



Universidade Estadual de Maringá
Departamento de Informática



Introdução à Engenharia de Software

Material baseado no original do Profº Renato Balancieri (DIN/UEM)

Prof.^a Juliana Keiko Yamaguchi
janeiro de 2019

Objetivos

- Conceitos básicos da Engenharia de Software
- Características do software
- Tipos de aplicações de software
- Evolução do Software
- Importância da Engenharia de Software

Contextualização

- O que é software?
- O que é Engenharia de Software?
- Para que serve a Engenharia de Software?

Conceitos básicos

O que é software?

- Definição por Pressman (1995)
 - conjunto de instruções (programas de computador) que, quando executadas, produzem a função e desempenho desejados;
 - estruturas de dados que possibilitam que os programas manipulem adequadamente a informação;
 - documentos que descrevem a operação e o uso dos programas.

Conceitos básicos

O que é software?

- Definição por Sommerville (2001)
 - Programas de computadores e respectivas documentações e dados de configuração necessários para que funcionem corretamente.
 - Softwares podem ser desenvolvidos para um cliente particular ou para um domínio de mercado.

Características do software

- O software é desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico (Pressman, 1995)



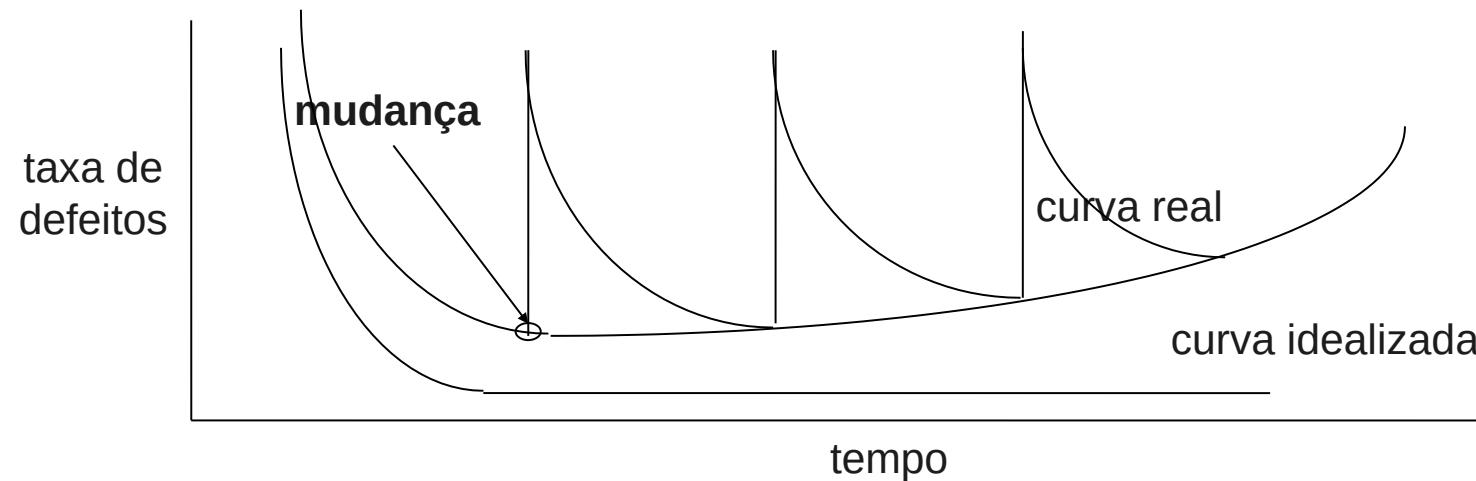
<https://www.youtube.com/watch?v=MX0V0cHUeNM>

Características do software

- “Fábrica de Software”
 - É importante observar que esse termo não implica que a manufatura de hardware e o desenvolvimento de software sejam equivalentes
 - Este conceito recomenda o uso de ferramentas automatizadas (CASE – *Computer-Aided Software Engineering*) para o desenvolvimento (Pressman, 1995)

Características do software

- Software não se desgasta, mas pode tornar-se obsoleto.



Curva de falhas para o software (Pressman, 2011; p.33)

Aplicações do software

- Categorizar o software torna-se muito difícil à medida em que sua complexidade cresce.
 - Software básico
 - Software de tempo real
 - Software comercial
 - Software científico e de engenharia
 - Software embutido
 - Software de computador pessoal
 - Software de inteligência artificial
 - Software para Web

