



Instituto Federal de Educação, Ciência e Tecnologia da Bahia - IFBA
Departamento de Ciência da Computação
Tecnólogo em Análise e Desenvolvimento de Sistemas

Conceitos básicos de Engenharia de Software

André L. R. Madureira <andre.madureira@ifba.edu.br>
Doutorando em Ciência da Computação (UFBA)
Mestre em Ciência da Computação (UFBA)
Engenheiro da Computação (UFBA)

Importância da Engenharia de Software

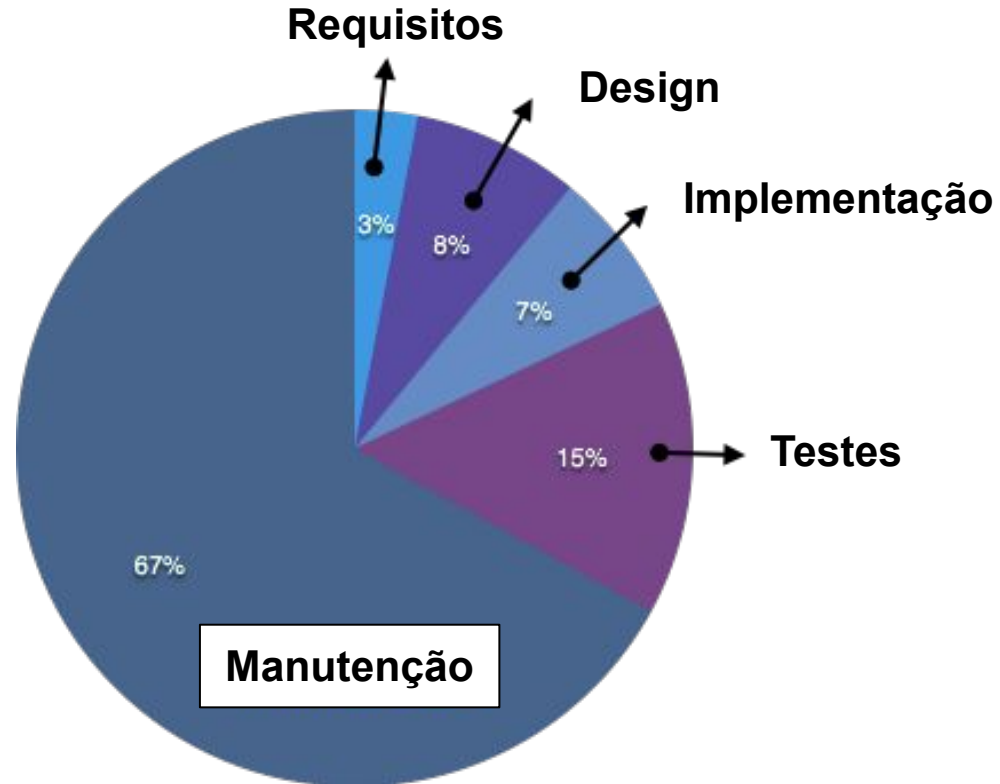
- **A engenharia de software:**
 - Facilita a construção
 - Gerencia a complexidade
 - Reduz custos com a manutenção

Como é possível construir softwares de alta qualidade sem a engenharia de software?



