



Introdução à Engenharia de Software

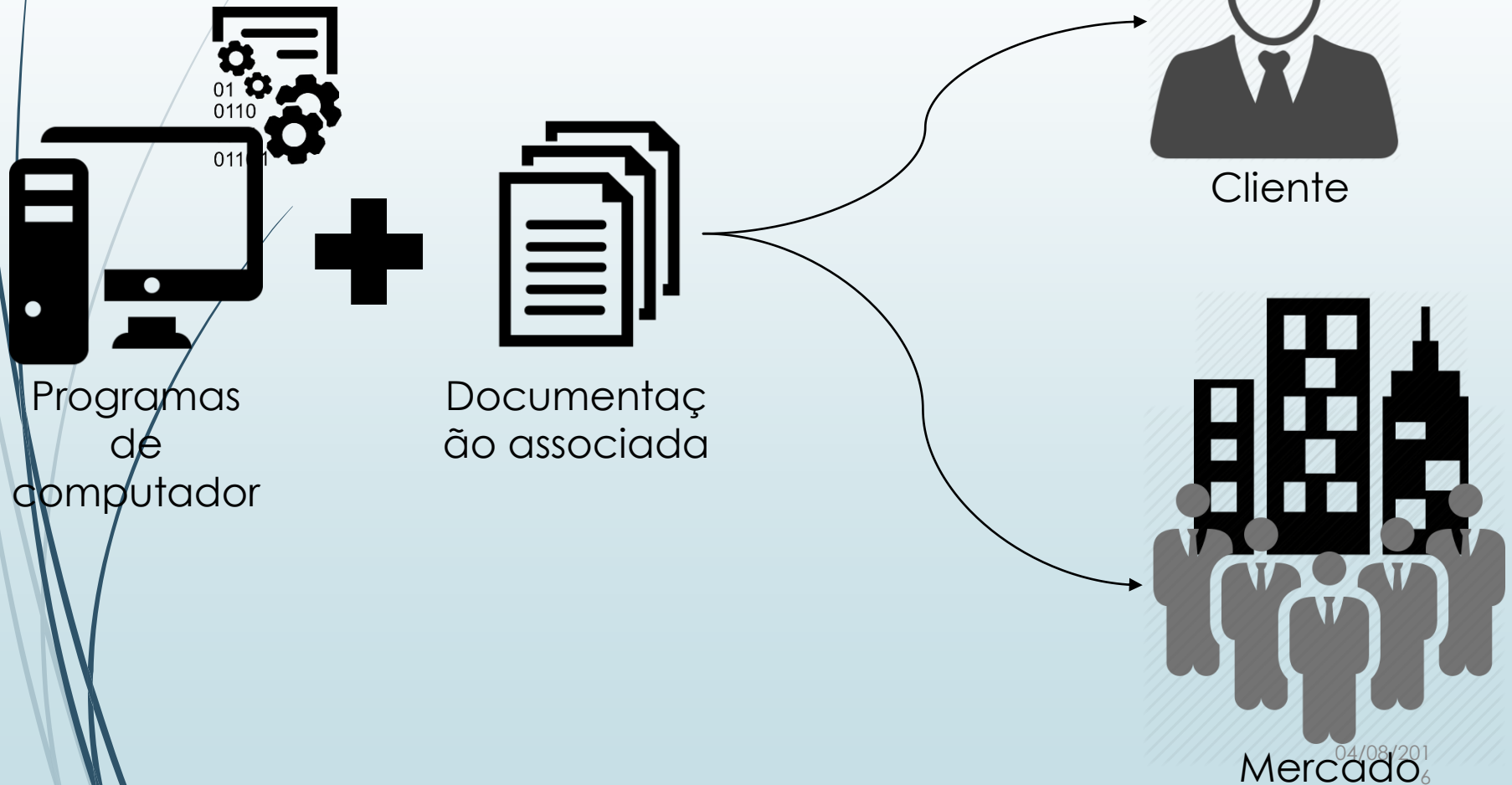
UNIP - Araraquara

Curso: Ciência da Computação

Disciplina: Engenharia de Software

Profº: João Paulo Moreira dos Santos

O que é software?



