



UNIVERSIDADE FEDERAL DO PIAUI – UFPI  
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - CSHNB  
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO



# Engenharia de Software I

Introdução a Engenharia de Software

**Professora Pâmela Carvalho**

**29/11/22**

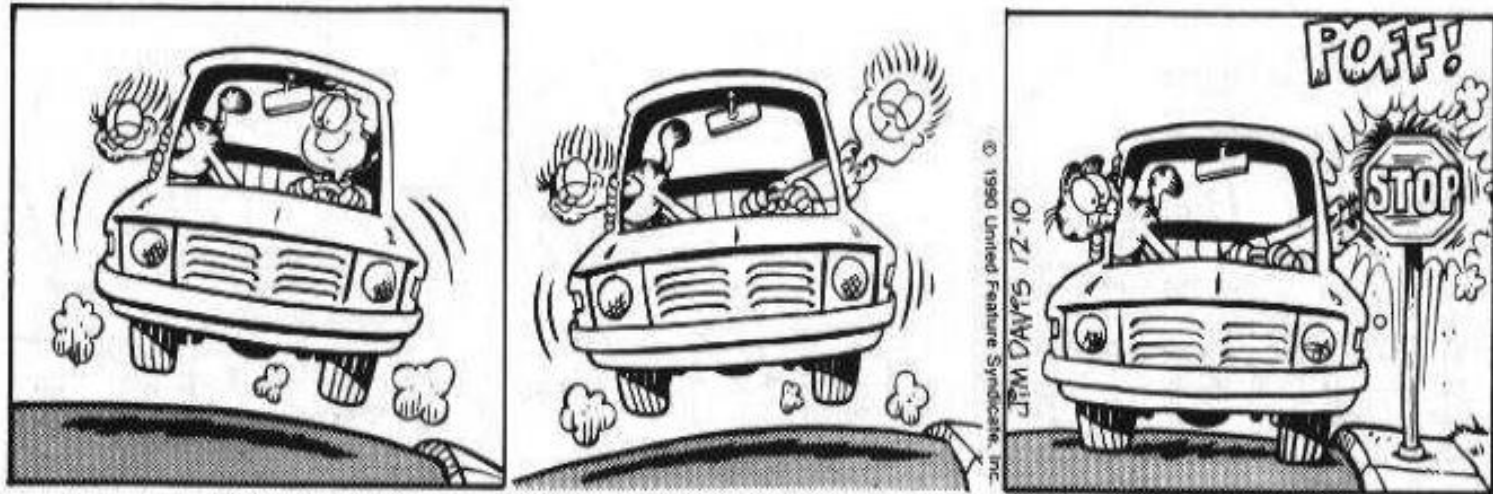
# Nada pode parar a automação



Nem toda apresentação será um sucesso



O que serve para um cliente pode  
não servir para o próximo



# Busque soluções eficientes

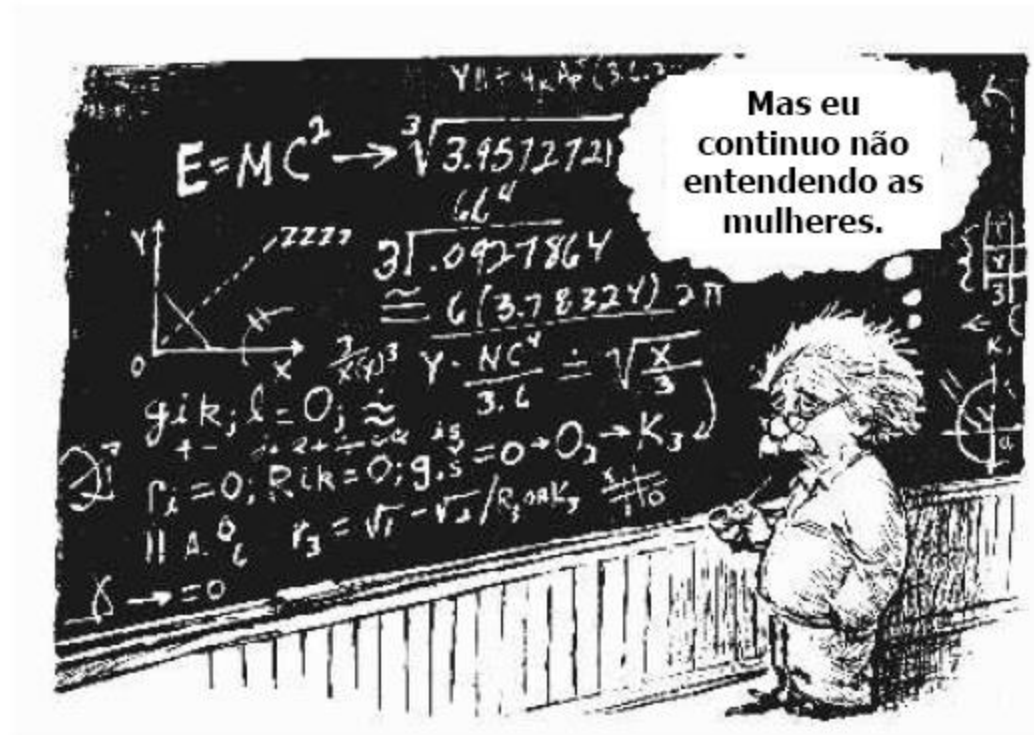




# Suporte! Ferramentas diferentes para situações diferentes



# Matemática não é tudo



Software está profundamente incorporado em praticamente todos os aspectos de nossas vidas e, conseqüentemente, o número de pessoas interessadas nos recursos e nas funções oferecidas por determinada aplicação<sup>1</sup> tem crescido significativamente. *Depreende-se, portanto, que é preciso fazer um esforço conjunto para compreender o problema antes de desenvolver uma solução de software.*

**Pressman, 2016**



# Crise do Software

O que vocês imaginam ser isso?



# Crise de Software

## Evolução do Software (1950 – 1965)

- O *hardware* sofreu contínuas mudanças;
- O *software* era uma arte “secundária” para a qual havia poucos métodos sistemáticos;
- O *hardware* era de propósito geral;
- O *software* era específico para cada aplicação;
- Não havia documentação.

# Crise de Software

- A crise do software, termo usado nos anos 70, referia-se as **dificuldades do desenvolvimento de software** da época.
