



## Unidade 1 – Conceitos e Evolução da Engenharia de Software

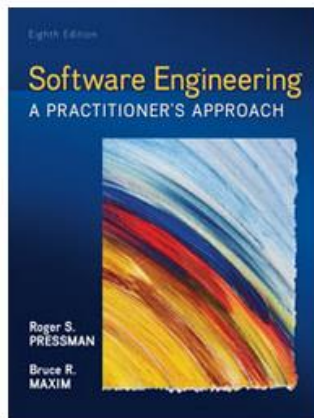


Prof. Aparecido V. de Freitas  
Doutor em Engenharia  
da Computação pela EPUSP  
[aparecidovfreitas@gmail.com](mailto:aparecidovfreitas@gmail.com)

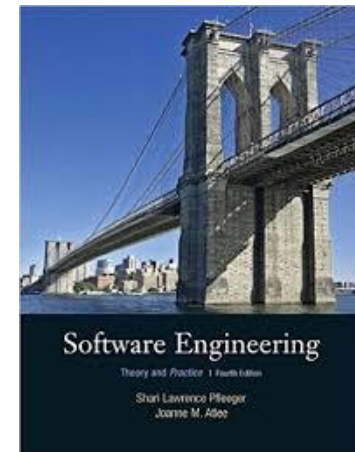
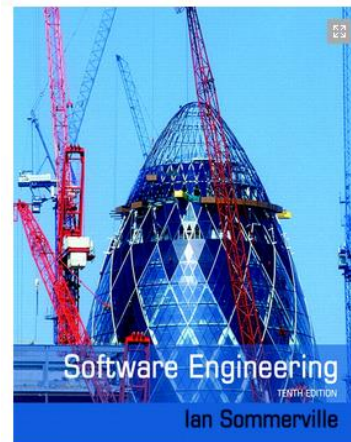


# Bibliografia

- **Software Engineering – A Practitioner's Approach – Roger S. Pressman – Eight Edition – 2014**
- **Software Engineering – Ian Sommerville – 10<sup>th</sup> edition – 2015**
- **Software Engineering – Pfleeger & Atlee – Theory and Practice – 4<sup>th</sup> edition – Prentice Hall 2009**
- Engenharia de Software – Uma abordagem profissional – Roger Pressman - McGraw Hill, Sétima Edição - 2011
- Engenharia de Software – Ian Sommerville – Nona Edição – Addison Wesley, 2007
- Engenharia de Software – Teoria e Prática - Shari Lawrence Pfleeger – Editora Pearson – 3<sup>a</sup> edição



[Software Engineering: A Practitioner's Approach, 8/e](#)





Em primeiro lugar...

O que é software ?

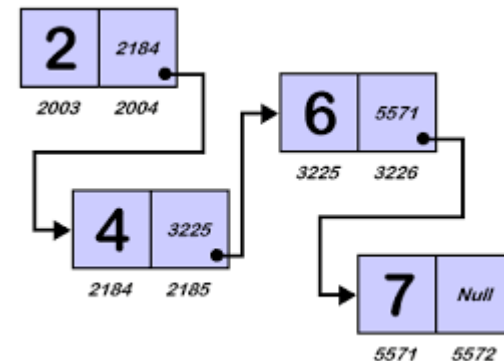




# Software



- São programas de computador (Conjunto de instruções de máquina)
- Apresentam **documentação** que descreve a operação e funcionalidade dos programas
- Quando executados provêem **funcionalidades** e **performance**
- Apresentam estruturas de dados que permitem acessar de modo adequado as informações
- Podem ser desenvolvidos para um cliente em particular ou para o mercado em geral





# Software é vida !

- Tecnologia indispensável para negócios
- Viabiliza a criação de novas pesquisas
- Mudança radical das tecnologias mais antigas
- Evolução de produto para serviço
- Incorporado em sistemas de todas as áreas



Robots





## Software: Duplo Papel

- **Produto**: representado pelo Hardware que o **processa**



- **Veículo** de distribuição da **informação**







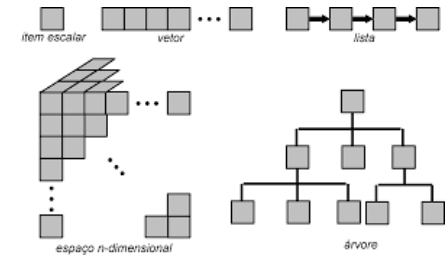
## Definindo Software



Software consiste de



1. **Instruções** (programas de computador que, quando executadas, fornecem funções e desempenhos desejados;
2. **Estruturas de dados**, que possibilitam aos programas manipular informações de forma adequada;
3. **Informação descritiva**, descrevendo a operação e o uso dos programas.