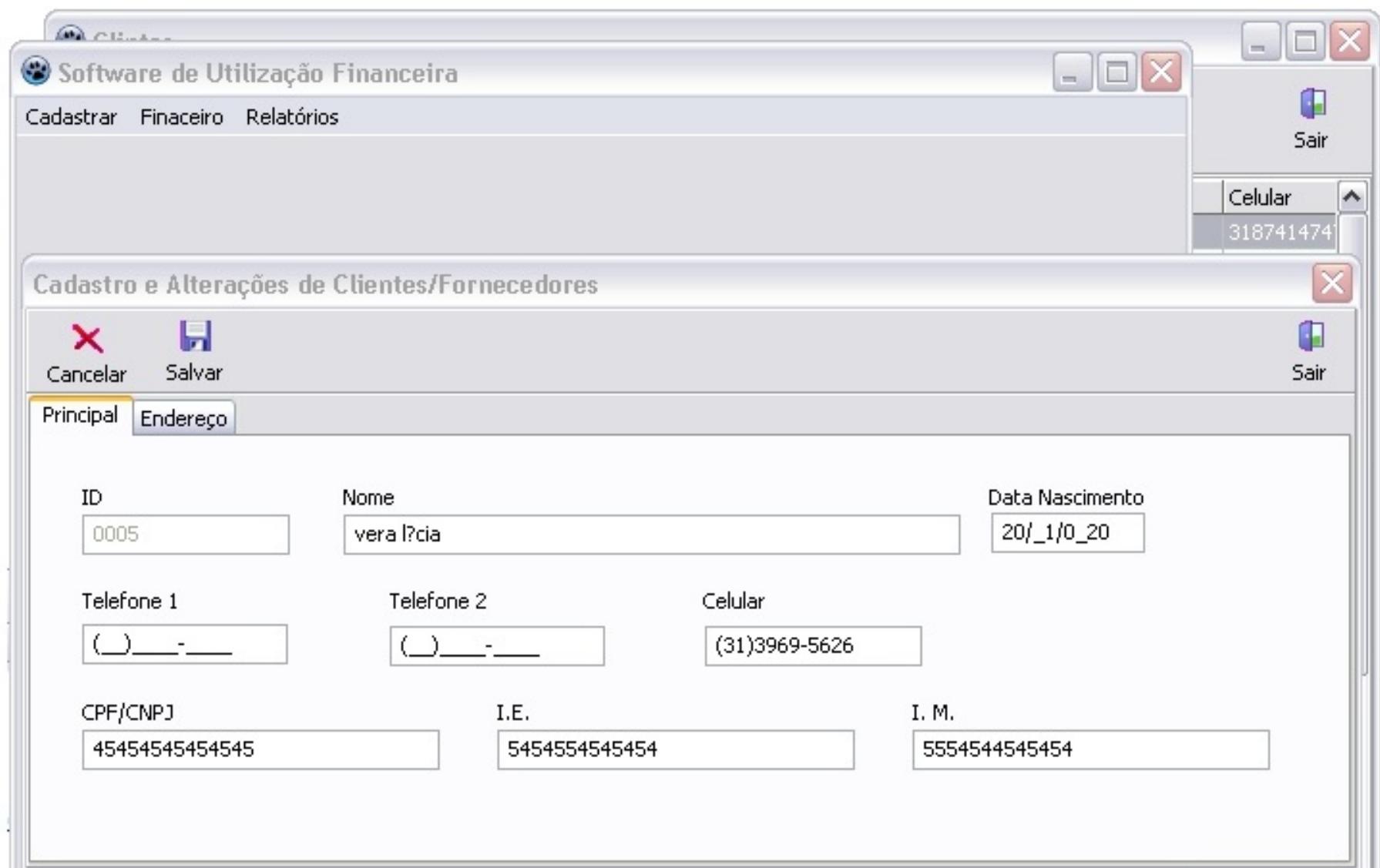
Aplicações do software

- **Software básico:** programas escritos para dar apoio a outros programas
 - Exemplos: compiladores, utilitários de gerenciamento de arquivos (Windows Explorer, Nautilus do Ubuntu), sistema operacional
- **Software de tempo real:** software que monitora, analisa e controla eventos do mundo real.
 - Exemplos: sistema de monitoramento de espaço aéreo, controle de processos industriais, videoconferência.

Aplicações do software

- **Software comercial:** sistemas de operações comerciais e tomadas de decisões administrativas.
 - Exemplos: sistemas de gestão empresarial, sistemas de vendas e controle de estoque, etc.

