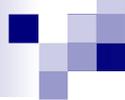


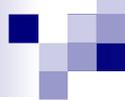
Modelos, Métodos e Técnicas de Engenharia de Software Introdução

**Prof. Osmar de Oliveira Braz Junior
Prof. Richard Henrique de Souza**



Objetivos

Apresentar a Engenharia de Software e conceitos fundamentais.



Introdução

O que é?

Motivadores?

O que é um software de qualidade?

O que são processos de desenvolvimento?

Software



Consiste em:

- (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados;
- (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente; e
- (3) informação descritiva, tanto na forma impressa quanto na virtual, descrevendo a operação e o uso dos programas.

Natureza do Software

Ele é um **produto** e o veículo para **distribuir** um produto.

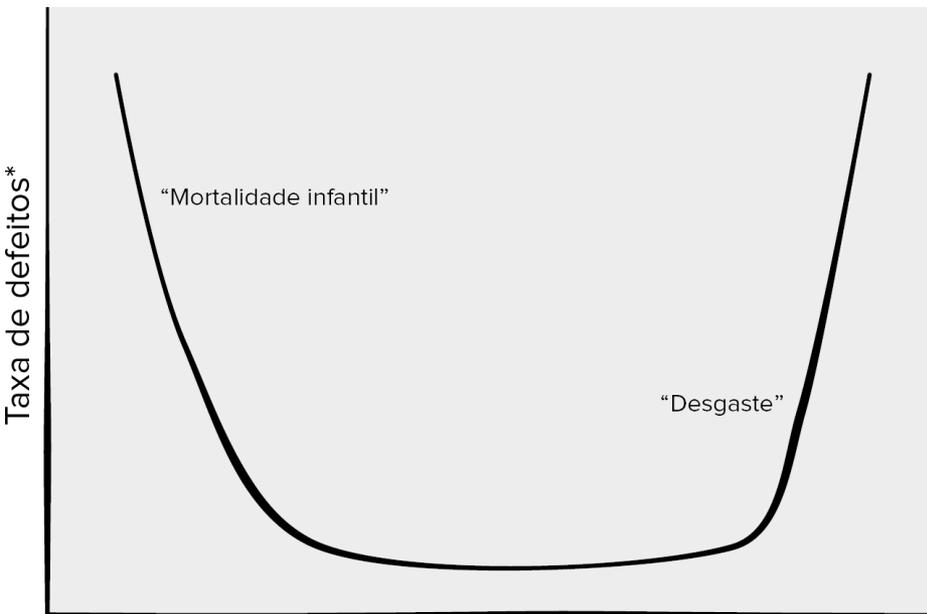
- **Produto**, fornece o potencial computacional representado pelo hardware ou, de forma mais abrangente, por uma rede de computadores que podem ser acessados por hardware local.
- **Veículo de distribuição** do produto, o software atua como a base para o controle do computador (sistemas operacionais), a comunicação de informações (redes) e a criação e o controle de outros programas (ferramentas de software e ambientes).

O software distribui o produto mais importante de nossa era – **a informação**

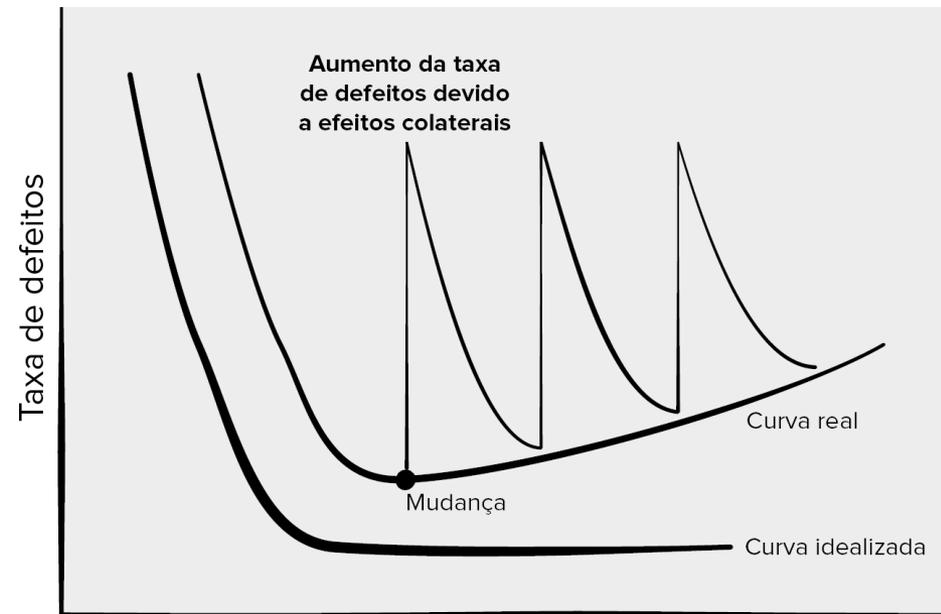
Natureza do Software

Pressman, 2021

- Software é mais um elemento de sistema lógico do que físico.
- Portanto, o software tem uma característica fundamental que o torna consideravelmente diferente do hardware: o **software não “se desgasta”**. (Pressman, 2021)