SOFTWARE  
≠  
PROGRAMA DE  
COMPUTADOR





Como seria o mundo de hoje sem software ?







Como seria a rotina de um banco sem Software ?





## E um supermercado ?





## Software está presente em todas as áreas

- Indústria
- Comércio
- Comunicação
- Entretenimento
- Sistema Financeiro
- Órgãos governamentais





## Categorias de Software

- Software Básico – Compiladores, Linkeditores, Debug, Sistemas Operacionais ...
- Aplicações Stand-Alone – Aplicações PC que não necessitam estar conectadas em rede (Lightroom, CAD ... )
- Aplicações Interativas baseadas em Transações – Emulação de Terminal Mainframe em PC acessando Mainframe
- Sistemas Batch – Geração de Backups, Atualização de BD, Relatórios...
- Sistemas embarcados – Software em microondas, automóveis...
- Entretenimento – Games
- Sistemas Científicos – Modelagem, Simulação







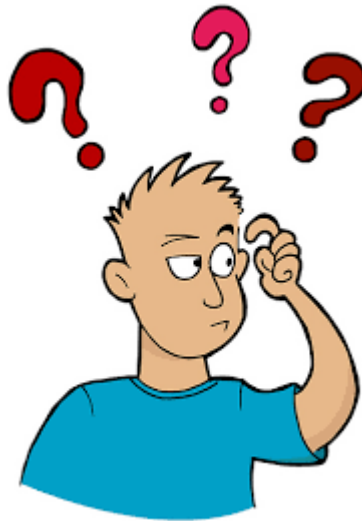
## Questões importantes

- Porque concluir um software leva tanto **tempo** ?
- Porque os **custos** de desenvolvimento são tão altos ?
- Porque não se consegue encontrar todos os **erros** antes de se entregar o software ?
- Por que se gasta tanto **esforço** para se manter software ?





Porque desenvolver software é uma tarefa complexa ?







# Complexidade

- Sistemas de software são abstratos e intangíveis;
- Não são regidos pelas leis da Física;
- Isso os torna extremamente complexos, difíceis de serem compreendidos e custosos para serem alterados.

```
var cep = {  
  valor : '',  
  validate : function(){  
    return this.valor.length == 8;  
  }  
};  
  
cep.valor = '12312043';  
var estaValido = cep.validate();
```

```
public List<Item> possui(User user) {  
  List<Item> result = new ArrayList<>();  
  for (Item item : itens) {  
    if (user.possui(item)) {  
      result.add(item);  
    }  
  }  
  return result;  
}
```



Produzir software é o mesmo que produzir hardware ?





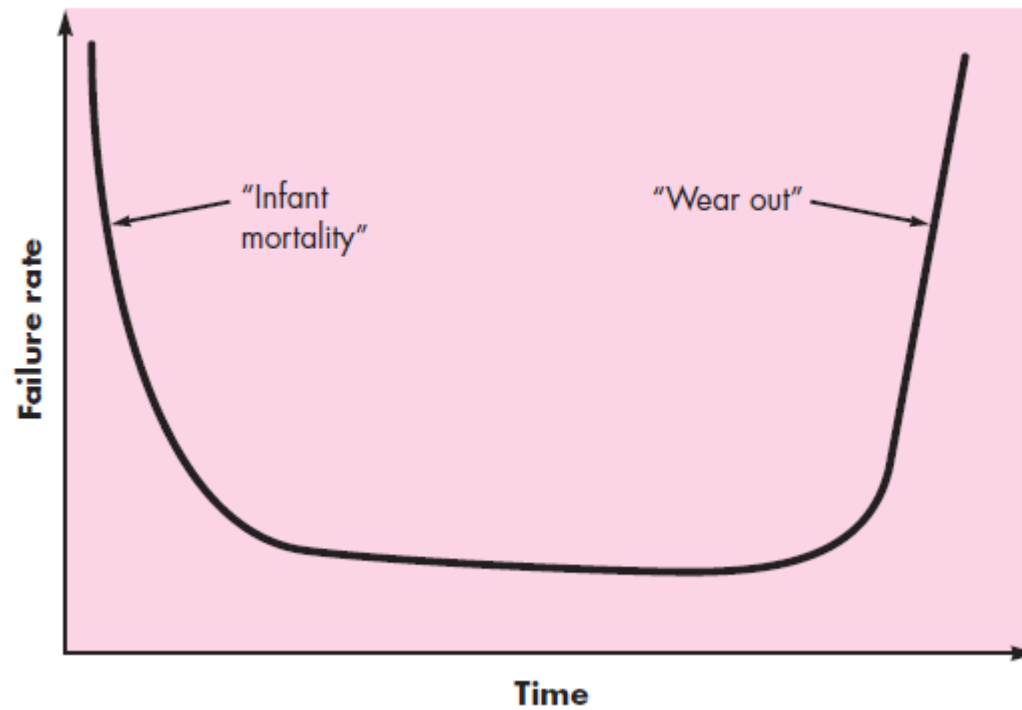
## Produzir software é o mesmo que produzir hardware ?