Aplicações do software Software comercial

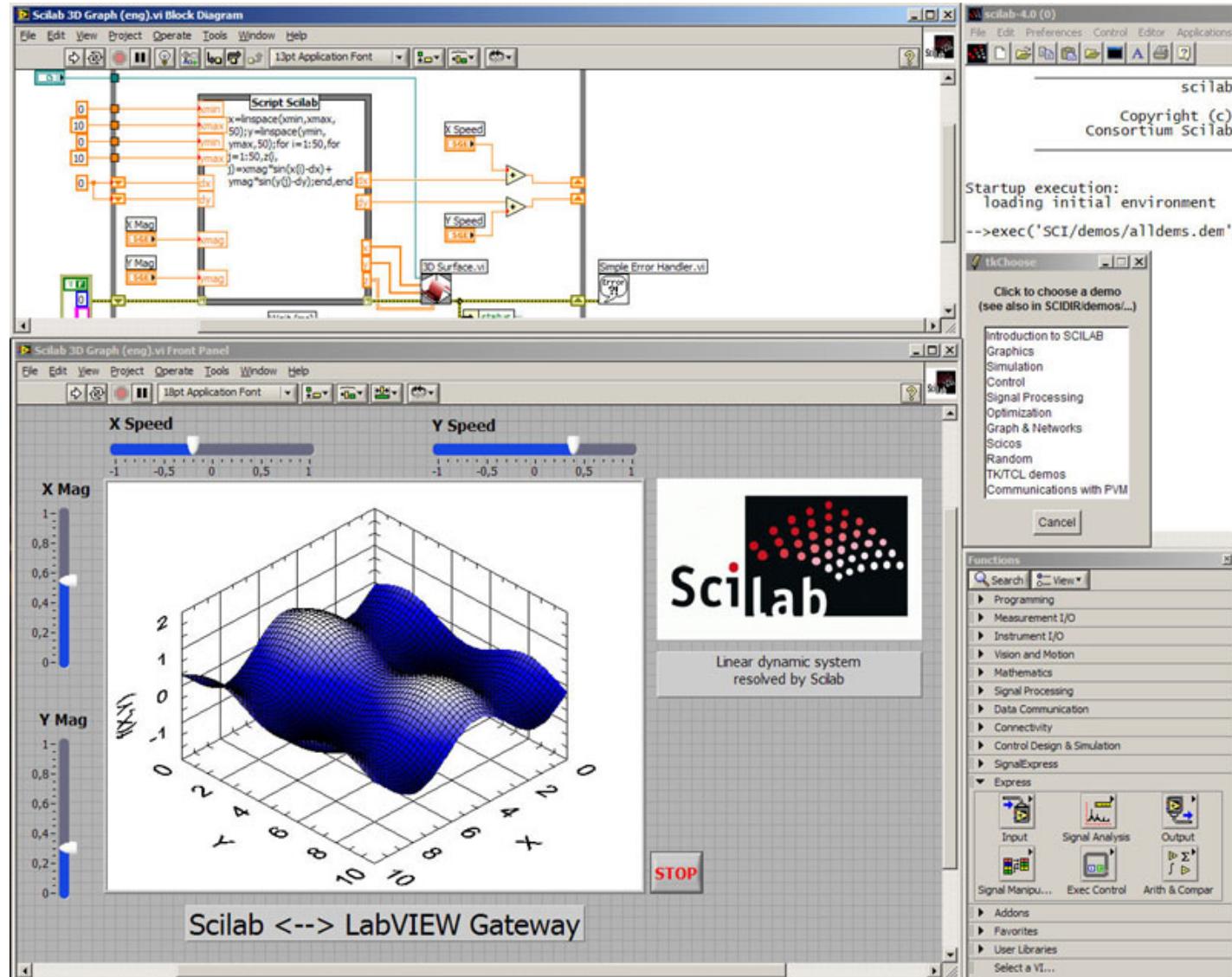


<http://www.vivaolinux.com.br/artigo/E-possivel-usar-o-Lazarus-em-alternativa-ao-Delphi-para-desenvolver-aplicacoes-comerciais?pagina=4>

Aplicações do software

- **Software científico e de engenharia:** caracterizado por algoritmos de processamento de números.
 - Exemplos: CAD (*Computer-aided Design*), softwares para astronomia, biologia molecular, etc.
 - Exemplos de softwares:
 - Scilab <<http://www.scilab.org/>>
 - GNU Octave <<http://www.gnu.org/software/octave/>>

Aplicações do software Software científico e de engenharia



Aplicações do software

- **Software embutido (*embedded software*):** usado para controlar produtos e sistemas para mercados industriais e de consumo. Geralmente possui funções limitadas e particulares.
 - Exemplos: sistemas para microondas, automóveis, impressoras, urna eletrônica, MP3/MP4 *players*, *smartphones*, etc.

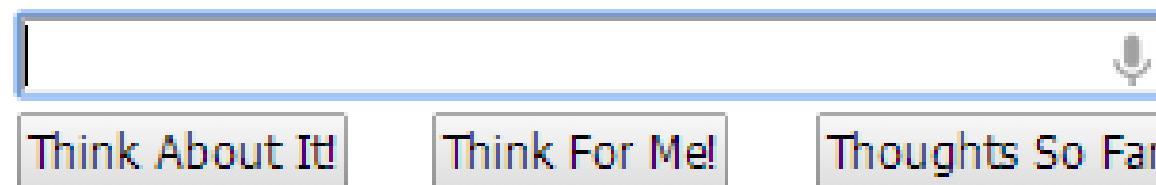
Aplicações do software

- **Software de computador pessoal:** Programas voltados para a utilização direta do usuário.
 - Exemplos: processadores de texto, planilhas eletrônicas, gerenciamento de dados, dentre outros.