Tempo
Curva de defeitos no hardware



Tempo
Curva de defeitos no software

Engenharia

- **Engenharia** é a prática da aplicação segura e econômica das leis científicas que governam as forças e materiais da natureza, por meio da organização, design e construção, para o benefício da humanidade. S. E. Lindsay (1920);
- **Engenharia** é a ciência e a arte de tratar eficientemente com materiais e forças. (...) Envolve o design e a construção de forma mais econômica (...), assegurando, quando realizado adequadamente, a combinação mais vantajosa de acuidade, segurança, durabilidade, velocidade, simplicidade, eficiência e economia possível para as condições de design e serviço. A. L. Waddel, Frank W. Skinner, Wessman (1933);
- **Engenharia**, em um sentido amplo, é a aplicação da ciência de maneira econômica para as necessidades da humanidade. Vanevar Bush (1939);

Engenharia de Software

- É uma **disciplina de engenharia** cujo o foco está em todos os aspectos da **produção** de software, desde os estágios iniciais da especificação do sistema até a sua manutenção, quando o sistema já está sendo usado. (Sommerville, 2011:5).
- Engenharia de Software é a **aplicação** de uma abordagem **sistemática, disciplinada e quantificável** ao **desenvolvimento, operação e manutenção** de software(IEEE Std 610.12 (1990)).

Camadas da Engenharia de Software

A engenharia de software é uma tecnologia em camadas.



Camadas da Engenharia de Software

Engenharia de Software

■ Logo, percebemos que a **Engenharia de Software** envolve:

■ Os processos de software, a engenharia de requisitos, a modelagem de sistemas, os testes de software, o reuso de software, os componentes de software, a arquitetura, o gerenciamento de projetos, a qualidade de software, a melhoria dos processos etc.

Mas eu já sei programar!

- Por que preciso de Engenharia de Software?
 - **Programar** é parte **importante** do processo de Engenharia de Software, mas não é tudo!
- Precisamos também saber...
 - como estimar um projeto (tamanho, custo, cronograma);
 - como monitorar o andamento de um projeto;
 - como testar o software;
 - como controlar a evolução do software;
 - como modelar um software, etc...

Programas de universidade/faculdade

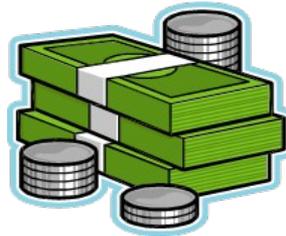
- Requisitos estáveis e bem definidos;
- Escopo pequeno (1-10 KLocks);
- Prazos razoáveis;
- Equipes pequenas;
- Mão de obra gratuita;
- Não entra em produção;
- Ausência de cliente;
- Ausência de manutenção;

Kilo Lines of Code
(mil linhas de código)

Programas no mundo real

■ Fazer software no “**mundo real**” deve considerar fatores como:

- Escopo
- Custo
- Prazo
- Qualidade



■ Em função do tamanho do software, esses fatores se tornam difíceis de garantir.

Cenário 1: Agenda pessoal

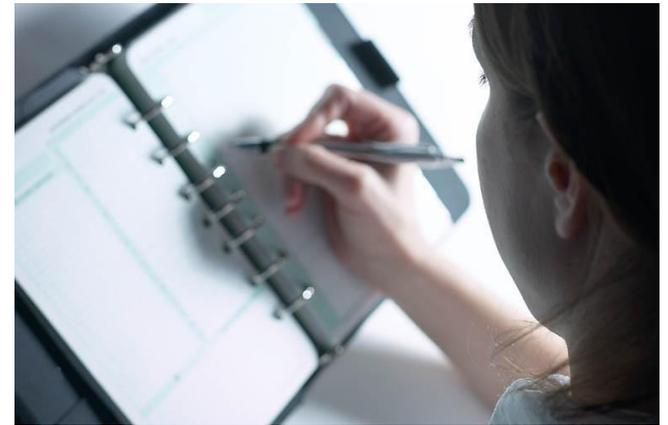
■ Objetivo

- Guardar o nome e o aniversário de até 50 pessoas

■ Quanto custa para fazer?

■ Quanto tempo vai levar para ficar pronto?

■ Qual a consequência no caso de defeito?



Cenário 2: Boeing 777

- Objetivo
 - Controlar todo o hardware do Boeing 777
- Quanto custa para fazer?
- Quanto tempo vai levar para ficar pronto?
- Qual a consequência no caso de defeito?



Cenário 2: Boeing 777

■ Tamanho

- Mais de 4 milhões de linhas de código
- Linguagem dominante (>99%): Ada

■ Documentação

- De 100 a 10.000 páginas por sub-sistema
- Total de 79 sub-sistemas integrados

■ Duração

- 4,5 anos de desenvolvimento

■ Ampla utilização de **Engenharia de Software**

■ Em operação **desde 1995**

- Zero acidentes graves até 2006

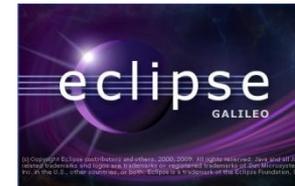


<http://www.stsc.hill.af.mil/crosstalk/1996/01/Boein777.asp>

<http://www.boeing.com/news/techissues/pdf/statsum.pdf>

Outros cenários extremos...

- Toyota Lexus LS460: > 7 MLOCs
- Eclipse: 24MLOCs e
- Windows XP: 40 MLOCs
 - 1800 desenvolvedores
 - 2200 testadores
- SAP: 250 MLOCs
- Debian GNU/Linux 4: 283 MLOCs
 - 1000 desenvolvedores



Mas quanto grande é isso?