- As atividades são bem diferentes...
- Em ambas, alta qualidade é obtida por meio de um bom projeto, mas software não se desgasta...





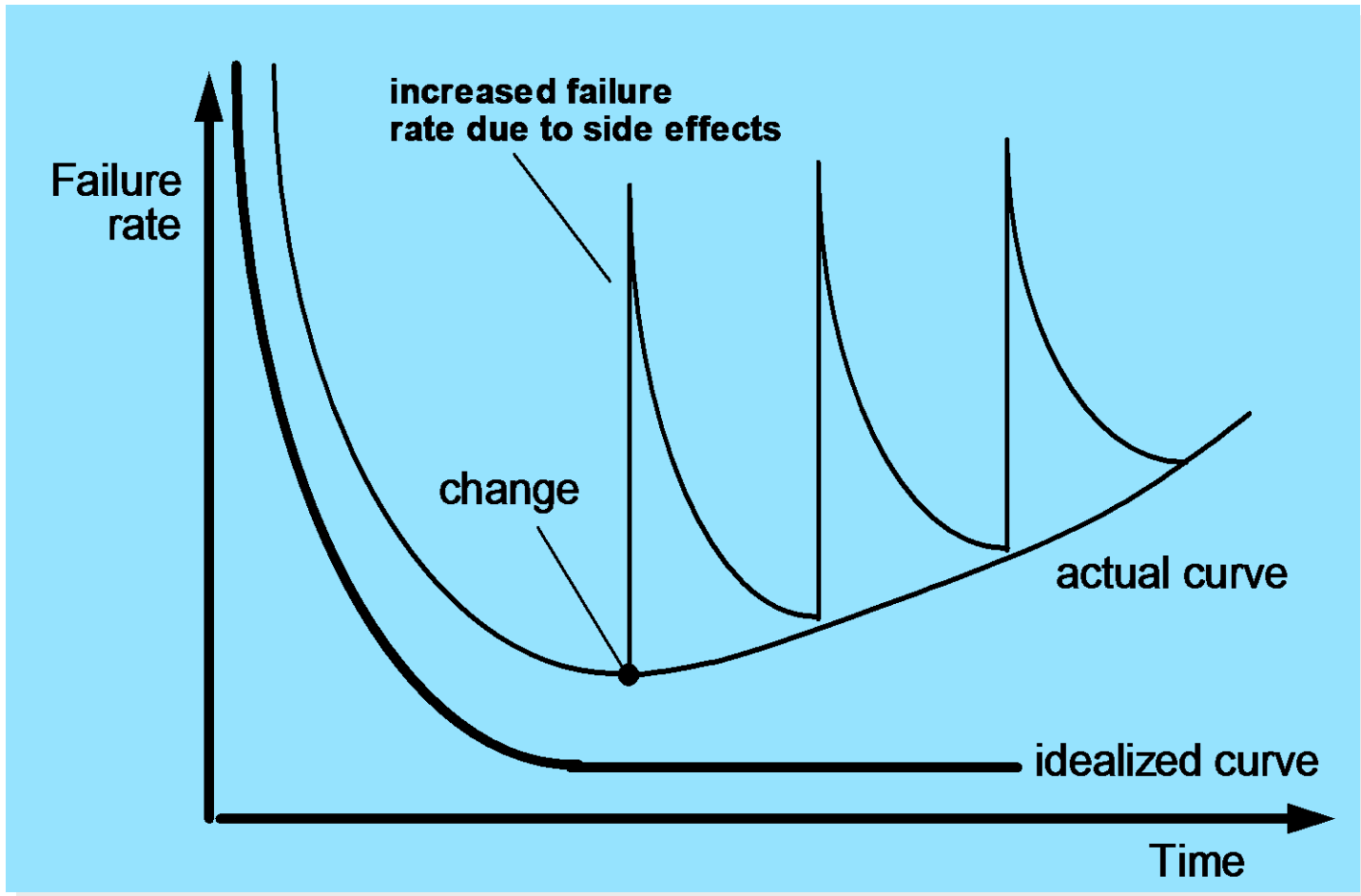
## Curva de Falhas para Hardware



Fonte: Pressman



## Curva de Falhas para Software



Fonte: Pressman



## Como o software é produzido ?



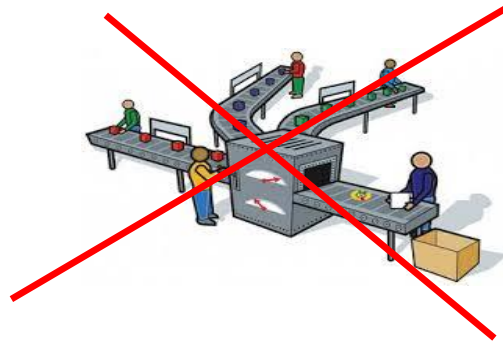




# Produção de Software



- Software não se desgasta com o tempo;
- Softwares são desenvolvidos, ou seja, NÃO são manufaturados no sentido clássico;



- Embora a indústria esteja se movendo em direção à construção baseada em componentes (component-based), a maior parte do software ainda é construído de forma customizada.



Quais são os atributos de um bom software ?





# Atributos de Software

- Bom software deve deliberar a funcionalidade requerida. Deve estar adequado ao tipo de usuário para o qual foi desenvolvido.;
- Deve atender aos requisitos de tempo de resposta, tempo de processamento, utilização de memória, etc. ;
- Deve ser manutenível e usável. Deve atender às mudanças do ambiente de negócios;
- Não deve apresentar impactos de ordem física ou econômica na ocorrência de falhas.