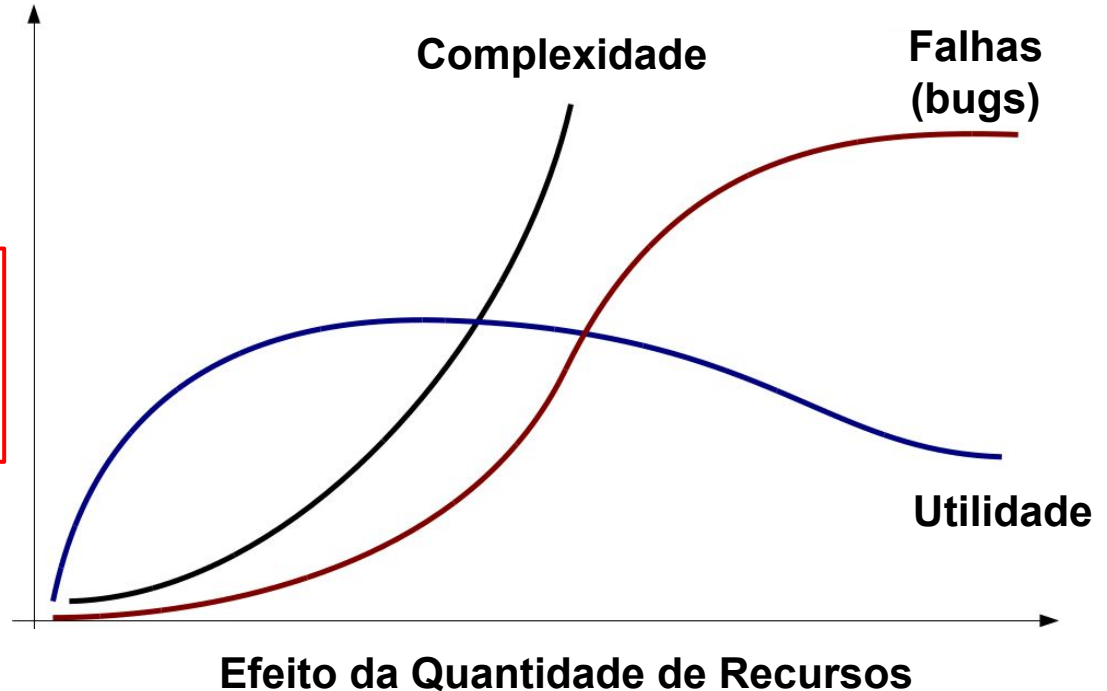
Importância da Engenharia de Software

Maior custo de um software é com **manutenção**
(correção de bugs, novos recursos, etc)



Importância da Engenharia de Software

Softwares têm uma
tendência de **crescer**
com o tempo



Objetivos da Engenharia de Software

- **Objetivos**

- **Facilitar** as etapas de desenvolvimento de software
- **Reduzir tempo e custos** envolvidos nas etapas
- Produzir software de **maior qualidade**

- **O que é um software de “alta qualidade”?**



Software de Qualidade

- Um bom software possui os seguintes atributos essenciais:
 - **Confiabilidade e Proteção**
 - **Eficiência**
 - **Aceitabilidade**
 - **Manutenibilidade**

Confiabilidade e Proteção

- Aplicações críticas exigem softwares seguros e confiáveis
 - Quando o software falha consequências danosas podem ocorrer
 - Doses letais de radiação em pacientes (Therac 25 - 1985)
 - Explosões de foguetes (Foguete Ariane 5 - 1996)
 - Problemas no recolhimento de impostos (Reino Unido - 2004)

Confiabilidade e Proteção

- Sistemas podem exigir altos níveis de segurança, pois:
 - Eles contêm dados sigilosos ou privados (LGPD)
 - Gerenciam sistemas críticos (usinas, hospitais, aeronaves, bancos, carros, etc)

Hacker é condenado por vazar fotos de celebridades nuas



**HACKER DE 19 ANOS ROUBAVA
SENHAS DE BANCOS**

**Setor de saúde tem epidemia de hackers – e
Brasil é um dos mais expostos**

Eficiência

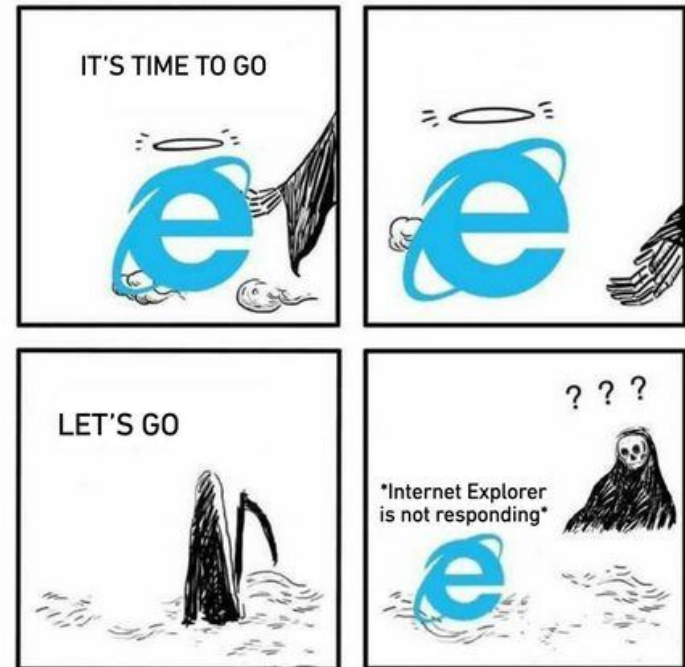
- Sistemas ineficientes podem representar desde um aborrecimento simples para o usuário, até levar a consequências fatais