Aplicações do software

- **Software de inteligência artificial:** faz uso de algoritmos não numéricos para resolver problemas que não sejam favoráveis à computação ou à análise direta.
 - Exemplos: Supercomputadores que jogam xadrez (*DeepBlue*), sistemas de reconhecimento de voz, sistemas de conversação (atendente virtual ou *Chatter Bots* – *Robô Ed da Petrobrás*), sistemas especialistas (baseados em conhecimento), redes neurais artificiais

Aplicações do software Software de inteligência artificial



Aplicações do software Software de inteligência artificial



**robô
ED**

[| INDIQUE PARA
UM AMIGO](#)

[| NOVIDADES DO ED
POR E-MAIL](#)

Redes Sociais

Você já conhece meus perfis oficiais nas redes sociais?

conpet 

CONVERSE COMIGO **MEU BLOG** **EVENTOS**
QUEM SOU EU **VENHA APRENDER** **IMPRENSA**
DIVERSÕES **CONHEÇA O CONPET** **LINKS**

Não use a temperatura máxima de seu aquecedor sem necessidade.

Bem vindo ao meu site!

Este espaço foi criado para que você possa me conhecer melhor, saber mais sobre meio ambiente, recursos naturais, petróleo, e muito mais. Aqui você vai aprender e ao mesmo tempo se divertir.

Meu site está cheio de informações e novidades. Espero que você aproveite bastante.

Um grande abraço,
Ed


MINHA MISSÃO


CONVERSE COMIGO



Aplicações do software Software de inteligência artificial



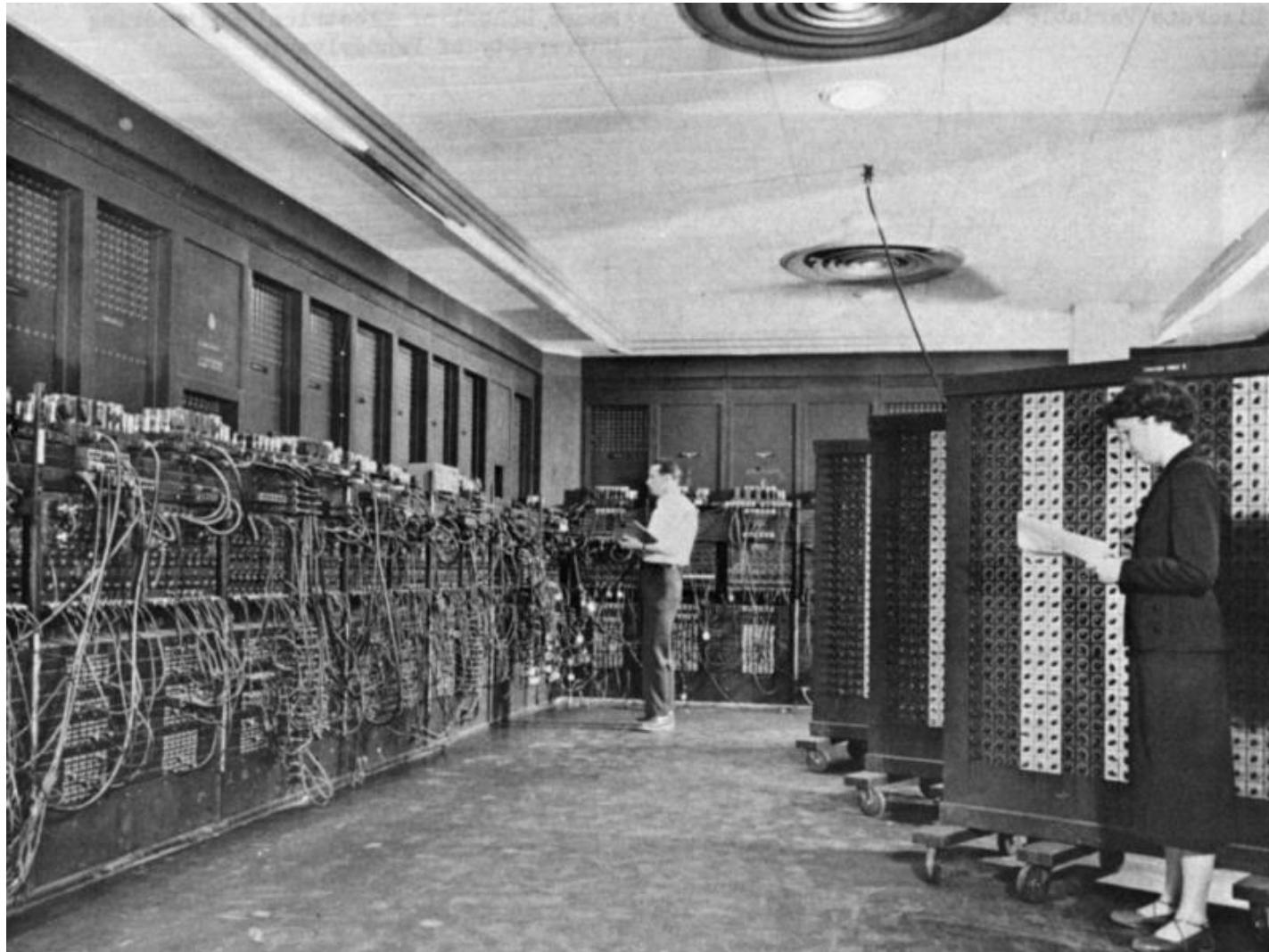
<https://noticias.bol.uol.com.br/ultimas-noticias/tecnologia/2018/04/24/simsimi-app-de-conversa-com-inteligencia-artificial-e-risco-para-criancas.htm>