- Assuma que **uma folha A4** tem em torno de 50 linhas.
- Assuma que uma pilha de **1000 folhas A4** tem em torno de 10 centímetros de altura
- Assim, **1.000.000 de LOCs**, caso impresso, seria uma pilha de **2 metros de altura!**
- Caso todo o código do Debian GNU/Linux fosse impresso, teria a altura de um prédio de **188 andares!!!**



Construindo aviões no ar!



<https://www.youtube.com/watch?v=J15itabl2xw>

Mas fazer software não é arte?

Parte arte, parte engenharia...

- Se o cantor/ator/pintor errar, a audiência fica chateada
- Se o engenheiro civil errar o prédio pode cair
- Se o médico errar o paciente pode morrer

Se o desenvolvedor de software errar o que pode acontecer?

Mas fazer software não é arte?

Se o desenvolvedor de software errar o que pode acontecer?

- Liste 2 problemas:

<https://www.menti.com/ale5i8sync>

Expira em 10/09/2023



respostas em

<https://www.mentimeter.com/app/presentation/al5xqzviuzwa12jrpTxporm56zqkefj/e941awm5isia/edit>

Caso real 1: Therac-25

■ Máquina de radioterapia controlada por computador

■ Problema:

- Doses indevidas de radiação emitidas

■ Causa:

- Interface com usuário inapropriada
- Documentação deficiente
- Software reutilizado sem ser adaptado para o novo hardware
- Software de sensores de falha com defeito

■ Consequências

- Ao menos 5 mortes entre 1985 e 1987



Caso real 2: Ariane 5

- Foguete lançador de satélites
- Problema:
 - O foguete se auto-destruiu 40 segundos após o lançamento
- Causa:
 - Software reutilizado sem ser adaptado para o hardware
 - Ausência de testes em solo deste software
 - Defeito apresentado em vôo



Motivação extra para estudar?

- Diversos concursos e oportunidades de emprego exigem conhecimento de Engenharia de Software
- Alguns exemplos:



Como será a UC?

Só os Métodos
clássicos prestam!



Só os Métodos
Ágeis prestam!



Como será UC MMT?

- Veremos **sem preconceito** técnicas clássicas e ágeis de Engenharia de Software
- Utilizaremos o que considerarmos melhor cada situação
- Mas... o processo base que utilizaremos é iterativo, incremental e ágil

Ciência da Computação X Engenharia de Software

- Ciência da computação foca a teoria e os fundamentos; a engenharia de software preocupa-se com o lado prático do desenvolvimento e entrega de software úteis. (SOMMERVILLE, 2011:4)

Engenharia de Sistemas X Engenharia de Software

- Engenharia de sistemas se preocupa com todos os aspectos do desenvolvimento de sistemas computacionais, incluindo engenharia de hardware, software e processo. Engenharia de software é uma parte mais específica desse processo mais genérico. (SOMMERVILLE, 2011:4).

Histórico (pré ESW)

- **1940s:** Primeiro computador eletrônico de uso geral – ENIAC
 - Custo estimado de US\$ 500.000,00
 - Início da programação de computadores
- **1950s:** Primeiros compiladores e interpretadores
- **1960s:** Primeiro grande software relatado na literatura – OS/360
 - Mais de 1000 desenvolvedores
 - Custo estimado de US\$ 50.000.000,00 por ano
- **1968:** Crise do software – nasce a Engenharia de Software

Histórico (era pós ESW)

■ 1970s:

- Lower-CASE tools (programação, depuração, colaboração)
- Ciclo de vida cascata
- Desenvolvimento estruturado

■ 1980s:

- Ciclo de vida espiral
- Desenvolvimento orientado a objetos

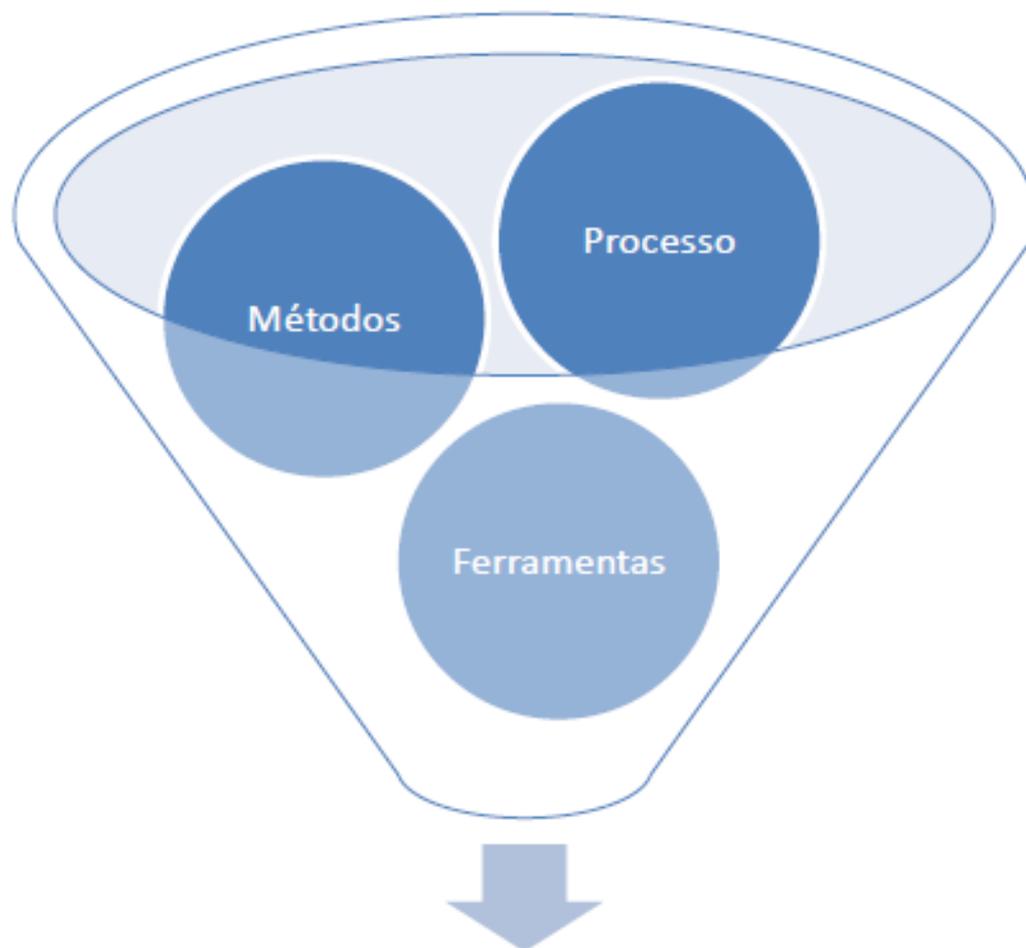
■ 1990s: Upper-CASE tools

- Processos
- Modelagem

■ Atualmente

- Métodos ágeis
- Desenvolvimento dirigido por modelos
- Linhas de produto
- Experimentação

Elementos da ES



Engenharia de Software

Elementos da ESW

■ Processo

- Define os passos gerais para o desenvolvimento e manutenção do software
- Serve como uma estrutura de encadeamento de métodos e ferramentas

■ Métodos

- São os “how to’s” de como fazer um passo específico do processo

■ Ferramentas

- Automatizam o processo e os métodos

Elementos da ESW

- Cuidado com o “desenvolvimento guiado por ferramentas”
- É importante usar a **ferramenta** certa para o problema;
- O problema não deve ser adaptado para a ferramenta disponível



“Para quem tem um martelo, tudo parece prego”

Elementos da ESW

1. Coloque em uma panela funda o leite condensado, a margarina e o chocolate em pó;
2. Cozinhe [no fogão] em fogo médio e mexa sem parar com uma colher de pau;
3. Cozinhe até que o brigadeiro comece a desgrudar da panela;
4. Deixe esfriar bem, então unte as mãos com margarina, faça as bolinhas e envolva-as em chocolate granulado.

**O que é
processo,
método ou
ferramenta?**

Elementos da ESW

1. **Coloque** em uma **panela** funda o leite condensado, a margarina e o chocolate em pó;
2. **Cozinhe** [no **fogão**] em fogo médio e **mexa** sem parar com uma **colher de pau**;
3. **Cozinhe** até que o brigadeiro comece a desgrudar da panela;
4. Deixe esfriar bem, então **unte as mãos** com margarina, **faça as bolinhas** e **envolva-as** em chocolate granulado.



método



ferramenta

Processo

Supermercado da ESW

- ESW fornece um conjunto de métodos para produzir software de qualidade
- Como em um supermercado...
 - Em função do problema, se escolhe o processo, os métodos e as ferramentas
- Cuidado
 - Menos do que o necessário pode levar a desordem
 - Mais do que o necessário pode emperrar projeto



Custos e desafios da ESW

■ CUSTOS

- Aproximadamente 60% dos custos de software são de desenvolvimento;
- 40% são custos de testes. Para software customizado, os custos de evolução freqüentemente superam os custos de desenvolvimento. (SOMMERVILLE, 2011:4).

■ DESAFIOS

- Lidar com o aumento de diversidade, demandas pela diminuição do tempo para a entrega e desenvolvimento de software confiável. (SOMMERVILLE, 2011:4).

Principais atributos de um bom software

■ MANUTENIBILIDADE

- O software deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes. Esse é um atributo crítico, porque a mudança de software é um requisito inevitável de um ambiente de negócio em mudança.

■ CONFIANÇA E PROTEÇÃO

- [...] Um software confiável não deve causar prejuízos físicos ou econômicos nos casos de falha de sistema. Usuários maliciosos não devem ser capazes de acessar ou prejudicar o sistema.

Principais atributos de um bom software

■ EFICIÊNCIA

- O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, eficiência inclui capacidade de resposta, tempo de processamento, uso de memória etc.

■ ACEITABILIDADE

- O software deve ser aceitável para o tipo de usuário para o qual foi projetado. Isso significa que deve ser compreensível, usável e compatível com outros sistemas usados por ele.

Crise do Software

- Nas décadas de 1960/1970 as técnicas e os processos da Engenharia de Software atual eram inexistentes. Esse cenário de desenvolvimento de software se caracterizava principalmente pela forma ad-hoc de desenvolvimento.
- Fatores que contribuíram para a crise:
 - Limitações de hardware.
 - Inexistência de uma metodologia de desenvolvimento.
 - Inexistência de documentação do projeto.
 - Falta de gerenciamento.
 - Inexistência de formação dos membros da equipe

A Crise do Software

■ Problemas resultantes da crise:

- Estouro de prazos dos projetos.
- Estouro no orçamento dos projetos.
- O software, na maioria das vezes, não satisfaz os requisitos dos usuários.
- Entrega de software com pouca qualidade.
- Dificuldades de manutenção.
- Resultado: O surgimento da “ENGENHARIA DE SOFTWARE”

Engenharia de Software

■ Atividade:

- No Quadro a seguir há 10 palavras que fazem referência aos nossos conteúdos.
- Encontre-as e discuta com seus colegas o significado de cada uma delas dentro do contexto de nossa UC.
(tempo 15 minutos)

Engenharia de Software

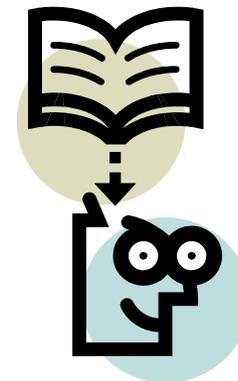
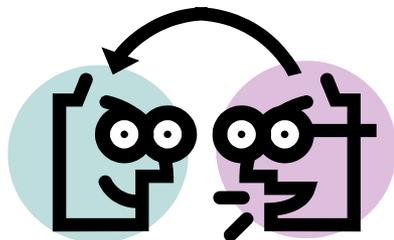
■ Atividade?

