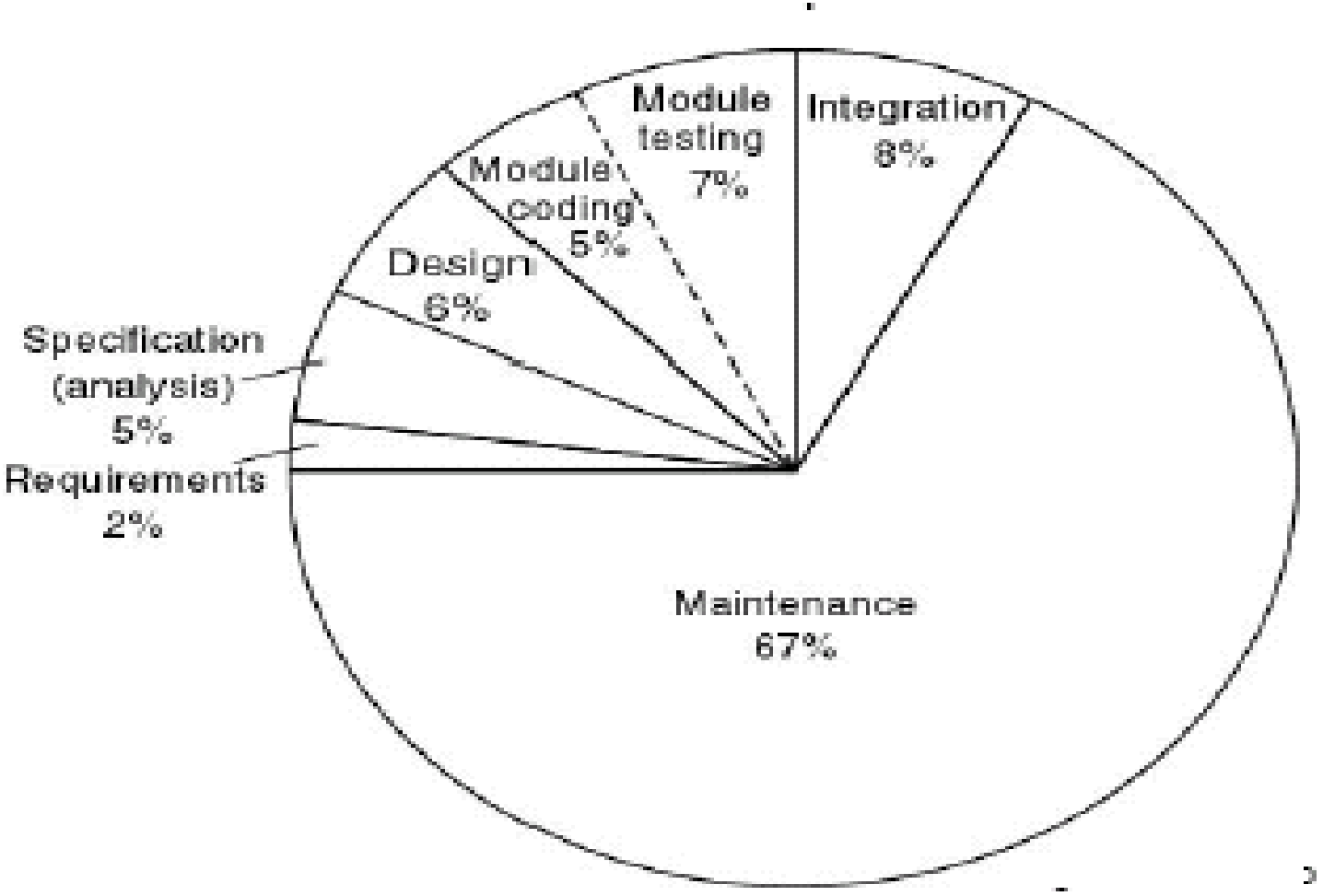
Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Quais os custos da Engenharia de Software?

- A distribuição dos custos através do processo de software depende do processo usado e do tipo de software a desenvolver.
- Custos de desenvolvimento de um software complexo quando se conseguem definir custos separadamente para especificação, desenho, implementação, integração e testes.