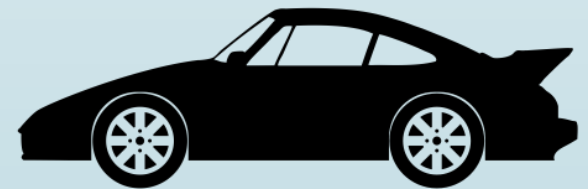
Características do Software

- Software não é um elemento físico; é um elemento lógico;
- O software não pode ser manufaturado, pois os custos estão concentrados no desenvolvimento;
- Existem diferentes abordagens para se chegar no produto final
- Não é construído aproveitando-se componentes prontos, pois muitas vezes é personalizado

Importância do Software



**Software está presente no mundo:
Elemento crítico no cotidiano das pessoas**



Desafios de desenvolver Software

Confiabilidade

► Exemplo: Vôo

1. Dados conflitantes (falha nos sensores)
2. Sistema assume o controle (piloto automático)
3. Piloto tenta reiniciar o sistema (boot)
4. Em 4 minutos o avião mergulha no oceano



Desafios de desenvolver Software

Preço e Desempenho

► Exemplo: Celular

1. Pouco espaço na memória
2. Grande variação em características de aparelhos



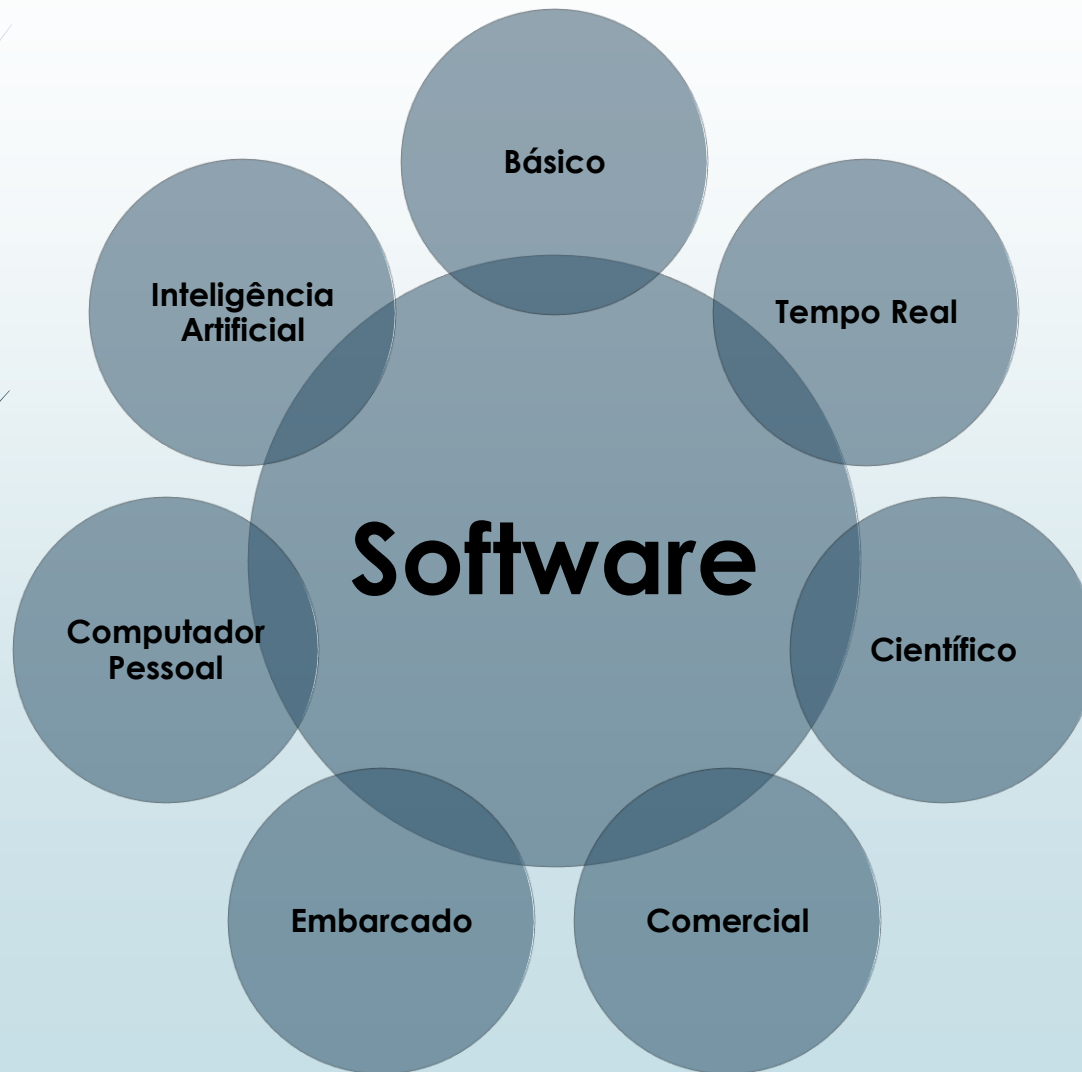
Desafios de desenvolver Software

Sistemas Críticos

- Exemplo: Equipamentos médicos
 1. Extremamente crítico
 2. Lidam com vidas

- Exemplo: Caixas eletrônicos
 1. Prejuízos financeiros

Aplicações



Evolução do Software

Crise de Software

- Cresce o número de sistemas baseado em computador
- Manutenção quase impossível

○ **1950**

- O software era uma arte "secundária" para a qual havia poucos métodos sistemáticos;
- O software era específico para cada aplicação;
- Não havia documentação

○ **1975**

- Multiprogramação e sistemas multiusuários;
- Técnicas interativas;
- Sistemas de tempo real – 1ª geração de SGBD's;

○ **1990**

- Sistemas distribuídos;
- Uso generalizado de microprocessadores;

○ **Atual**

- Tecnologias orientadas a objetos;
- Software de inteligência artificial usados na prática;
- Computação Paralela

Crise do Software

Refere-se a um conjunto de **problemas** encontrados no **desenvolvimento de software** e na etapa de **manutenção**.

Crise do Software - Problemas

1. As estimativas de prazo e de custo frequentemente são imprecisas

“Falta de dedicação na etapa de coletas de dados no processo de desenvolvimento de software”