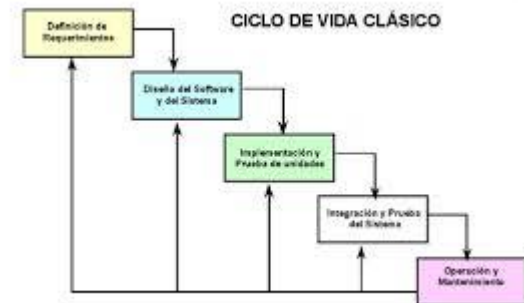
# O que é Engenharia de Software ?





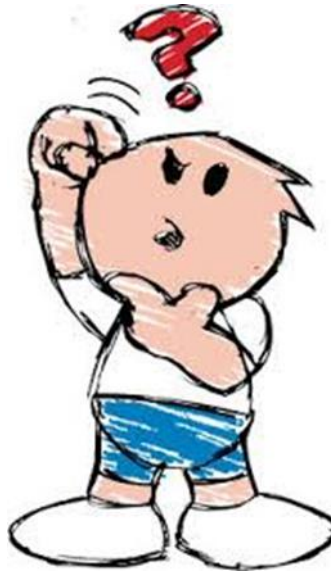
# Engenharia de Software

- É uma disciplina da Engenharia;
- Trata de todos os aspectos da produção de software, desde os primeiros estágios da especificação do sistema até a manutenção na produção.
- Permitem que os **custos** de desenvolvimento sejam **minimizados** e que os **prazos** de entrega sejam mais **realistas**.
- Cada vez mais, indivíduos e sociedade confiam e dependem de sistemas de software.





Quais são as atividades principais da Engenharia de Software ?







# Atividades da Engenharia de Software

■ Especificação



■ Desenvolvimento



■ Validação



■ Evolução





Qual a diferença entre Engenharia de Software e Ciência da Computação ?





- **Ciência da Computação** está focada no desenvolvimento de novas tecnologias e Teoria da Computação



- **Engenharia de Software** está focada na aplicabilidade prática para deliberar softwares úteis e viáveis





Qual a diferença entre Engenharia de Software e Engenharia de Sistemas ?