- Com a inexistência da Engenharia de Software, **não haviam técnicas estabelecidas para o desenvolvimento de sistemas** que funcionassem adequadamente ou que pudessem ser validadas.

# Crise de Software



- 25% dos projetos são cancelados;
- O tempo de desenvolvimento é bem maior do que o estimado;
- 75% dos sistemas não funcionam como planejado;
- A manutenção e reutilização são difíceis e custosas;
- Os problemas são proporcionais a complexidade dos sistemas.

# Crise de Software - Problemas

## 1 - As estimativas de prazo e de custo frequentemente são imprecisas:

- “ Não dedicamos tempo para coletar dados sobre o processo de desenvolvimento de software”;
- “Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões”.





# Crise de Software - Problemas

## **2 - Insatisfação do cliente com o sistema concluído:**

- “Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente”.

## **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.

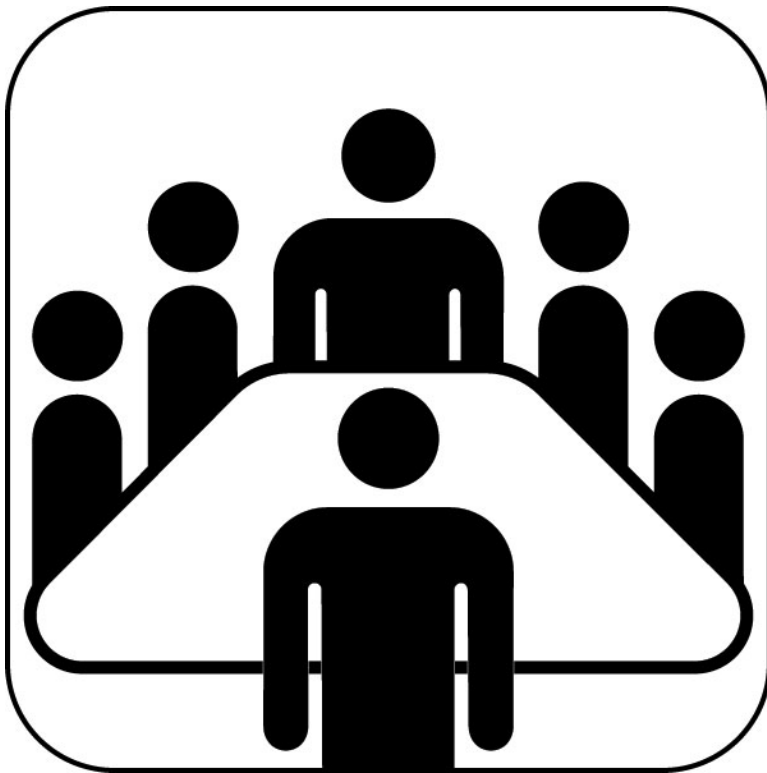
# Crise de Software - Problemas

## **4 - 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.

# Causas dos problemas associados à crise de software

## Falhas das pessoas Responsáveis pelo Desenvolvimento de Software:



- Gerentes sem nenhuma experiência em software;
- Profissionais da área de software têm pouco treinamento formal em novas técnicas para o desenvolvimento de software;
- Resistência a mudanças.

