

# Modelagem de Software



---

Prof. Dr. Ronaldo Castro de Oliveira

[ronaldo.co@ufu.br](mailto:ronaldo.co@ufu.br)

[www.facom.ufu.br/~ronaldooliveira](http://www.facom.ufu.br/~ronaldooliveira)

FACOM - 2020



---

*Introdução a Engenharia  
de Software*



# Introdução

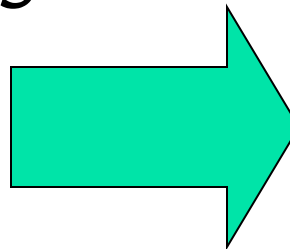
---

- “O Software ultrapassou o Hardware como chave para o sucesso de muitos sistemas baseados em computador” (Pressman, pg. 3, 1992)

# O Software é o que faz a diferença!!!

---

- *Completeza* da informação
- *user-friendliness*
- *web-enhanced*
- inteligência
- funcionalidade
- compatibilidade
- suporte



Tornam cada produto melhor que outro



# A importância do Software

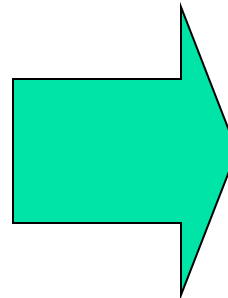
---

- Durante as 3 primeiras décadas da era do computador, o principal desafio era desenvolver um **HARDWARE** de baixo custo e alto desempenho.
- O hoje o desafio é melhorar a qualidade (e reduzir os custos) das soluções baseadas em **SOFTWARE!**

# A evolução do Software

---

Computação



- Nova  
Revolução  
Industrial

- 3a. Onda

# Revolução Industrial

## Primeira Onda

---

- Ferro (Darby, 1709)
- Máquina a vapor:
  - Inventada (Newcomen, 1712)
  - Aperfeiçoada (WATT, 1766 - '69 -'82)
- Mecanização da indústria têxtil:
  - Tear Mecânico (Kay, 1722)
  - Máquina de fiar (Hargreaves, 1764)
- Aspectos sociais, políticos e econômicos  
Têxteis, Carvão e Ferro

# Revolução Industrial

## Segunda Onda

---

- Aço (Bessemer, 1856 e 1885 - Liga)
- Locomotiva a Vapor (Rede de Transporte - 1830)
- Máquina de Costura (SINGER, 1851)
- Motor a combustão interna:
  - Primeiro eficiente (OTTO, 1876)
  - Produção automobilística em massa (Daimler e Benz, 1896)
- Desemprego e fim da escravidão



# Revolução Industrial Terceira Onda



- Energia Nuclear (Fermi, 1942)
- Uso Industrial/Comercial da Eletricidade
- Computadores Eletrônicos (ENIAC 1946)
- Transistor (Shockley, 1948)

Sociedade  
Industrial

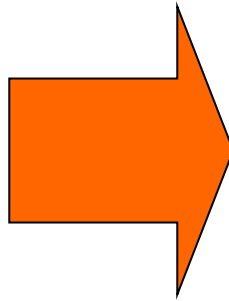
Sociedade  
da Informação

# Revolução Industrial

## Uma nova Onda

---

**SOCIEDADE  
INDUSTRIAL**



**SOCIEDADE  
DA INFORMAÇÃO**

transformação



# Filosofando...

---

- A mudança de uma sociedade industrial para uma baseada na informação é uma *Radical Mudança Econômica*:
  - Material tem menos valor
  - Informação tem mais valor

# A Evolução do Software

---

- Primeiros Anos 1950 – 1965
  - Orientação Batch
  - Distribuição Limitada
  - Software Customizados
  - Poucos Profissionais
- Segunda Era 1965 – 1975
  - Multiusuário
  - Tempo Real
  - Bancos de Dados
  - Produtos de Software



# A Evolução do Software

---

- Terceira Era 1975 - 1988
  - Sistemas Distribuídos
  - “Inteligência” Embutida
  - Hardware de Baixo Custo
  - Impacto de Consumo
- Quarta Era 1988 – 2000
  - Sistemas Desk-top poderosos
  - Tecnologia Orientada a Objetos
  - Sistemas Especialistas
  - Redes Neurais Artificiais
  - Computação Paralela

# A Evolução do Software

---

- Quinta era 2000 .....
- Realidade virtual
- Computação móvel
- Explosão da Internet
- Sistemas computacionais poderosos
- Software livre
- .....