“Vaga indicação de produtividade não se pode avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões”

Crise do Software - Problemas

2. Insatisfação do cliente com o sistema concluído:

“Projetos de desenvolvimento de software são efetuados com um vago indício das exigências do cliente”

Crise do Software - Problemas

3. A qualidade de software às vezes é menos que adequada:

“Só recentemente começam a surgir conceitos quantitativos sólidos de garantia de qualidade de software”

Crise do Software - Problemas

4. O software existente é muito difícil de manter (sem Manutibilidade):

“A tarefa de manutenção devora o orçamento destinado ao software”

“A facilidade de manutenção não foi enfatizada como um critério importante”

Crise do Software - Causas

- Características próprias do software:

“O software é um elemento de sistema lógico”

“O software não se desgasta, mas se deteriora com o tempo”

Crise do Software - Causas

- Falhas das pessoas responsáveis pelo desenvolvimento do software:

“Gerentes sem nenhum background em software”

“Profissionais da área de software têm pouco treinamento formal em novas técnicas para o desenvolvimento de software”

“Resistência às mudanças”

Crise do Software - Causas

- **Mitos do Software:**
 - Falta de Planejamento e Comunicação



Crise do Software - Causas

► Mitos do Software:

Administrativo

MITO

Já temos um **manual** repleto de **padrões** e **procedimentos** para desenvolver o software

Equipe tem **ferramentas** de desenvolvimento de software de **última geração**.

Se **atrasar os prazos**, é possível adicionar mais programadores e recuperar o atraso.

REALIDADE

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

É preciso muito mais do que **computadores recentes** e **melhores ferramentas** para desenvolver um software de alta qualidade.

O desenvolvimento de software **não** é um **processo mecânico**. Acrescentar pessoas em um projeto torna-o ainda mais atrasado.

Crise do Software - Causas

► Mitos do Software:

Cliente

MITO

Uma **declaração geral** dos objetivos é **suficiente** para começar a escrever programas

Mudanças dos requisitos podem ser facilmente acomodadas, porque o software é flexível.