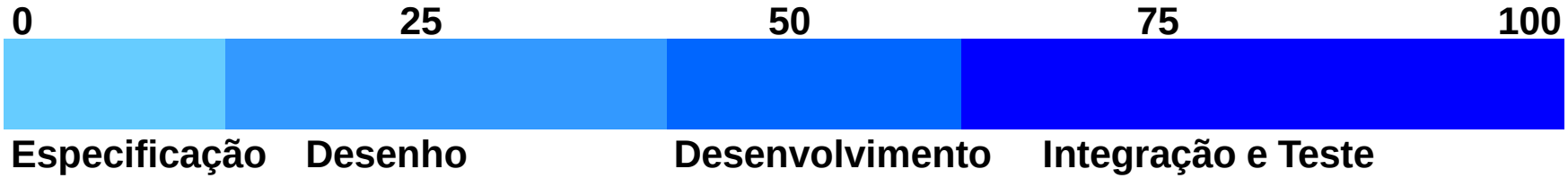
Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Quais os custos da Engenharia de Software?



Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Quais os custos da Engenharia de Software?

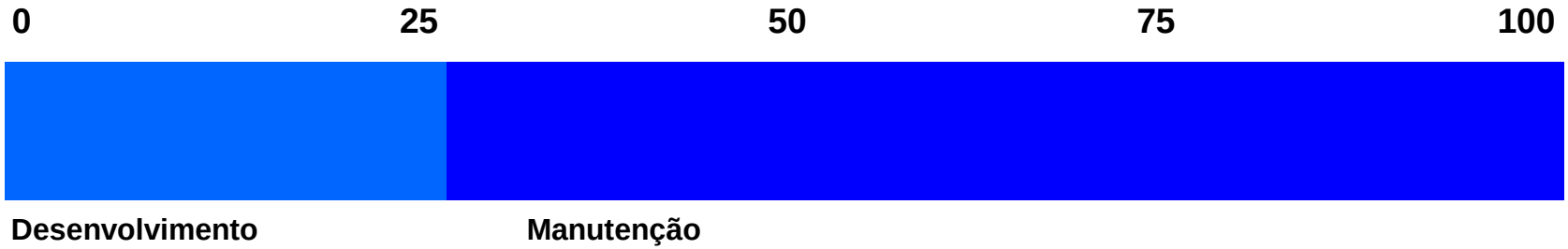


- Custos segundo aproximação evolutiva

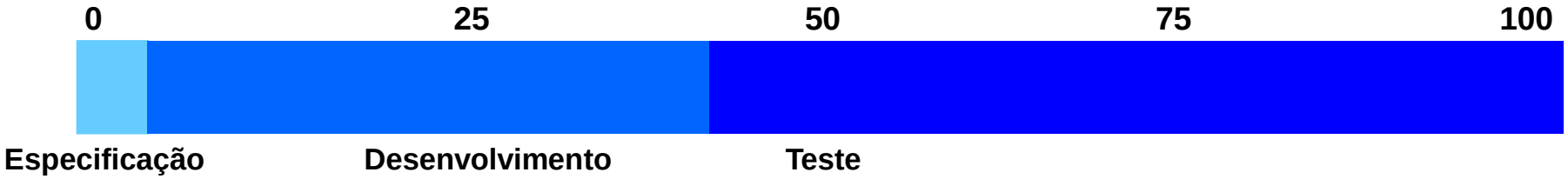
Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Quais os custos da Engenharia de Software?