# O que é Software?

---

- Definição - Software é:

1<sup>o</sup> - instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados;

2<sup>o</sup> - estruturas de dados que permitem a manipulação das informações;

3<sup>o</sup> - documentos que descrevem a operação e uso dos programas.



# Características do Software - 1

---

- O Software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico:
  - Custos são concentrados no trabalho de engenharia.
  - Projetos não podem ser geridos como projetos de manufatura.
  - “Fábrica de Software!”





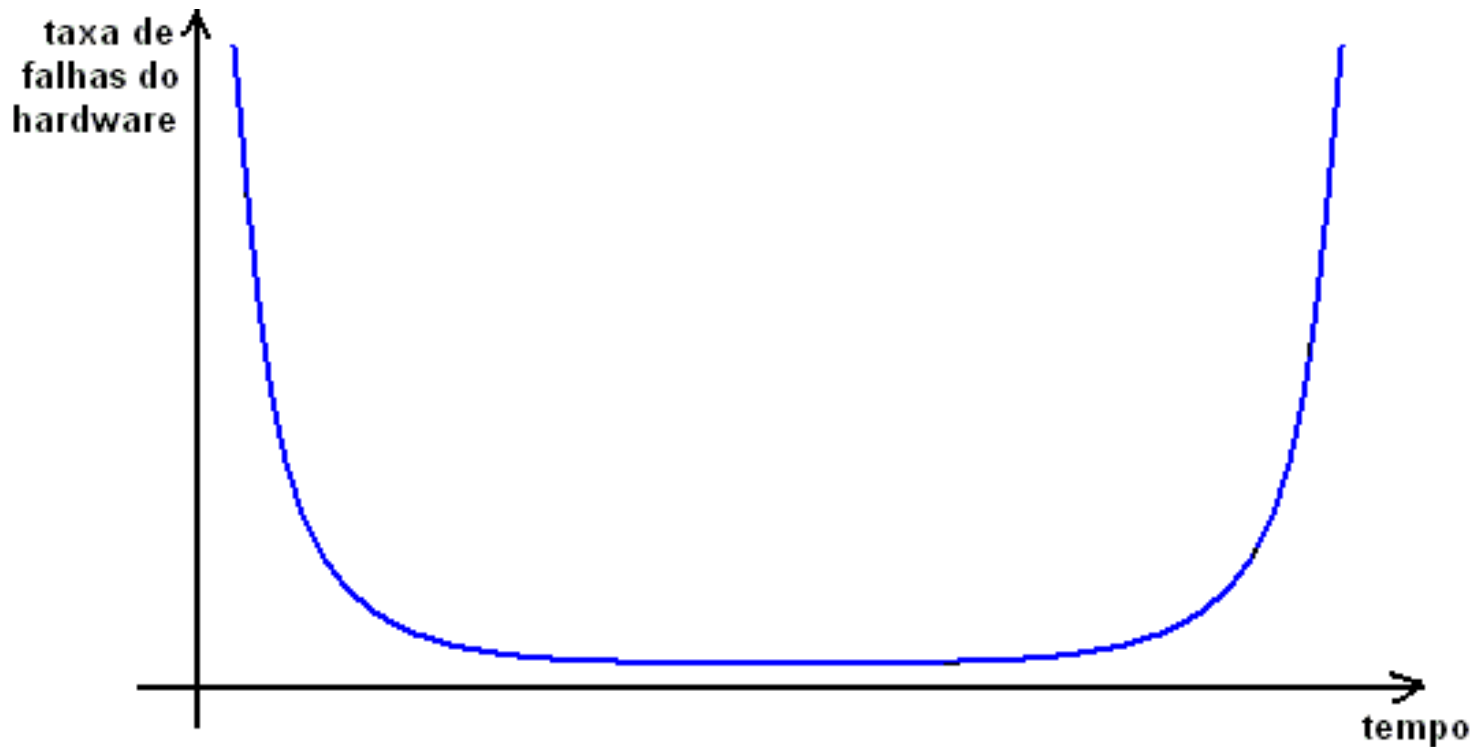
# Características do Software - 2

---

- Software não desgasta!
  - Software não é sensível aos problemas ambientais que fazem com que o hardware se desgaste.
  - Toda falha indica erro de projeto ou implementação: manutenção do Software é mais complicada que a do Hardware.