- **Engenharia de Sistemas** atua com todos os aspectos do desenvolvimento de sistemas baseados em computador, incluindo hardware, software e processos de Engenharia;



- **Engenharia de Software** é parte desse processo mais geral e foca diretamente na produção de software.





Quais os principais desafios enfrentados pela Engenharia de Software ?







# Desafios

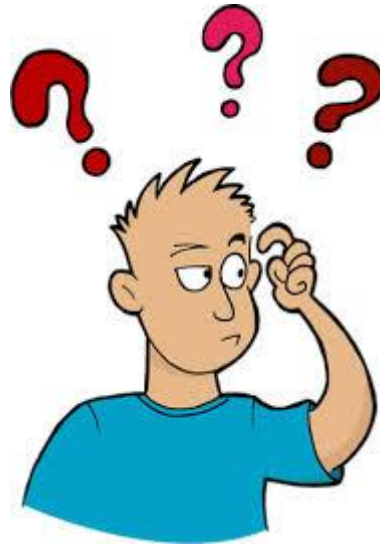


- Exigências de prazos de entrega reduzidos;
- À medida que novas técnicas de Engenharia de Software são empregadas, mais cresce a demanda por softwares complexos;
- Diversidade de aplicações (softwares embarcados, aplicações web, games, drivers, etc);
- Desenvolvimento de software confiável .





# Quanto custa desenvolver software ?





# Custos

■ **60%** no desenvolvimento



■ **40%** nos testes



Fonte: Software Engineering – Sommerville – 9th edition



É possível desenvolver software sem usar as técnicas da Engenharia de Software ?





## Sem o uso da Engenharia de Software ...

- O desenvolvimento de software sem o uso dos métodos e técnicas da Engenharia de Software, resulta em maiores custos de desenvolvimento, maior comprometimento de prazos e produtos menos confiáveis.





# Quando iniciou a Engenharia de Software ?





# Histórico

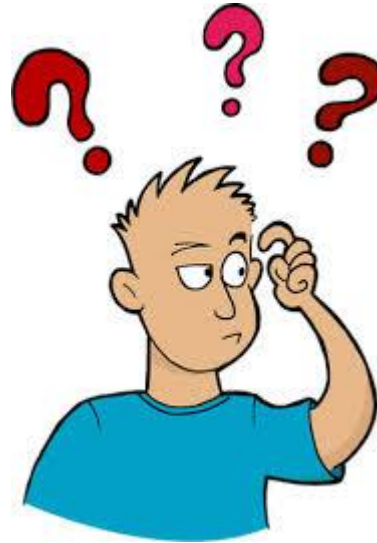
- A noção de Engenharia de Software foi primeiramente proposta em 1968, numa conferência na Alemanha, para discutir a “Crise de Software” (Naur e Randell, 1968).







## Crise de Software ?





# Crise de Software

- Expressão usada no final dos anos 60, quando a Engenharia de Software era praticamente inexistente;
- O termo expressava as dificuldades de desenvolvimento de software, causadas principalmente por:
  - ⊕ Custos elevados de desenvolvimento;
  - ⊕ Comprometimento de prazos;
  - ⊕ Baixa qualidade de software;
  - ⊕ Requisitos muitas vezes não atendidos;
  - ⊕ Projetos difíceis de serem gerenciados;
  - ⊕ Códigos difíceis de serem mantidos.





Qual o escopo da Engenharia de Software ?

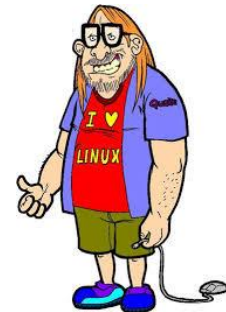




# Escopo



- Software profissional geralmente é desenvolvido em equipes;
- Engenharia de Software tem como meta suportar o desenvolvimento de software profissional e não o desenvolvimento individual;
- Inclui assim, técnicas que suportam especificação, design e manutenção, relevantes para a produção de software;
- Software profissional possui, frequentemente ,mais de um programa;
- Engenharia de Software não trata apenas código de programação, mas documentação e dados de configuração requeridos produção de software de qualidade.





# Quais os tipos de produtos de software ?





# Produtos de Software

- Produtos Genéricos
- Produtos Customizados



# Produtos Genéricos

- São sistemas stand-alone;
- Produzidos por uma empresa de software e vendidos no mercado para qualquer cliente;
- Exemplos: Processadores de Texto, ferramentas de gerenciamento de projetos, pacotes de desenho, etc.