# **Causas dos problemas associados à crise de software**

- **Mitos do software:**
  - **Crenças infundadas sobre o software e sobre o processo utilizado para criá-lo** – remontam aos primórdios da computação.
  - Atualmente, a maioria dos profissionais da engenharia de software reconhece os mitos por aquilo que eles representam – atitudes enganosas que provocaram sérios problemas tanto para gerentes quanto para praticantes da área. Entretanto, antigos hábitos e atitudes são difíceis de ser modificados, e resquícios de mitos de software permanecem.

# Causas dos problemas associados à crise de software

- **Mitos do software:**
  - Vamos dividir em 3 categorias:
    - Gerenciamento;
    - Cliente;
    - Profissional.





# Mitos do Software (Gerenciamento)

- **Mito 1:**

- Já temos um manual repleto de padrões e procedimentos para a construção de software;
- Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?

- **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?



# Mitos do Software (Gerenciamento)

- **Mito 2:**

- Meu pessoal tem ferramentas de desenvolvimento de software de última geração, afinal lhes compramos os mais novos computadores.

- **Realidade:**

- É preciso muito mais do que os mais recentes computadores para se fazer um desenvolvimento de software de alta qualidade.



# Mitos do Software (Gerenciamento)

- **Mito 3:**

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

- **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, mas somente de uma forma planejada.



# Mitos do Software (Cliente)



- **Mito 1:**

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

- **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.

# Mitos do Software (Cliente)



- **Mito 2:**

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

- **Realidade:**

- Uma mudança, quando solicitada tardiamente num projeto, pode ser maior do que a ordem de magnitude mais dispendiosa da mesma mudança solicitada nas fases iniciais.



# Mitos do Software (**Profissional**)

- **Mito 1:**
  - Assim que escrevermos o programa e o colocarmos em funcionamento nosso trabalho estará completo.
- **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.



# Mitos do Software (Profissional)

- **Mito 2:**

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



- **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.

# Resposta à Crise de Software



**ENGENHARIA DE SOFTWARE**

# Em resumo...

- O termo “**crise do software**” foi criado no final dos anos 60 para tratar de uma séria questão que começava a despontar: a **difículdade enfrentada para desenvolvimento de softwares**.
- Essa situação foi desencadeada pela complexidade cada vez maior dos problemas, ausência de técnicas bem estabelecidas e crescente demanda por novas aplicações.
- Para contorná-la, surge a **Engenharia de Software**, que traz uma **abordagem disciplinada e sistemática sobre o desenvolvimento, operação e manutenção**, incluindo métodos para projeção, implementação, teste e documentação.

# Importância da Engenharia de Software



Como o cliente explicou



Como o líder de projeto entendeu



Como o analista planejou



Como o programador codificou



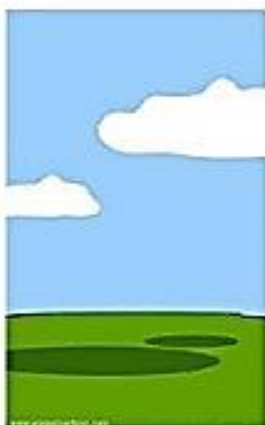
O que os beta testers receberam



Como o consultor de negócios descreveu



Valor que o cliente pagou



Como o projeto foi documentado



O que a assistência técnica instalou



Como foi suportado



Quando foi entregue



O que o cliente realmente necessitava



# Introdução a Engenharia de Software

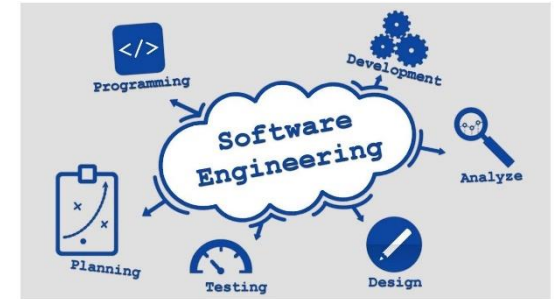
Vamos responder umas perguntas básicas sobre o assunto:

- O que é Engenharia de Software?
- Quem realiza?
- Porque é importante?
- Quais são as etapas envolvidas?