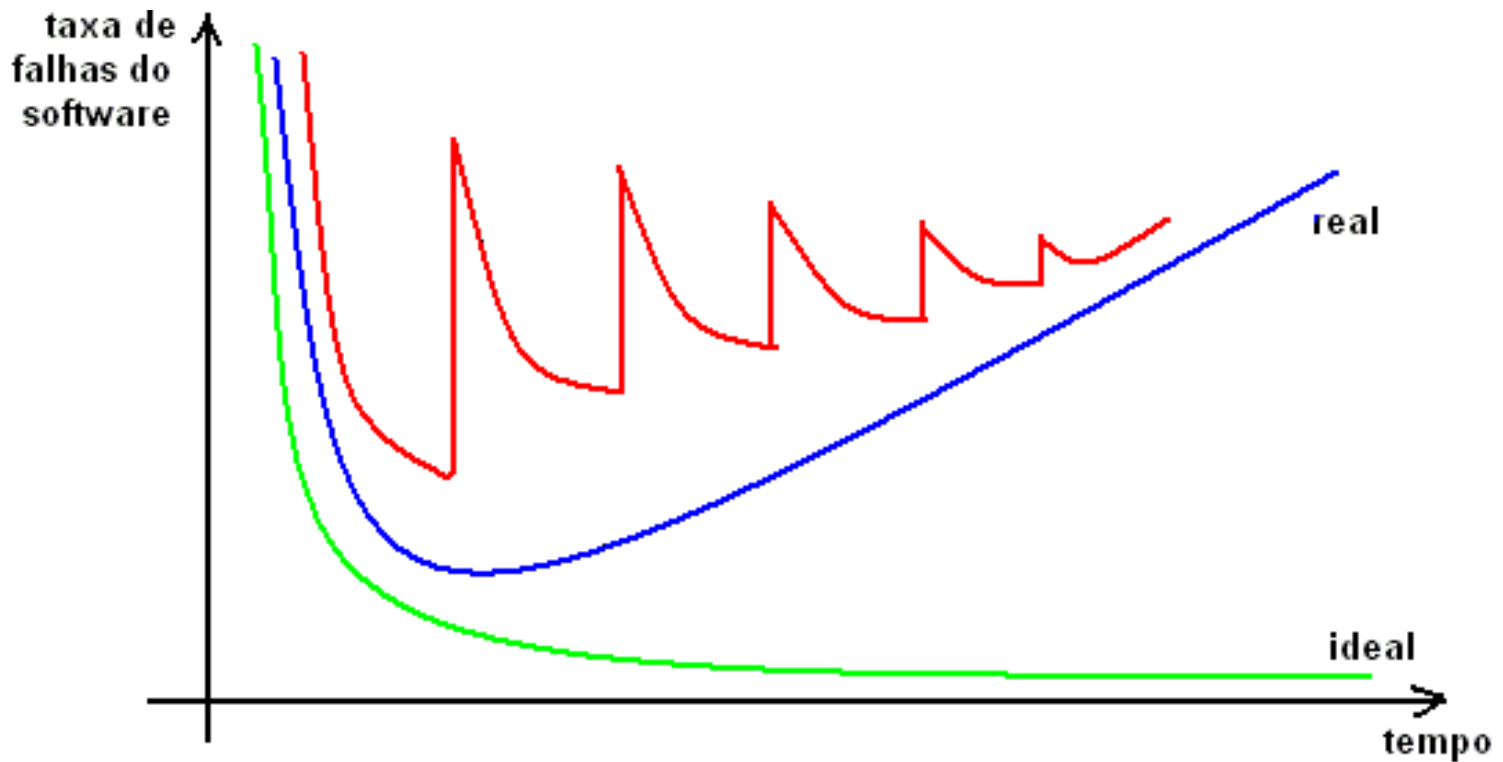
# Características do Software - 3

## Taxa de Falhas de Hardware



# Características do Software - 4

## Taxa de Falhas do Software





# Observações

---

- A maioria dos softwares é feita sob medida e não montada a partir de componentes existentes.
- O hardware não é tão mais importante.
- Situação esta mudando:
  - Orientação a objetos.
  - Reusabilidade é a tendência do mercado (diminui custos e melhora projetos).



# Engenharia: Definição

---

- “ Aplicação de conhecimentos científicos e empíricos, e certas habilitações específicas, à criação de estruturas, dispositivos e processos para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas”

Aurélio

# Engenharia de Software:

## Definição

---

- “Engenharia de Software é o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável, que atenda os requisitos dos clientes e que funcione eficientemente em máquinas reais”



# Aplicações de Software

---

- Software Básico
- Software de Tempo Real
- Software Comercial
- Software Científico ou de Engenharia
- Software Embutido
- Software de Computador Pessoal
- Software de Inteligência Artificial
- Software para Internet



# Crise de Software

---





# Uma Crise no horizonte

---

- A indústria de Software tem passado por uma “crise” que a acompanha há quase 30 anos:
  - Aflição Crônica != Crise
- Problemas não se limitam ao software que não funciona adequadamente, mas abrange:
  - desenvolvimento, testes, manutenção, suporte, treinamento, implantação, etc.



# Therac-25

---

- Equipamento de Radioterapia.
- Entre 1985 e 1987 se envolveu em 6 acidentes, causando mortes por overdoses de radiação.
- Software foi adaptado de uma antecessora, Therac-6:
  - falhas por falta de testes integrados
  - falta de documentação

# Denver International Airport



- Custo do projeto: US\$ 4.9 bilhões
  - 100 mil passageiros por dia
  - 1,200 vôos
  - 53 milhas quadradas
  - 94 portões de embarque e desembarque
  - 6 pistas de pouso / decolagem

# Denver International Airport



- Erros no sistema automático de transporte de bagagens (*misloaded, misrouted, jammed*):
  - Atraso na abertura do aeroporto com custo total estimado em US\$360 Milhões
- 86 milhões para consertar o sistema