# Produtos Customizados

- São sistemas que são desenvolvidos para atender as necessidades de um usuário em particular;
- Uma empresa contratada de software desenvolve o software especialmente especificado para o cliente;
- Exemplo: sistema escrito para suportar um particular processo de negócio.





# Observação

- Recentemente, a distinção entre produtos genéricos e produtos customizados tem-se tornado tênue, uma vez que tem-se desenvolvido software tomando-se por base um produto genérico e implementando-se adaptações às necessidades da empresa contratante.
- Exemplo: Sistemas **ERP – Enterprise Resource Planning**, tais como **SAP, TOTVS**.





# Software legado

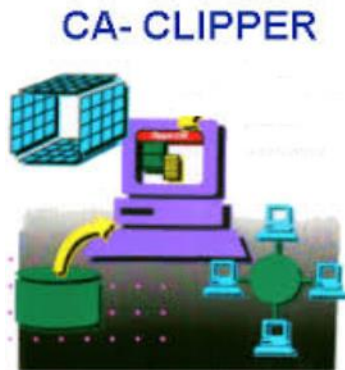
- São softwares **antigos** que tem sido continuamente modificados para atender as mudanças nos requisitos de negócios;
- Infelizmente, em geral, apresentam pobre qualidade, pouca ou inexistente documentação e casos de testes.





# Software legado

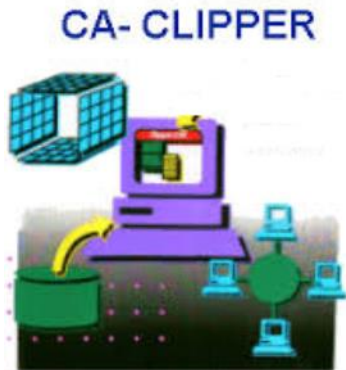
- Desenvolvidos décadas atrás;
- Tem sido constantemente modificados para se adequar às mudanças de requisitos de negócios e plataformas computacionais;
- Caracterizados pela longevidade e criticidade de negócios;
- Podem apresentar “baixa qualidade”.





# Software legado

- Podem ter projetos não expansíveis;
- Podem ter código complexo;
- Podem ter pobre ou inexistente documentação;
- Podem ter ausência de testes arquivados.





Mas, softwares legados ainda dão suporte à funções vitais do negócio e são indispensáveis para ele !!!





O que fazer então ?







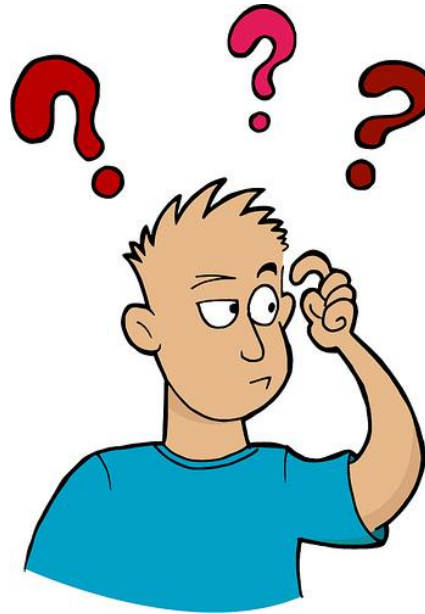
# O que fazer então ?

- Se ainda atende com suas funções básicas, o melhor é não fazer nada...
- Caso contrário, o sistema deve passar por uma reengenharia;
- Para isso, a Engenharia de Software contribui com metodologias baseadas na noção de evolução, ou seja, desenvolver sistemas com a capacidade de se evoluírem continuamente.





# Mito ou Verdade ?





# Mito ou Verdade ?

1. Gerente de software afirma que possui um livro repleto de padrões e procedimentos para desenvolver software. Esse livro supre a sua equipe com todas as informações que são necessárias para o desenvolvimento de software.



Fonte: Pressman



## Mito !!!

Gerente de software afirma que possui um livro repleto de padrões e procedimentos para desenvolver software. Esse livro supre a sua equipe com todas as informações que são necessárias para o desenvolvimento de software.

1. O livro é usado ?
2. A equipe sabe que o livro existe ?
3. O livro reflete a prática moderna de Engenharia de Software ?
4. O livro é completo ?
5. O livro é adaptável ?
6. O livro apresenta práticas que melhorem o tempo de entrega ?
7. O livro está focado em qualidade ?