A	R	Q	U	I	T	E	T	U	R	A
Q	C	O	N	F	I	A	N	Ç	A	J
S	M	R	K	Z	E	S	C	O	P	O
Z	O	V	E	S	X	U	T	K	N	P
L	D	M	V	U	S	I	D	B	K	R
T	E	S	T	E	S	O	D	P	Q	O
Z	L	A	K	I	H	O	R	R	G	T
O	A	Q	U	A	L	I	D	A	D	E
U	G	Q	I	T	T	M	M	Z	Y	Ç
L	E	V	J	C	M	H	D	O	G	Ã
R	M	A	P	N	S	D	T	S	H	O

Processos implícitos x explícitos

Lembrem-se: **Processos** sempre existem, seja de forma **implícita** ou **explícita**!

- Processos **implícitos** são difíceis de serem seguidos, em especial por novatos
- Processos **explícitos** estabelecem as regras de uma forma clara



Processos de qualidade

Última palavra para medir a **qualidade** de um processo: **Satisfação do Cliente**

Outros **indicadores** importantes:

- Funcionalidade dos produtos gerados
- Custo real do projeto
- Duração real do projeto



Modelos de ciclo de vida

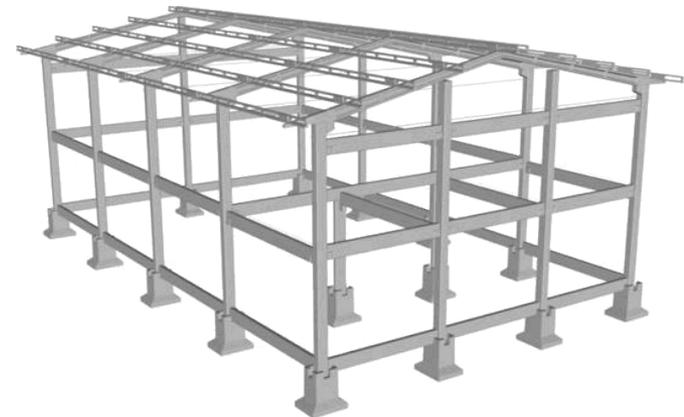
Existem alguns processos **pré-fabricados**

■ Esses processos são conhecidos como **modelos de ciclo de vida**

■ Esses processos apresentam características **predefinidas**

Devem ser adaptados para o contexto real de uso

- Características do **projeto**
- Características da **equipe**
- Características do **cliente**



Processo de desenvolvimento de software

- Todo processo deve possuir uma **arquitetura** bem definida e os **passos iniciais** para definir essa arquitetura é o modelo do ciclo de vida.
- Processo de Software trata-se de um **conjunto de atividades, ordenadas**, eventualmente com fases/etapas, papéis e marcos definidos com a finalidade de obter um produto (um sistema, um programa de computador, um serviço) com o objetivo de obter qualidade, entregas nos tempos determinados e dentro do orçamento.

Exemplos de Processos/Frameworks de Desenvolvimento de Software

UP : Unified **P**rocess.

EUP : Enterprise **U**nified **P**rocess.

PSP : **P**ersonal **S**oftware **P**rocess.

TSP : Team **S**oftware **P**rocess.

PRAXIS : **P**Rocesso para **A**plicativos e**X**tensíveis **I**nterativos.

RUP : Rational **U**nified **P**rocess.

ICONIX : **ICONIX** Software Engeneering.

XP : e**X**treme **P**rogramming.

SCRUM : Framework de gestão de projetos, desenvolvimento ágil.