Aplicações do software

- **Software para Web:** softwares que podem ser acessados pela rede.
 - Em geral, apresentam conteúdo dinâmico por meio de linguagens de *scripting* (JavaScript)

Evolução do computador

1^a geração (1946 a 1958)



ENIAC (1946)

Considerado o primeiro computador digital eletrônico

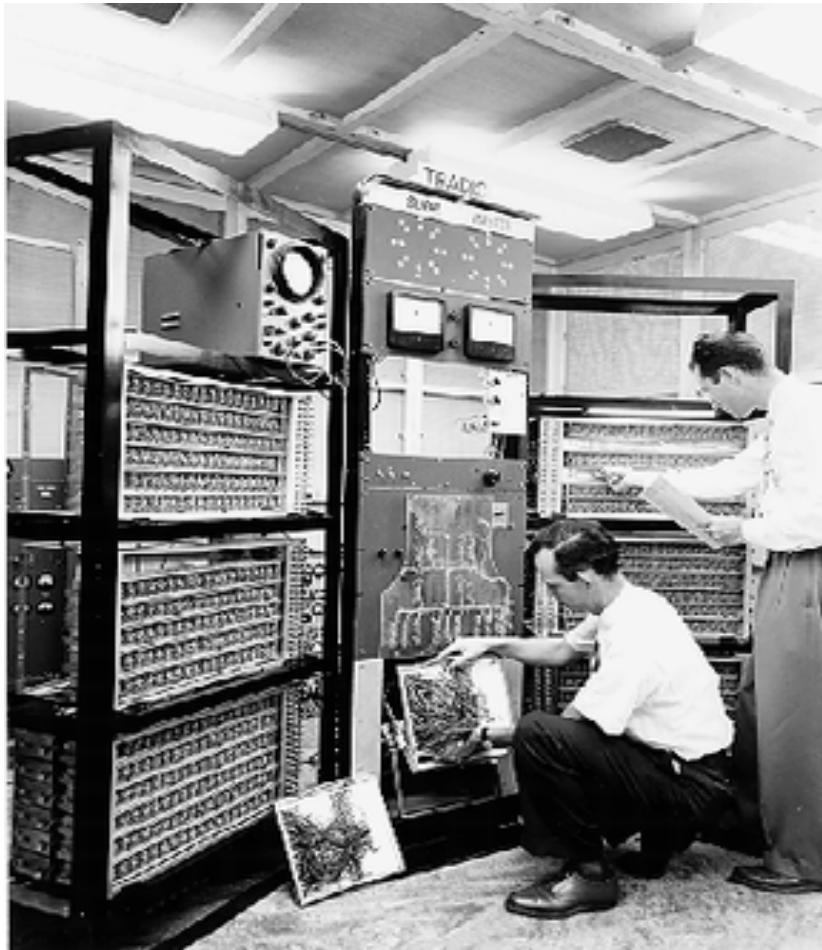
Evolução do computador

1^a geração (1946 a 1958)

Hardware	Software	Linguagem
Computadores baseados em tecnologia de válvula	As instruções eram introduzidas manualmente	Linguagem de máquina (cada válvula funcionava como um interruptor de luz, que ligava e desligava, soltando pequenas cargas elétricas. Se a válvula ligasse, o computador entendia 1 (um); se desligasse, era 0 (zero)).