REALIDADE

Uma **definição inicial ruim** é a **principal causa de fracassos** no desenvolvimento de software

Uma **mudança**, quando solicitada tardiamente num projeto, pode ser mais **dispendiosa**

Crise do Software - Causas

➤ Mitos do Software

MITO

*Implementa o programa e coloca em **funcionamento**, então o trabalho estará **completo***

*Enquanto não tiver o programa "funcionando", não é possível avaliar sua **qualidade***

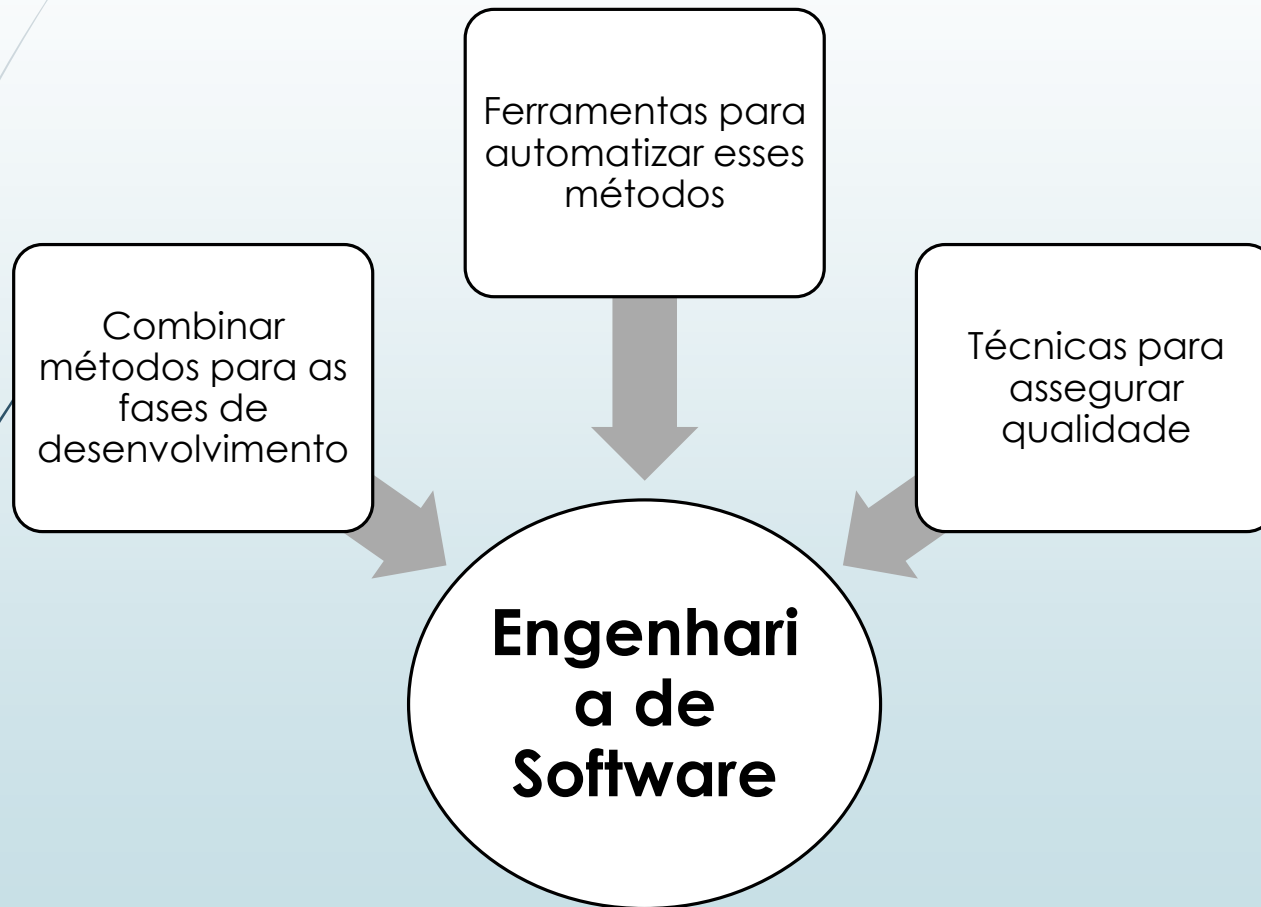
REALIDADE

Os dados da indústria indicam que entre **50 e 70% do esforço gasto** num programa serão **despendidos** depois que for **entregue** pela primeira vez ao **cliente**.