painel de controle do piloto automático do Cessna 182

Aceitabilidade

- O software deve ser aceitável para o tipo de usuário para o qual ele foi projetado.
- Isto é, ele deve **atender** às necessidades dos usuários e estes devem “**querer/aceitar**” usar o sistema



Manutenibilidade

- Um software de difícil manutenção tende a acumular bugs, além de comprometer a adição de novas funcionalidades
- Bons softwares estão em constante evolução e são fáceis de dar manutenção
 - *“Um software que não está em constante evolução, ou deixou de ser utilizado, ou está fadado ao desuso”*

Exercício

Considerando o escopo da engenharia de software, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - Softwares tendem a crescer, o que aumenta a sua utilidade pois eles possuem mais recursos. **F**

II - Um dos objetivos da engenharia de software é produzir softwares de maior qualidade. Contudo, o uso de técnicas de engenharia de software aumentam o tempo necessário para o desenvolvimento de software. **F**

III - Os atributos de um software de qualidade incluem a confiabilidade, proteção, eficiência, aceitabilidade, e manutenibilidade. **V**

IV - Manutenibilidade é uma propriedade de um software que esta em constante evolução, sendo fácil de dar manutenção em seu código. **V**

☐ Todas as assertivas são verdadeiras.

☐ Somente I, III, e IV.

☒ Somente III e IV.

☐ Somente III.

☐ Somente IV.

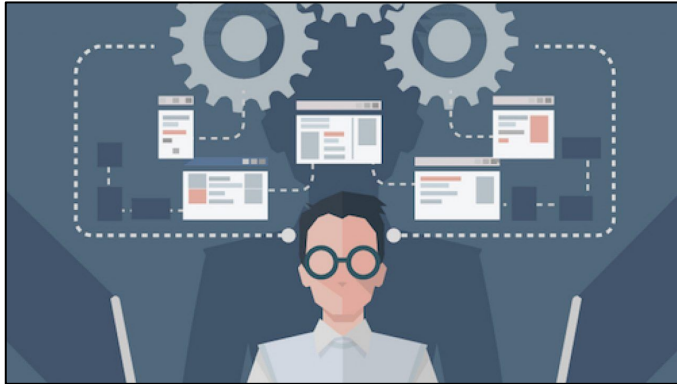
Seguem as assertivas com os ajustes necessários para torná-las VERDADEIRAS:

I - Softwares tendem a crescer, POREM isto reduz a sua utilidade pois a presença de mais recursos no software induz falhas (bugs) no sistema.

II - Um dos objetivos da engenharia de software é produzir softwares de maior qualidade. O uso de técnicas de engenharia de software REDUZEM o tempo necessário para o desenvolvimento de software, pois aumentam a qualidade do código e facilitam o seu entendimento. Isto é, a engenharia de software facilita a manutenção de sistemas complexos.

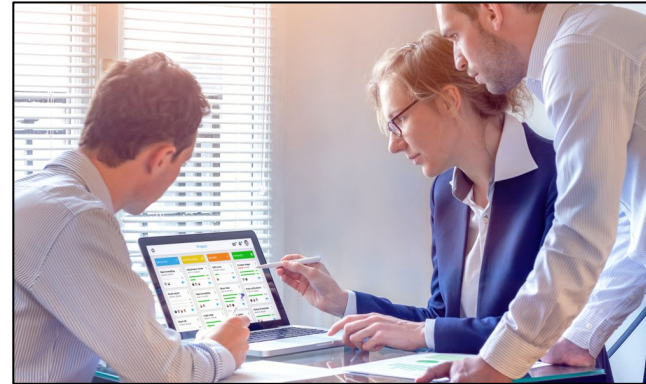
Engenharia de Software

Processos técnicos do desenvolvimento de software




Ex: levantamento dos requisitos e projeto do sistema

Atividades de apoio



Ex: gerenciamento do projeto

Processo de Software

- Sequência de atividades que leva a produção de um software
 - **Especificação**
 - **Desenvolvimento** (Projeto e Implementação)
 - **Verificação e Validação** 

Foco dessa disciplina
 - **Evolução**

Especificação

Regras de negócio
(objetivo da
empresa, propósito
do software, ...)