Evolução do computador

2^a geração (1959 a 1963)



TRADIC
(*Transistor Digital Computer – 1955*)
Construído pela Bell Laboratories)



IBM 1401 (1959)

Evolução do computador

2^a geração (1959 a 1963)

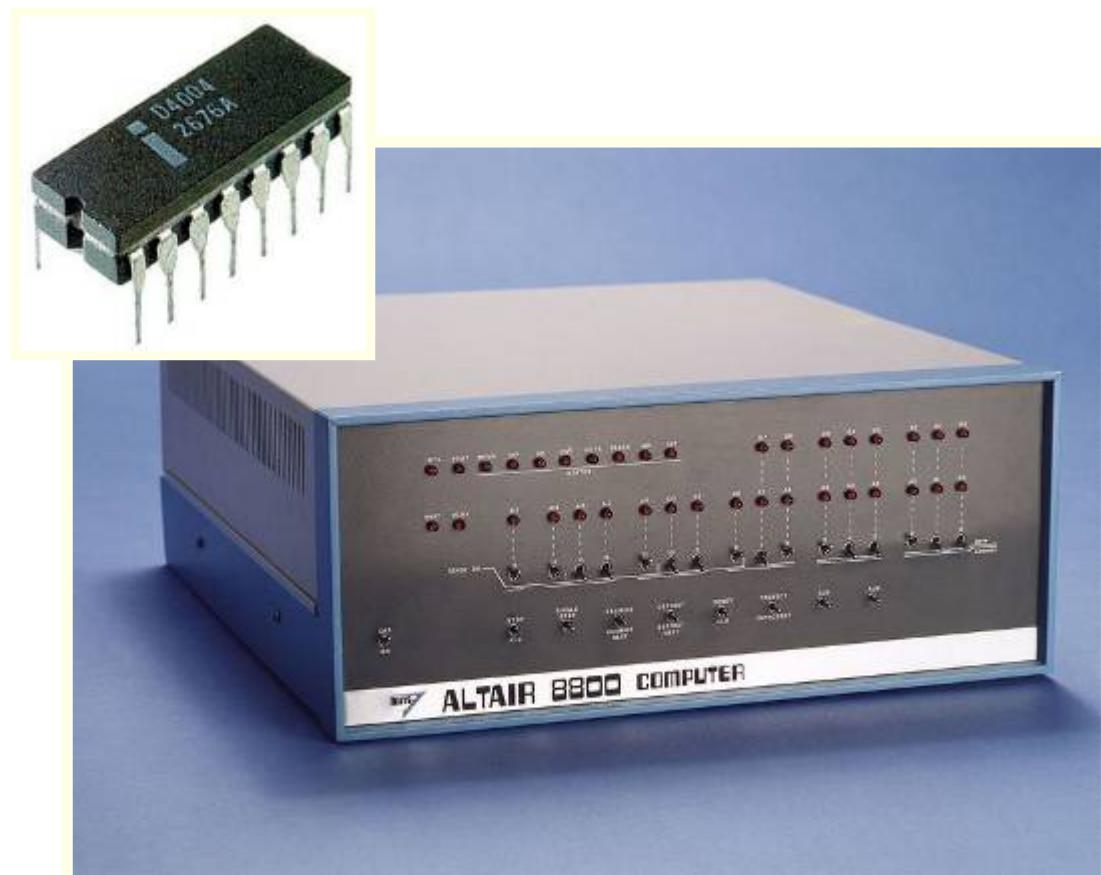
Hardware	Software	Linguagem
Computadores baseados em tecnologia de transistores (mais eficientes e confiáveis do que as válvulas)	Não havia métodos de construção de software	Linguagem de montagem (Assembly)

Evolução do computador

3^a geração (1964 a 1975)



SYSTEM / 360 (1964)
Primeiro computador de uso comercial
– 400 Kg, 7MB em disco, 128 Kb RAM



Chip Intel 4004, primeiro
microprocessador, e o computador
ALTAIR 8800, primeiro computador
pessoal (1971)

Evolução do computador

3^a geração (1964 a 1975)

Hardware	Software	Linguagem
Computadores baseados em tecnologia circuitos integrados	<ul style="list-style-type: none">▪ Surgimento da multiprogramação e sistemas multisuários▪ <i>Software houses</i> – bibliotecas de software (ampla distribuição num mercado interdisciplinar)▪ Cresce n.^o de sistemas baseado em computador▪ Manutenção quase impossível▪ Crise do software...	FORTRAN (1954), COBOL (1959), ALGOL (1958), Lisp (1958), Pascal (1971), C (1974)

Evolução do computador

4^a geração (1976 a 1995)



Apple II (1977)
Primeiro computador doméstico
de sucesso



Macintosh (1985)

Evolução do computador

4^a geração (1976 a 1995)

Hardware	Software	Linguagem
Tecnologias de micro-processadores LSI, VLSI	<ul style="list-style-type: none">■ Sistemas distribuídos■ Redes locais e globais■ Uso generalizado de microprocessadores (produtos inteligentes)■ Hardware de baixo custo	Linguagens OO: Smalltalk (1981), Python (1991), Java (1995)

Evolução do computador

5^a geração (1995 até hoje)