# Introdução a Engenharia de Software

## O QUE É?

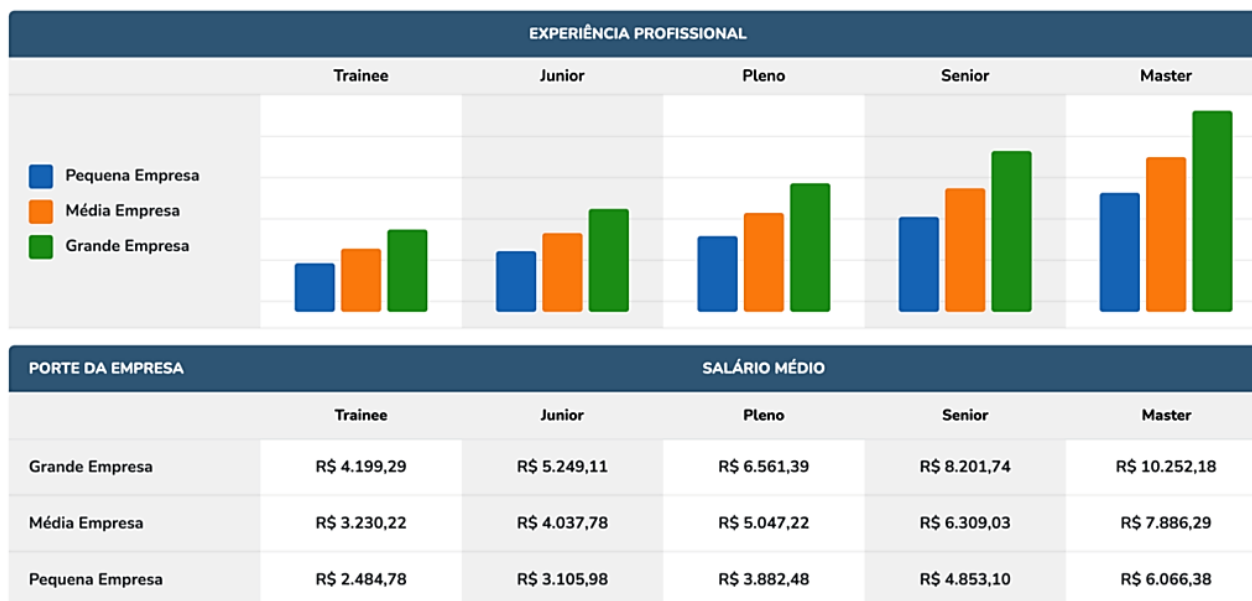


- A engenharia de software abrange um processo, um conjunto de métodos (práticas) e um leque de ferramentas que possibilitam aos profissionais desenvolverem software de altíssima qualidade.
- O IEEE elaborou a seguinte definição para ES:
  - **Engenharia de software:** É a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software; isto é, a aplicação de engenharia ao software.

# Introdução a Engenharia de Software

## QUEM REALIZA?

- Os engenheiros de software aplicam o processo de engenharia de software.



# Introdução a Engenharia de Software

## PORQUE É IMPORTANTE?

- Porque nos capacita para o desenvolvimento de sistemas complexos dentro do prazo e com alta qualidade.
- Ela impõe disciplina a um trabalho que pode se tornar caótico, mas também permite que as pessoas produzam *software* de computador adaptado à sua abordagem, da maneira mais conveniente às suas necessidades.



# Introdução a Engenharia de Software

## QUAIS SÃO AS ETAPAS ENVOLVIDAS?

- Cria-se software para computadores da mesma forma que qualquer produto bem-sucedido: aplicando-se um **processo adaptável e ágil** que conduza a um resultado de alta qualidade, atendendo às necessidades daqueles que usarão o produto.
- O processo de software incorpora **cinco atividades estruturais**: (1) Comunicação, (2) Planejamento, (3) Modelagem, (4) Construção e (5) Entrega.

# Processo de Engenharia da Software

- A base da engenharia de software é a camada de processos.
- O processo de engenharia de software é a liga que mantém as camadas de tecnologia coesas e possibilita o desenvolvimento de software de forma racional e dentro do prazo.

## Camadas da Engenharia de Software:



Leitura base: Pressman, 2016 (8ª edição)  
Páginas 1 a 28

**Até a próxima aula...**