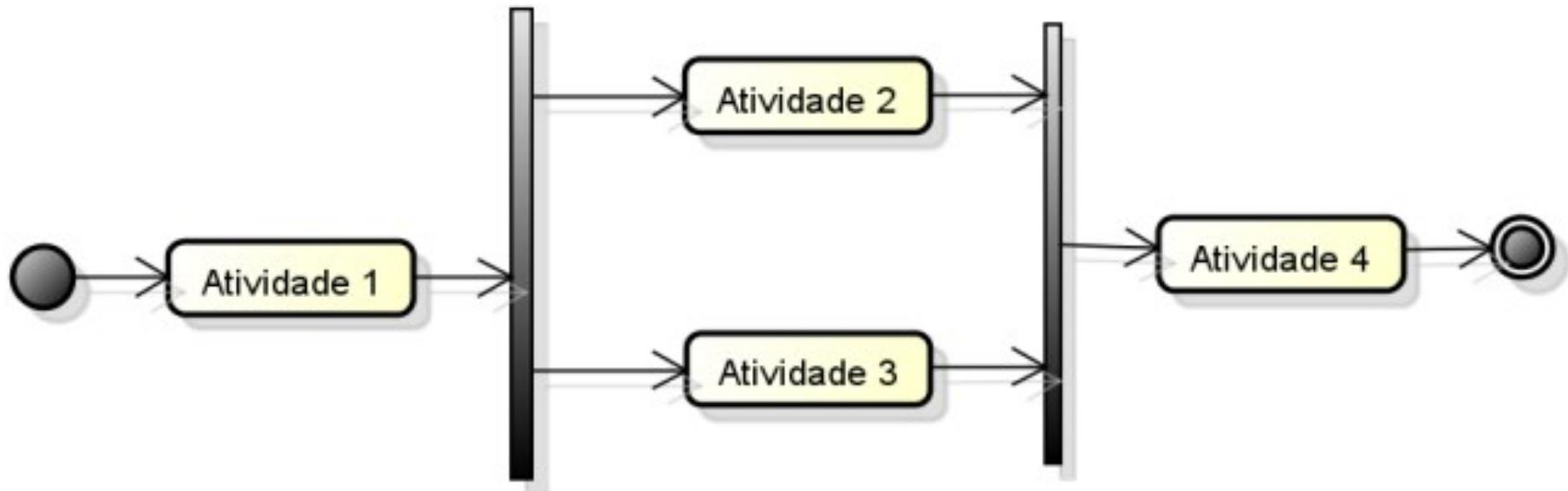
KANBAN : Sistema de controle e gestão do fluxo de produção (Kanban).



Engenharia de Software

PROCESSOS/FRAMEWORKS DE DESENVOLVIMENTO DE SOFTWARE

- Podem existir passos em paralelos em um processo.



O que é uma “Norma”?



- As normas tem por **objetivo** regular os procedimentos, atos, atividades, regras e padrões.
- As empresas aplicam as normas de acordo com a sua **interpretação**.
- As normas são usadas para **padronizar** processos visando a qualidade do resultado deste, o produto ou serviço.

Engenharia de Software

A norma que estabelece uma estrutura comum para os processo de ciclo de vida de software é a ISO/IEC 12207.

ALGUMAS CARACTERÍSTICAS:

- Estabelece uma arquitetura de alto nível do ciclo de vida de software;
- Descreve um conjunto de processos e seus inter-relacionamentos;
- Não possui nenhuma ligação com métodos, ferramentas, treinamentos, métricas ou tecnologias empregadas (independência de cultura organizacional);
- Flexibilidade, define "o que fazer" e não "como fazer";
- Modularidade são fortemente coesos e fracamente acoplados.

Engenharia de Software

Outras Normas:

- NBR ISO 9001: Garantia de qualidade de software,
- NBR ISO 900-3: Qualidade, aplicação da iso 9000 para o processo de desenvolvimento de software;
- NBR ISO 10011: Auditoria e Qualidade,
- ISO 9126: Associado às características da qualidade de produtos de software,
- ISO 14598: Relacionado a avaliação de produtos de software, baseados na ISO 9126
- ISO 12119: Relacionado aos softwares de prateleiras.

Modelos de Ciclo de vida de desenvolvimento de software

- Segundo BEZERRA (2007), o desenvolvimento de software é uma atividade complexa. Essa complexidade corresponde à sobreposição das complexidades relativas ao desenvolvimento dos seus diversos componentes: software, hardware, procedimentos etc. Isto reflete no alto número de projetos de software que não chegam ao final, ou que extrapolam recursos de tempo e dinheiro alocados. (BEZERRA, 2007:21)

Problemas ESW...

Grande número dos projetos não chegam ao final ou exigem recursos adicionais. Levantamento da Chaos Report (1994). Fonte: <http://standishgroup.com>

- Somente 10% dos projetos terminam dentro do prazo.
- 25% dos projetos são descontinuados antes de chegarem ao final.
- 60% dos projetos atingem custo acima do esperado.
- O projetos atrasam em média 1 ano.



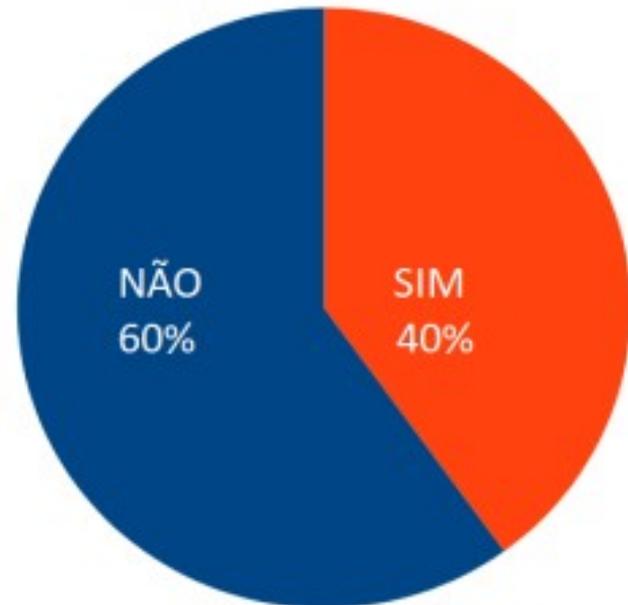
Porque usar modelos de processos de desenvolvimento de software?

NO ORÇAMENTO



Porcentagem de projetos dentro do orçamento. Dados CHAOS Report 2015.

DENTRO DO PRAZO



Porcentagem de projetos entregues dentro do prazo. Dados CHAOS Report 2015.

Porque usar modelos de processos de desenvolvimento de software?