Em 60 anos, migramos do ENIAC para os *Laptops* e *Smartphones*

Evolução do computador

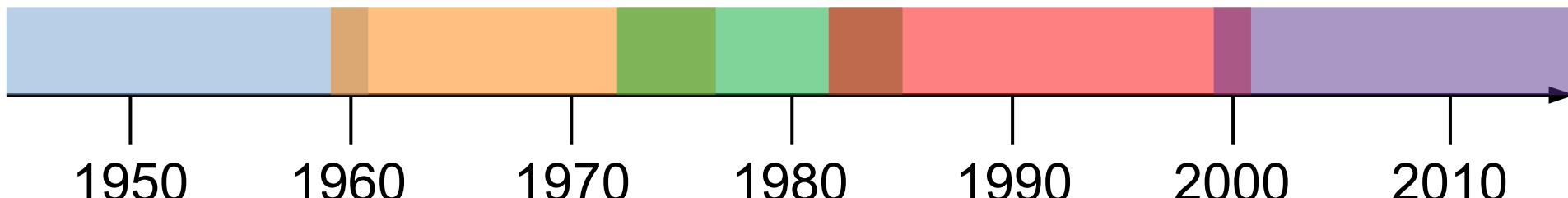
5^a geração (1995 até hoje)

Hardware	Software	Linguagem
<ul style="list-style-type: none">▪ Tecnologias de micro-processadores ULSI▪ Processamento paralelo▪ Computação quântica	<ul style="list-style-type: none">▪ Sistemas especialistas▪ Sistemas multimídia (combinação de textos, gráficos, imagens e sons)▪ Banco de dados distribuídos▪ Redes neurais	<ul style="list-style-type: none">▪ Linguagens de 4^a geração (4GL)▪ Linguagens de alto nível de abstração▪ Geradores de relatório

Obs.: As eras do hardware, software e linguagem apresentadas neste material não são rigorosamente precisas e possuem abordagens diferentes na literatura.

Evolução do computador

Primeira era	Terceira era	Quinta era
<ul style="list-style-type: none">• Sistemas batch (lote)• Distribuição limitada• Software personalizado	<ul style="list-style-type: none">• Sistemas distribuídos• Incorporação de inteligência• Hardware de baixo custo• <i>Workstations</i>• Impacto do consumidor	<ul style="list-style-type: none">• Desenvolvimento Web• Computação em nuvem• Desenvolvimento orientado a serviços
Segunda era	Quarta era	
<ul style="list-style-type: none">• Sistemas multiusuários• Sistemas de tempo real• Sistemas de banco de dados• <i>Software houses</i>	<ul style="list-style-type: none">• Sistemas desktop poderosos• Tecnologia de orientação a objetos• Sistemas especialistas• Redes neurais• Computação paralela	



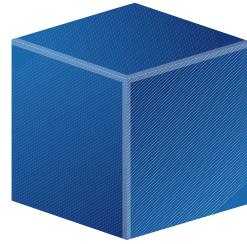
Problemas que persistem Apesar da evolução do software

- A habilidade em construir software deixa a desejar em relação ao potencial do hardware.
- A construção de software não é rápida o suficiente para atender as necessidades do mercado.
- A sociedade depende cada vez mais de software confiável; quando ele falha, podem ocorrer gastos enormes e desgaste de muitos profissionais para arrumá-lo.
- O esforço para construir software confiável e de qualidade é muito grande.
- O suporte aos programas existentes é apoiado por projetos pobres e recursos inadequados.

Como produzir software?



Mundo real



Modelo do
mundo real



Sistema de
Software

Uma perspectiva industrial

Hoje, é o software que custa mais do que o hardware.

Já há algum tempo, gerentes e técnicos se perguntam:

- Porque é preciso tanto tempo para terminar os programas?
- Porque os custos são tão altos?
- Porque não se consegue encontrar todos os erros antes que o software seja liberado para os clientes?
- Porque existe uma dificuldade em medir o progresso à medida que o software está sendo construído?

Competitividade do software

- Atualmente o software é um negócio competitivo.
- Os principais direcionadores que propiciarão uma intensa competição na área de software são: custo, adequação de prazo e qualidade.
- Intensifica-se, portanto, uma rápida movimentação dos desenvolvedores para adotar práticas modernas de Engenharia de Software.

O que é Engenharia de Software?

Definição por Sommerville (2001)

- Disciplina relacionada à todos os aspectos de produção de software, desde os estágios iniciais da especificação do sistema até a manutenção do produto entregue ao cliente.
- Envolve questões técnicas e não técnicas, tais como a especificação do conhecimento, técnicas de projeto e implementação, conhecimento dos fatores humanos pelo engenheiro de software e ainda, gestão de projetos.

O que é Engenharia de Software?