# Ariane 5

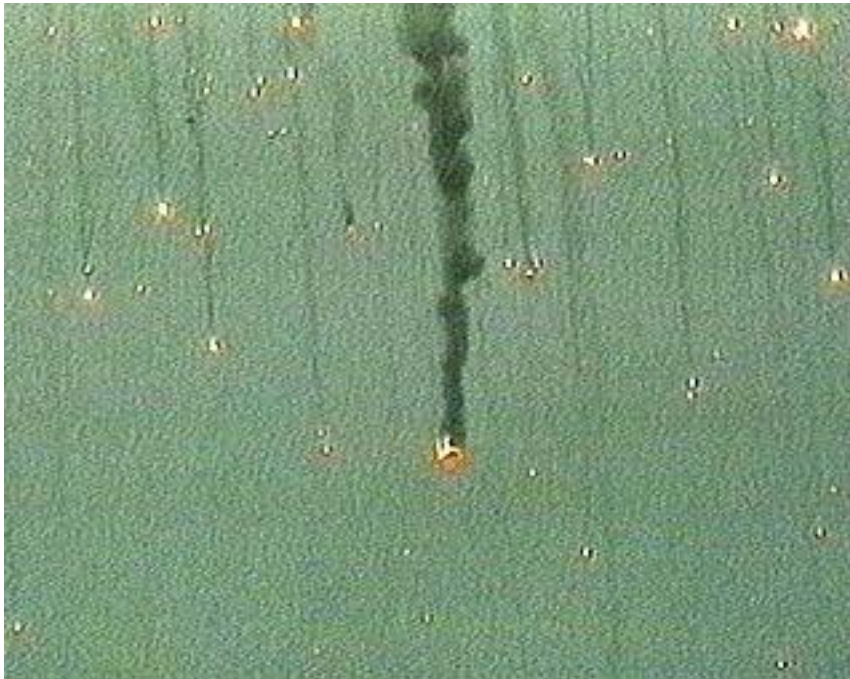


- Projeto da Agência Espacial Européia que custou:
  - 10 anos.
  - US\$ 8 Bilhões.
- Capacidade 6 toneladas.
- Garante supremacia européia no espaço.

# Vôo inaugural em 4/junho/1996



# Resultado



- Explosão 40 segundos após a decolagem.
- Destruição do foguete e carga avaliada em US\$ 500 milhões.



# O que aconteceu? (I)

---

- Fato: o veículo detonou suas cargas explosivas de autodestruição e explodiu no ar. Por que?
- Porque ele estava se quebrando devido às forças aerodinâmicas. Mas por que?
- O foguete tinha perdido o controle de direção (atitude). Causa disso?
- Os computadores principal e back-up deram shut-down ao mesmo tempo.





## O que aconteceu? (II)

---

- Por que o Shut-down? Ocorreria um *run time error* (out of range, overflow , ou outro) e ambos computadores se desligaram. De onde veio este erro?
- Um programa que convertia um valor em ponto flutuante para um inteiro de 16 bits recebeu como entrada um valor que estava fora da faixa permitida.

# Ironia...

- O resultado desta conversão não era mais necessário após a decolagem...





# Quais são os problemas?

---

- A sofisticação do software ultrapassou nossa capacidade de construção.
- Nossa capacidade de construir programas não acompanha a demanda por novos programas.
- Nossa capacidade de manter programas é ameaçada por projetos ruins.

# Perguntas que Engenharia de Software quer responder:

---

- Porque demora tanto para concluir um projeto (não cumprimos prazos)?
- Porque custa tanto (uma ordem de magnitude a mais)?
- Porque não descobrimos os erros antes de entregar o software ao cliente?
- Porque temos dificuldade de medir o progresso enquanto o software está sendo desenvolvido?



# Causas óbvias

---

- Não dedicamos tempo para coletar dados sobre o desenvolvimento do software - resulta em estimativas “a olho”.
- Comunicação entre o cliente e o desenvolvedor é muito fraca.
- Falta de testes sistemáticos e completos.



# Causas menos óbvias

---

- O Software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico (característica 1).
- Gerentes sem *background* em desenvolvimento de SW.
- Profissionais recebem pouco treinamento formal.
- Falta investimento (em ES).
- Falta métodos e automação.



# Mitos da Engenharia de Software

---

# Mitos do Software - Administrativos



---

- Um manual oferece tudo que se precisa saber.
- Computadores de última geração solucionam problemas de desenvolvimento.
- Se estamos atrasados, basta contratar novos programadores e tirar o atraso.



# Mitos do Software – do Cliente

---

- Uma declaração geral é suficiente para começar a escrever programas.
- Mudanças podem ser facilmente acomodadas em um projeto.

# Mitos do Software - do Profissional

---

- Um programa está terminado ao funcionar.
- Quanto mais cedo escrever o código, mais rápido terminarei o programa.
- Só posso avaliar a qualidade de um programa em funcionamento.
- A única coisa a ser entregue em um projeto é o programa funcionando.