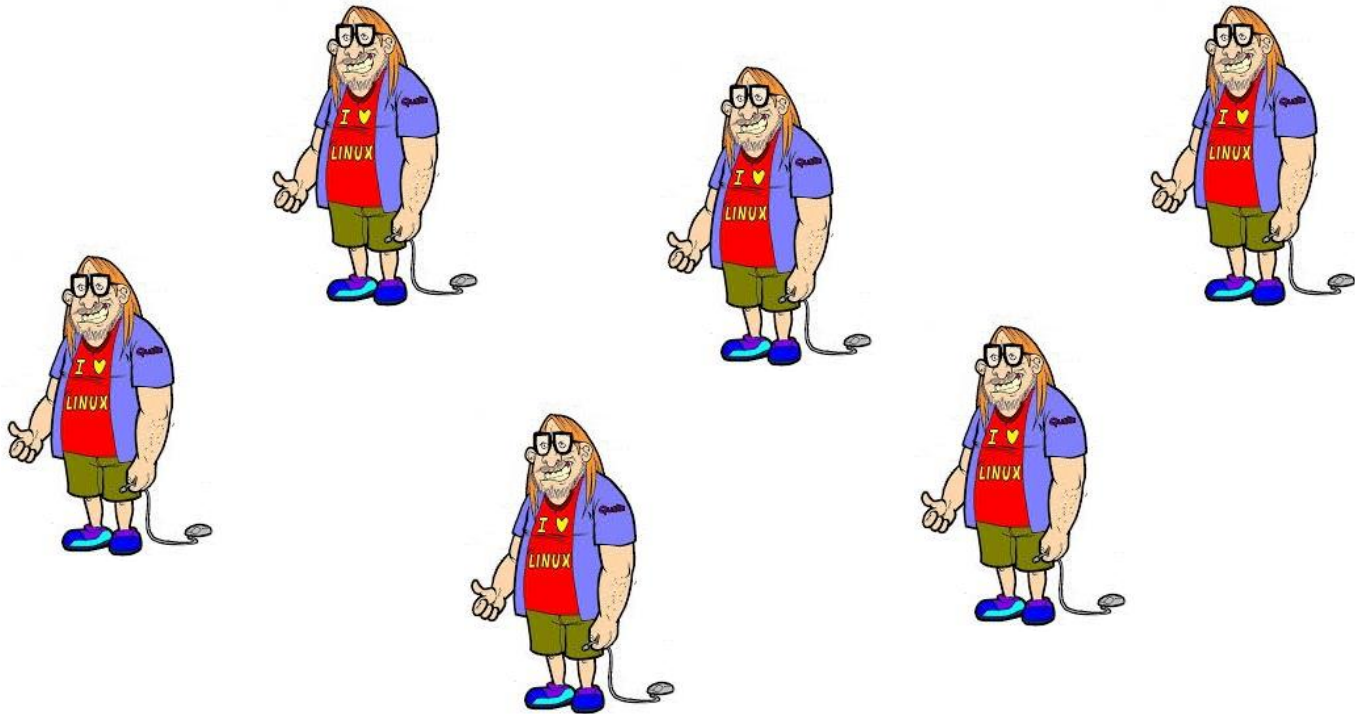
**A resposta para estas questões em geral é NÃO !**

Fonte: Pressman



# Mito ou Verdade ?

2. Se houver atrasos no cronograma, basta acrescentar mais programadores no projeto !



Fonte: Pressman



## Mito !!!

Se houver atrasos no cronograma, basta acrescentar mais programadores no projeto !

1. O desenvolvimento de software não é como o de um produto qualquer manufaturado numa indústria.
2. Acrescentar pessoas num projeto atrasado somente o atrasará mais ainda.
3. O que ocorre é que, quando novas pessoas entram em um projeto, as que já estavam terão de gastar tempo posicionando-as, reduzindo assim o tempo destinado ao projeto.
4. Pode-se adicionar recursos ao projeto, mas somente de forma **planejada** e bem coordenada.

Fonte: Pressman



## Mito ou Verdade ?

3. Se houver problemas no desenvolvimento do software, basta terceirizar o projeto e deixar por conta da empresa contratada.