Um **programa funcionando** é somente uma parte todos os itens produzidos durante a **construção** e **manutenção** do software

Profissional

Solução da Crise do Software



Engenharia de Software (ES)

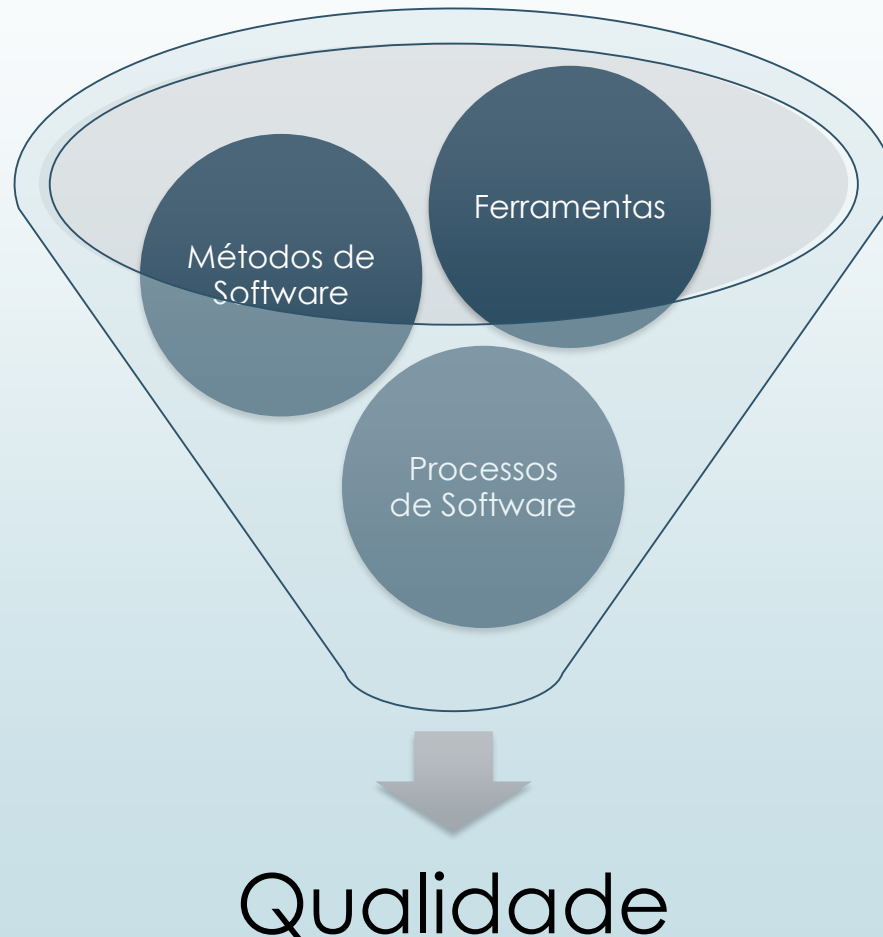
“Disciplina que utiliza um conjunto de métodos, técnicas e ferramentas para analisar, projetar e gerenciar desenvolvimento e manutenção de software”

Engenharia de Software (ES)

► Objetivos:

- Controle sobre o desenvolvimento de software dentro de **custos**, **prazos** e níveis de **qualidade** desejados;
- Produtividade no desenvolvimento, operação e manutenção de software;
- Qualidade versus Produtividade;
- Permitir que profissionais tenham controle sobre o desenvolvimento de software dentro de custos, prazos e níveis de qualidade desejados;

Engenharia de Software (ES)



ES – Qualidade de Software

- O software que satisfaz os **requisitos** solicitados pelo usuário. Deve ser fácil de **manter**, ter **boa performance**, ser **confiável** e **fácil de usar**
- Atributos de qualidade:
 - **Manutenibilidade**: O software deve evoluir para atender os requisitos que mudam;
 - **Eficiência**: O software não deve desperdiçar os recursos do sistema;
 - **Usabilidade**: O software deve ser fácil de usar pelos usuários para os quais ele foi projetado.

ES – Processo de Software

- Conjunto de **atividades** (e resultados) cujo objetivo é o **desenvolvimento de software**;
- O processo oferece **estabilidade, controle e organização** no ciclo de desenvolvimento;
- Atividades principais:
 - Especificação
 - Desenvolvimento
 - Validação
 - Evolução

ES – Métodos de Software

- Abordagens estruturadas para desenvolvimento de software;
- Os métodos fornecem a técnica de como fazer;
- Métodos incluem:
 - Modelos
 - Notações
 - Regras, etc.

ES – Ferramentas

- Fornecem apoio automatizado (ou semi-automatizado) para o processo e para os métodos
- **Ex:** Ferramentas de modelagem
 - Permitem definir ações, tarefas, produtos, etc. de um modelo de processo