Requisitos
(o que o software
deve ser capaz de
fazer e sob quais
condições)

Especificação

- Clientes e engenheiros **definem os requisitos do software**, que podem ser classificados como:

Requisitos funcionais:
o que o software deve fazer?

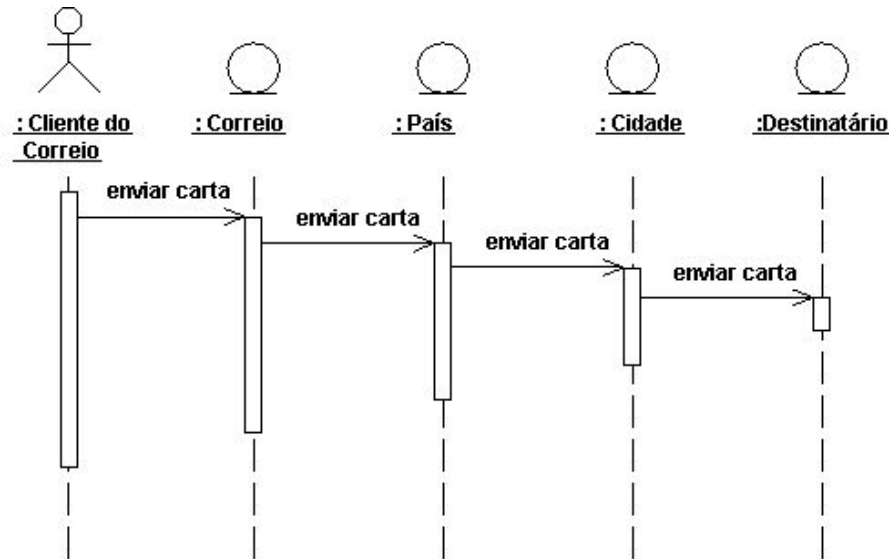


Requisitos não-funcionais:
condições e restrições



Projeto de Software

- No projeto de software, a estrutura e comportamento do sistema são descritos



Implementação de Software

- Construção efetiva do programa

```
struct icmphdr _hdr;  
const struct icmphdr *hdr;  
  
hdr = skb_header_pointer(skb, skb_transport_offset(skb),  
                          sizeof(_hdr), &hdr);  
if (!hdr)  
    return 1;
```

Deve atender aos requisitos do sistema levantados na atividade de especificação

Deve seguir o projeto do sistema construído na atividade anterior

Verificação e Validação

- **Verificar o software** através de técnicas e testes
- **Objetivo:**
 - Garantir que ele **atende às especificações** do cliente
 - Garantir que ele **cumpre** os seus **requisitos**
 - **Identificar falhas** (bugs)



Evolução

- São feitas **modificações no software** para
 - Refletir ajustes nos requisitos
 - Corrigir erros (bugs)

Internet Explorer era **lento e cheio de bugs**

Faltavam recursos que outros navegadores tinham
(Ex: extensões)