RESOLUÇÃO TRADICIONAL PARA TODOS OS PROJETOS

	2011	2012	2013	2014	2015
Bem-sucedidos	39%	37%	41%	36%	36%
Fracassados	22%	17%	19%	17%	19%

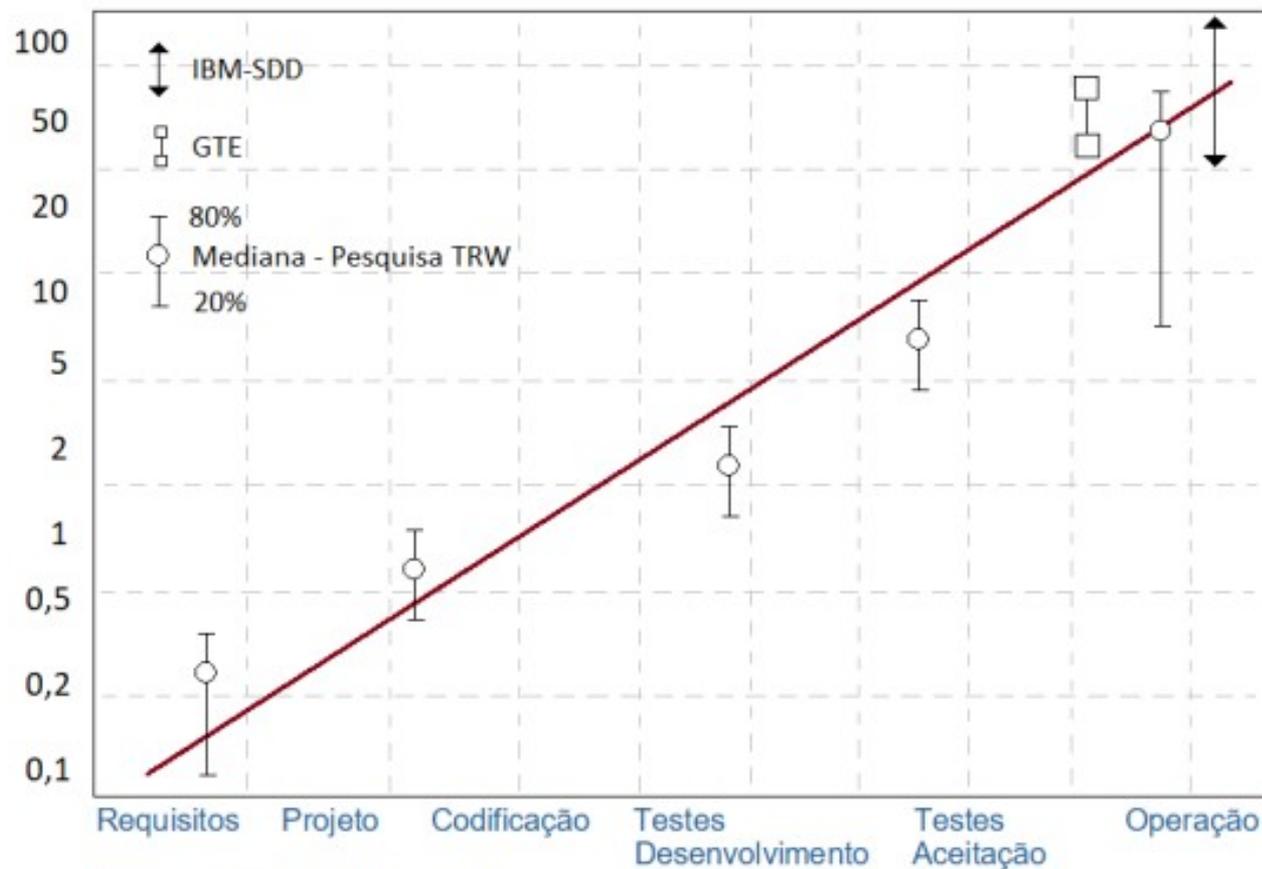
Resolução tradicional de todos os projetos de software. Fonte CHAOS REPORT 2015.

Porque usar modelos de processos de desenvolvimento de software?

- Acompanhar o desenvolvimento de software,
- Gerenciar requisitos
- Usar uma arquitetura adequada de desenvolvimento,
- Usar modelos para desenvolver software,
- Controlar continuamente a qualidade do software,
- Acompanhar as mudanças.

Engenharia de Software

Custo relativo para corrigir um erro durante o desenvolvimento do sistema



(Fonte: Gane, 1983, Página 7)

Problemas ESW...

A importância dos modelos e da documentação do projeto: redução de custos e previsões futuras. O modelo da **Casa de Cachorro**.
BEZERRA (2003).

Problemas ESW...

Um características intrínseca de sistemas de software é a complexidade de seu desenvolvimento, que cresce à medida que cresce o tamanho do sistema.



Curiosamente, muitas empresas de desenvolvimento de SW começam querendo construir prédios altos como se estivessem fazendo uma casinha de cachorro.

Engenharia de Software

PROBLEMAS . . .

Desavença entre membros do projeto de TI e os clientes giram em torno dos requisitos. PENDER (2004).

Um estudo sobre litígio em cima de software realizado por *Capers Jones da Software Producty Research*, quase sem exceção o litígio de software é baseado na discussão a respeito de requisitos.

Os clientes afirmam que os seus requisitos não foram satisfeitos...

O pessoal da TI diz que os requisitos dos clientes ficam mudando a todo momento ...



Como o cliente explicou...



Como o líder de projeto entendeu...