Fonte: Pressman





## Mito !!!

Se houver problemas no desenvolvimento do software, basta terceirizar o projeto e deixar por conta da empresa contratada.

Se uma organização não sabe gerenciar e controlar um projeto de software, provavelmente terá as mesmas dificuldades ao terceirá-lo.



Fonte: Pressman



## Mito ou Verdade ?

4. Uma definição geral dos objetivos do software é suficiente para iniciar a codificação dos programas. Os detalhes podem ser definidos posteriormente.



Fonte: Pressman



## Mito !!!

Uma definição geral dos objetivos do software é suficiente para iniciar a codificação dos programas. Os detalhes podem ser definidos posteriormente

Embora nem sempre seja possível uma definição ampla e estável dos requisitos, uma definição de objetos ambígua é receita para um desastre.

**Ambiguidade:** “O menino viu a menina com a luneta”



Fonte: Pressman



# Mito ou Verdade ?

5. Os requisitos de software mudam continuamente, mas as mudanças são absorvidas pelo software, pois o software é flexível.



Fonte: Pressman



## Mito !!!

Os requisitos de software mudam continuamente, mas as mudanças são absorvidas pelo software, pois o software é flexível

- ◆ O impacto da mudança varia dependendo do momento em que ocorre.
- ◆ Quando a mudança ocorre no início do projeto, o impacto nos custos é relativamente pequeno. Entretanto, conforme o tempo passa o impacto aumenta de forma drástica.

Fonte: Pressman



## Mito ou Verdade ?

6. Uma vez feito o programa e o colocado em uso, nosso trabalho estará terminado.



Fonte: Pressman



# Mito !!!

Uma vez feito o programa e o colocado em uso, nosso trabalho estará terminado

- ◆ Levantamentos indicam que entre 60% e 80% de todo o esforço será despendido após a entrega do software ao cliente pela primeira vez.



Fonte: Pressman



## Mito ou Verdade ?

7. Até que o programa entre em funcionamento, não há maneira de avaliar sua qualidade.



Fonte: Pressman





## Mito !!!



Até que o programa entre em funcionamento, não há maneira de avaliar sua qualidade.

- ◆ A revisão técnica pode ser efetuada desde a concepção de um projeto;
- ◆ Esse mecanismo corresponde a um “filtro de qualidade” e mostra ser mais eficiente do que testes para encontrar-se defeitos.

Fonte: Pressman



## Mito ou Verdade ?

8. O único produto passível de entrega é o programa em funcionamento.

```

010000000001010001101100000010010110001
11000101110100010001111111110100000100
0010100101100001101011101101011011001000
0110110000010101100100010000111000100111
01001100101101001101101001111011101110
0001101001#include <stdio.h>01101000011010
1001001101001010001110
10001001int main()000010111
01010001{00011000
111001100printf("Hello World");0001100
001000001return 42;010101110110
000110100010001110001101000011010
10010011011101011101110000001010001110
100010010001010110010011101110100010111
01010001110011010101110001010100011000
11100110000011011111010100111110001100
00100000111111010100100100110101110110
  
```

Fonte: Pressman



O único produto passível de entrega  
é o programa em funcionamento

- Fonte: Pressman



# Mito ou Verdade ?

9. A Engenharia de Software nos fará criar documentação volumosa e desnecessária e, invariavelmente, irá nos retardar.



Fonte: Pressman



## Mito !!!

A Engenharia de Software nos fará criar documentação volumosa e desnecessária e, invariavelmente, irá nos retardar.

- ◆ A Engenharia de Software não trata de criação de documentos, trata da criação de um software de qualidade;
- ◆ Melhor qualidade conduz à redução do retrabalho;
- ◆ Menor retrabalho resulta em maior rapidez na entrega.

Fonte: Pressman



## Referências Adicionais

- ◆ Anderson, E. et al., Software Engineering for Internet Applications, The MIT Press, 2006
- ◆ Glass, R., Fact and Fallacies of Software Engineering, Addison-Wesley, 2002
- ◆ Jalote, P., An integrated Approach to Software Engineering, Springer, 2006
- ◆ Schach, S., Object-Oriented and Classical Software Engineering, 7ª edição, McGraw-Hill, 2006
- ◆ Tsui, F. and O. Karam, Essentials of Software Engineering, Jones & Bartlett Publishers, 2006
- ◆ Pfleeger, S., Software Engineering: Theory and Practice, 4ª ed. , Prentice-Hall, 2010