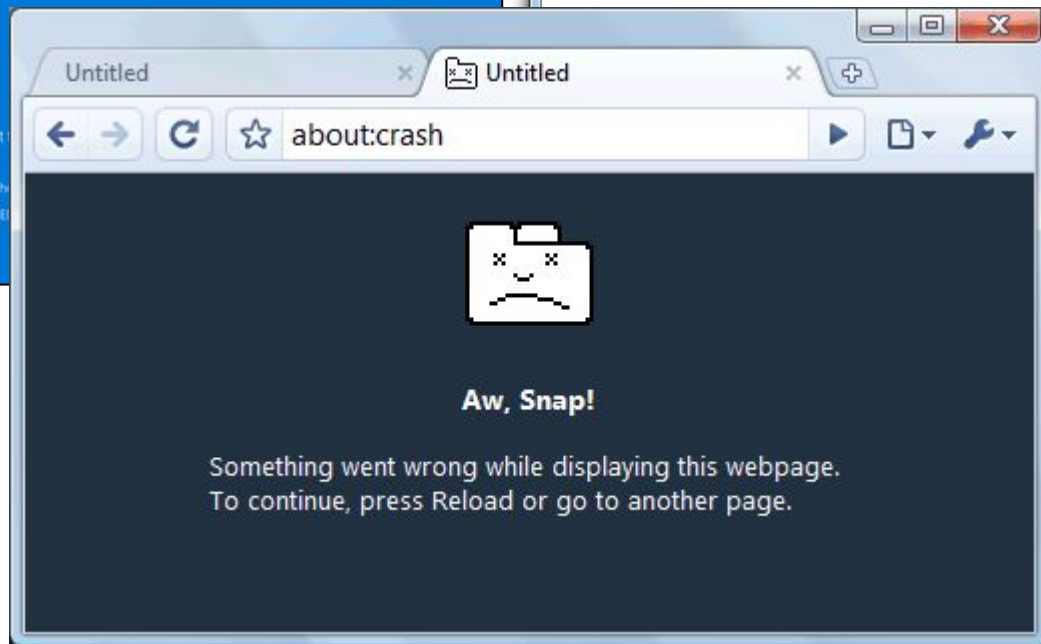
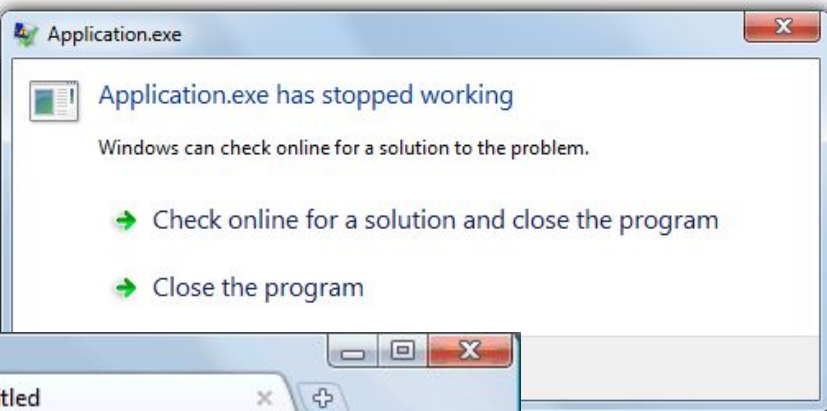


Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete



For more information about:
If you call a support person, give them
Stop code: CRITICAL_PROCESS_DIED



Exercício

Considerando o escopo da engenharia de software, marque a alternativa que contém **somente** as assertivas VERDADEIRAS.

I - A engenharia de software é composta por processos técnicos e atividades de apoio ao desenvolvimento de software. **V**

II - As atividades de especificação e desenvolvimento são as mais importantes dentre as atividades que compõem um processo de software. Isto se deve ao fato do software ser concebido e construído nessas atividades, sendo essas atividades as que representam os maiores custos associados a um sistema. **F**

III - Ao término da atividade de especificação do sistema, a próxima atividade a ser desenvolvida é a implantação do software. **F**

IV - A atividade verificação e evolução é responsável por atestar que o software atende às especificações e requisitos do cliente. Já a atividade de evolução corrige os erros identificados no sistema, sendo responsável também por realizar ajustes em seus requisitos. **V**

☐ Todas as assertivas são verdadeiras.

☐ Somente I, II e IV.

☒ Somente I e IV.

☐ Somente II.

☐ Somente II e IV.

Seguem as assertivas com os ajustes necessários para torná-las VERDADEIRAS:

II - As atividades de especificação e desenvolvimento **NÃO** são as mais importantes dentre as atividades que compõem um processo de software. APESAR do fato do software ser concebido e construído nessas atividades, elas não representam os maiores custos associados a um sistema. Os maiores custos estão atrelados a manutenção do sistema.