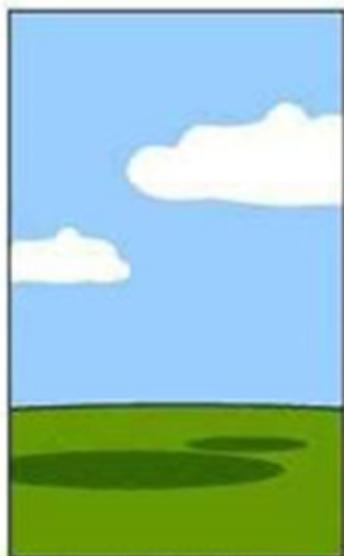
Como o analista projetou...



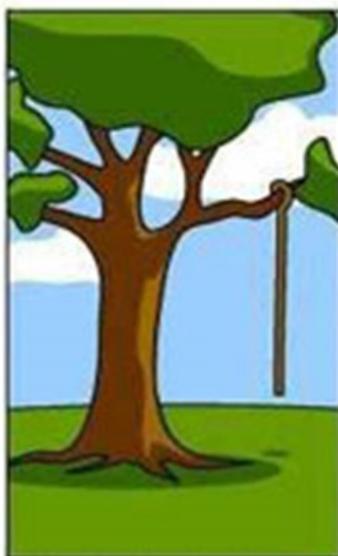
Como o programador construiu...



Como o Consultor de Negócios descreveu...



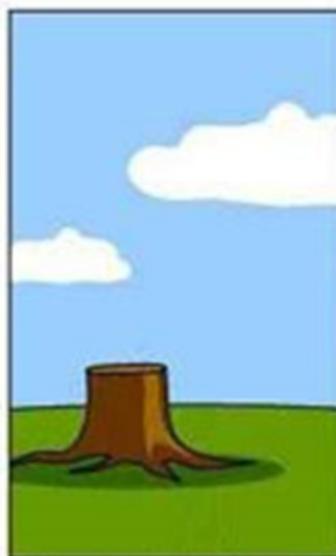
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

Problemas ESW...

Um estudo baseado em 6.700 sistemas, mostrou que os custos resultantes da má realização da etapa de levantamento de requisitos, podem ser duzentas vezes maiores que o realmente necessário (JONES apud BEZERRA, 2002, p. 22).

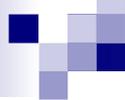
Atividade

- Responda o questionário a seguir.
- A turma será dividida em grupos para responder às questões.
- Posteriormente será sorteado um grupo para responder uma das questões.
- Tempo estimado para resolver o questionário 45 minutos.
- O grupo deve:
 - Eleger um integrante para controlar o tempo da atividade;
 - Eleger um integrante para registrar as informações;
 - Eleger um integrante para apresentar o resultado da pesquisa.

Engenharia de Software

QUESTÕES

1. Citar os principais fatores que contribuíram para a crise do software.
2. Citar 3 (três) problemas resultantes da crise do software.
3. Conceituar Engenharia de Software.
4. Diferenciar Ciência da Computação x Engenharia de software.
5. Diferenciar Engenharia de Sistemas x Engenharia de software.
6. Quais são os principais desafios da Engenharia de Software?
7. Quais são os principais atributos de um bom software?



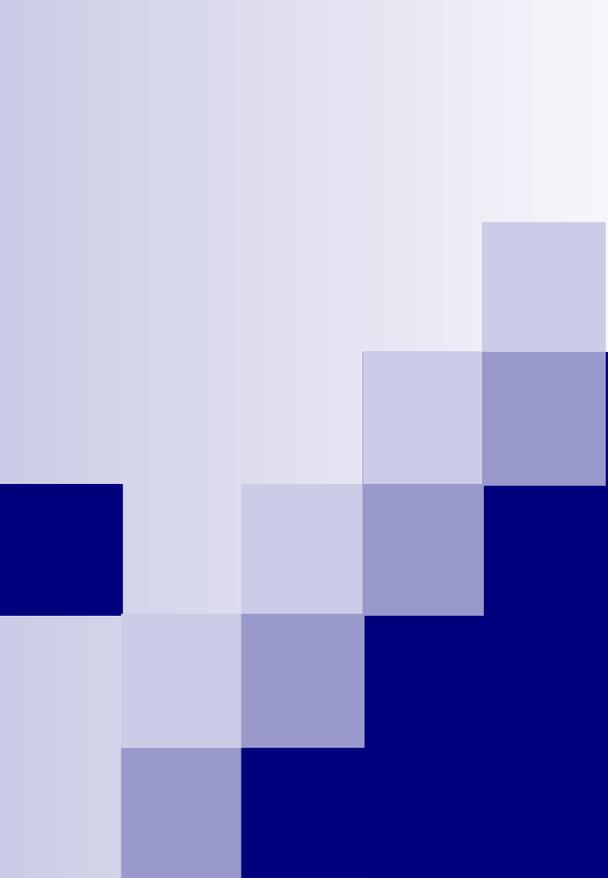
Conclusão

Alguns conceitos são compartilhados com outras áreas da engenharia.

Estes conceitos são fundamentais para compreender a importância da ESW e seus processos.

Referências

- PRESSMAN, Roger; MAXIM, Bruce. Engenharia de software: uma abordagem profissional. 8.ed. Bookman, 2016. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788580555349>
- SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. E-book. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/2613/epub/0>
- LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e desenvolvimento iterativo. 3. ed Porto Alegre: Bookman, 2007. E-book. Disponível em: <https://integrada.minhabiblioteca.com.br/books/9788577800476>



Fim