- Custos relativos entre desenvolvimento e manutenção do software

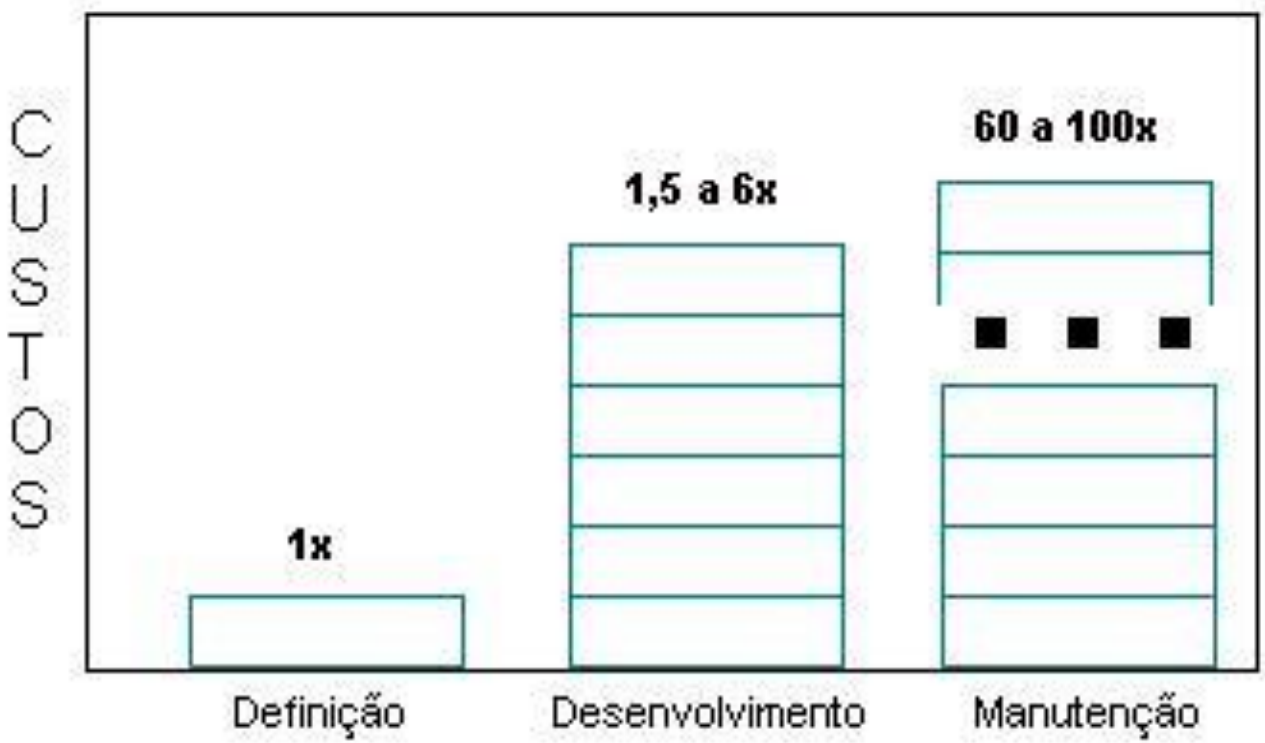


- Custos de desenvolvimento de produtos de software genéricos:



Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Custo em relação a falhas



Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Impacto dos custo de manutenção

- Manutenção corretiva [aproximadamente 20%]
  - 60 a 70% das necessidades de correção são falhas de especificação ou design.
- Manutenção evolutiva (melhoria)
  - Aperfeiçoamento [aproximadamente 60%]
  - Adaptação [aproximadamente 20%]

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Principais desafios enfrentados pela Engenharia de Software

- **Sistemas de legado -** Devem ser mantidos e devem ser atualizado.
- **Heterogeneidade** - Operar com sistemas distribuídos e incluem uma mistura de hardware e software.
- **Fornecimento -** Entrega mais rápida de software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Responsabilidade profissional e ética

- **Confiabilidade –** respeitar a confiabilidade de seus empregadores ou clientes.
- **Competência** – os engenheiros não devem aceitar serviços que estejam for a do seu limite de competência.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Responsabilidade profissional e ética

- **Direito de propriedade intelectual** – os engenheiros devem estar cientes das leis locais que regulam o uso da propriedade intelectual, como patentes e direitos autorais.

Resumindo
Engenharia de Software

- Aplicação de teoria, modelos, formalismos, técnicas e ferramentas da ciência da computação e áreas afins para o **desenvolvimento** sistemático de software.

- Produção da **documentação** formal destinada a comunicação entre os membros da equipe de desenvolvimento bem como aos usuários.

Resumindo
Engenharia de Software

- Encontrar caminhos para se **"construir"** softwares de qualidade.

- Fatores externos, perceptíveis aos usuários e clientes, devem ser distinguidos dos fatores internos, perceptíveis aos projetistas e implementadores.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Resumindo
Engenharia de Software

- A manutenção de software, que consome grande parte dos custos do software, é penalizada pela dificuldade em se implementar mudanças no software produto, e pela excessiva dependência dos programas da estrutura física dos dados que eles manipulam.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Pesquisa

- Visualizando a informática em uma empresa, cite exemplos de:
  - Mitos utilizados pelo analista ou pelo administrador de uma empresa específica.
  - Falhas que ocasionaram grandes custos.
  - Falhas que ocasionaram mudanças de projetos.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição

Pesquisa

- Quais são os quatro atributos que todo software profissional deve possuir? Sugira outros atributos que podem ser significantes.
- Debater com os seus colegas o código de ética na Engenharia de Software.

Fonte:
PRESSMAN, ROGER - Engenharia de Software - 6° Edição
SOMMERVILLE - Engenharia de Software - 8° / 9° Edição