III - Ao término da atividade de especificação do sistema, a próxima atividade a ser desenvolvida é o DESENVOLVIMENTO do software, que pode ser subdividido em duas atividades: projeto e implementação.

Ferramentas CASE

- **CASE** = Engenharia de software assistida por computador
- Toda ferramenta que fornece suporte às atividades de engenharia de software
- São classificadas de acordo com a sua finalidade:

**Automação de
Compilação**

**Teste de
software**

**Controle de
versões**

**Modelagem do
sistema**

**Automação de
Documentação**

**Ambiente de
desenvolvimento integrado**

**Automação de
Implantação**

Automação de compilação

- Facilita a compilação de projetos grandes
 - Melhora o desempenho da compilação
 - Evita problemas relacionados a dependências



Teste de software

- Facilitam a execução de testes automatizados
 - Existem varios tipos de testes que visam verificar e validar o funcionamento de diferentes funcionalidades e partes do sistema



Controle de Versões

- Facilita identificação e correção de falhas em atualizações de software
 - Permite identificar que partes do código foram alteradas, quando as alterações foram feitas, e quem as fez



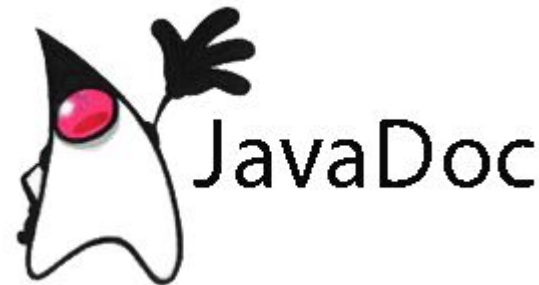
Modelagem do sistema

- Auxilia o projetista na construção de modelos do sistema
 - Reduz o tempo necessário para construção de modelos
 - Permite localizar falhas no projeto com maior facilidade



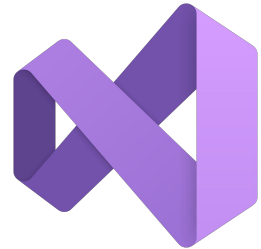
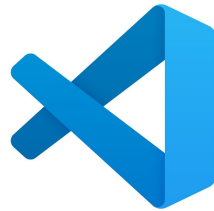
Automação de documentação

- Auxilia na criação de documentações sobre um sistema
 - **Objetivo:** descrever como utilizar um sistema e como ele foi projetado
 - A documentação pode ser destinada ao usuário final ou aos desenvolvedores

The logo for Doxygen, featuring the word "doxygen" in a blue, 3D-style font with a blue underline.

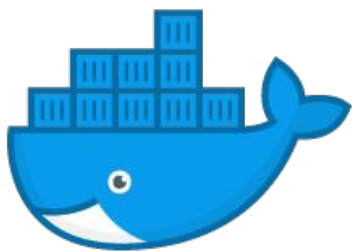
Ambiente de desenvolvimento integrado (IDE)

- Auxilia o desenvolvedor nas atividades diretamente relacionados ao desenvolvimento de software
 - Programação, verificação de sintaxe, compilação, etc



Automação de implantação (*deployment*)

- Auxilia o desenvolvedor na instalação, configuração e atualização de um sistema nas máquinas dos usuários (ambiente de produção)



docker



kubernetes



ANSIBLE

Seminário

- Escolham um dos grupos de ferramentas CASE abaixo, e dentro dessa categoria, escolha uma ferramenta para apresentar no seminário (ex: Git, JUnit, Eclipse, etc)
 - Os integrantes da equipe, apresentação e relatório devem ser postados no Classroom

**Automação de
Compilação**

**Teste de
software**

**Controle de
versões**

**Modelagem do
sistema**

**Automação de
Documentação**

**Ambiente de
desenvolvimento integrado**

**Automação de
Implantação**

Referencial Bibliográfico

- SOMMERVILLE, Ian. **Engenharia de Software**. 6. ed. São Paulo: Addison-Wesley, 2003.
- PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: Makron Books, 1995.
- JUNIOR, H. E. **Engenharia de Software na Prática**. Novatec, 2010.

Obrigado!

- Perguntas?