Definição por Rezende (2005)

- Visa sistematizar a produção, a manutenção, a evolução e a recuperação de produtos de software, de modo que ocorra dentro dos prazos e custos estimados, com progresso controlado e utilizando princípios, métodos, tecnologia e processos em contínuo aprimoramento, garantindo a qualidade do software.

Engenharia de Software

Importância

- Durante as três primeiras décadas da era do computador (1950, 1960, 1970), o principal desafio era desenvolver um *hardware* que reduzisse o custo de processamento e armazenagem de dados (Pressman, 1995)
- Atualmente, o principal desafio é melhorar a qualidade e reduzir o custo de soluções baseadas em computador, que são implementadas via *software*.

Engenharia de Software

Características

- Engenharia de Software está relacionada à construção de programas de grande porte.
 - Técnicas e ferramentas tradicionais de programação suportam de maneira geral a programação de pequeno porte (tipicamente feita por uma única pessoa).

Engenharia de Software

Características

- A ideia central é controlar a complexidade
 - De maneira geral os problemas são tão complexos que estes não podem ser abordados na sua totalidade
 - O problema deve ser dividido em partes relacionadas cuja complexidade individual seja adequada

Engenharia de Software

Características

- O software evolui
 - A maioria dos software modela uma parte de uma realidade
 - À medida em que esta realidade se transforma, o software deve evoluir para se adequar às transformações para não se tornar obsoleto
- A eficiência com a qual o software é desenvolvido é de vital importância
 - Quanto maior a eficiência, menor é o custo

Engenharia de Software

Características

- O desenvolvimento de software requer a cooperação regular entre as pessoas
 - Com o trabalho em paralelo de várias pessoas, torna-se essencial a distribuição das tarefas, a comunicação, as responsabilidades, etc.
- O software deve suportar de forma efetiva as necessidades de seus usuários

Engenharia de Software

Características

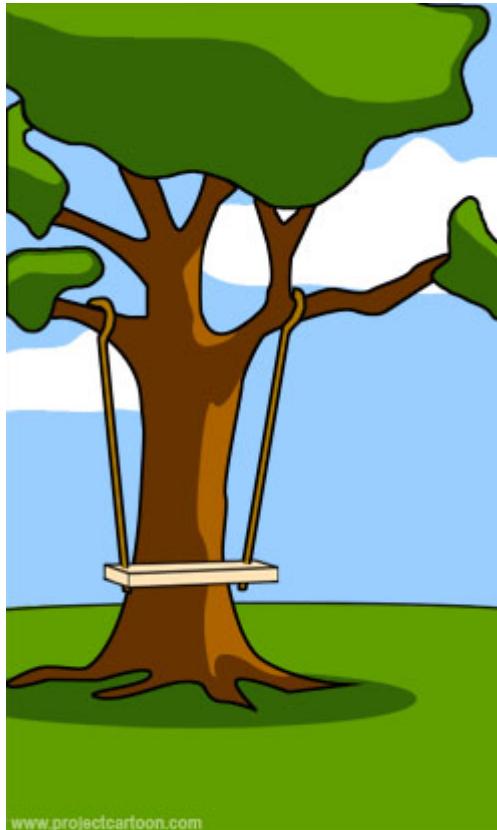
- Na Engenharia de Software, membros de uma cultura criam artefatos para membros de outras culturas
 - Os engenheiros de software normalmente não são especialistas no domínio do problema

Desenvolvimento de Software: arte ou engenharia?



Como o cliente explicou

Desenvolvimento de Software: arte ou engenharia?



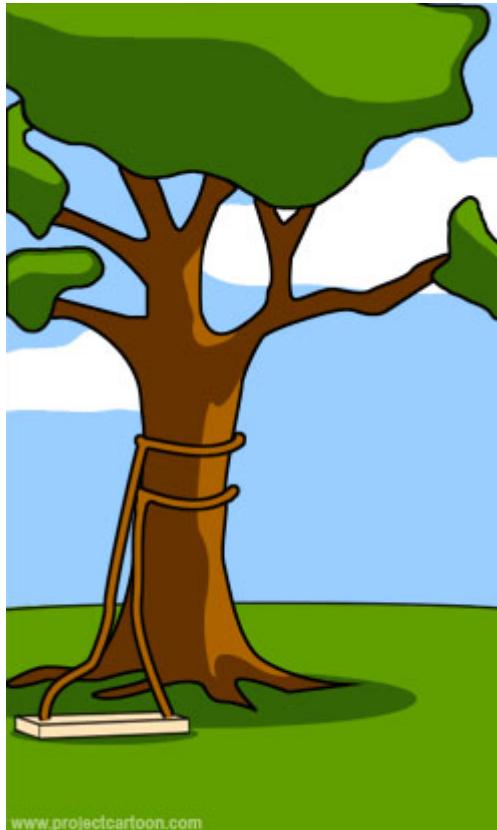
Como o líder de projeto entendeu

Desenvolvimento de Software: arte ou engenharia?



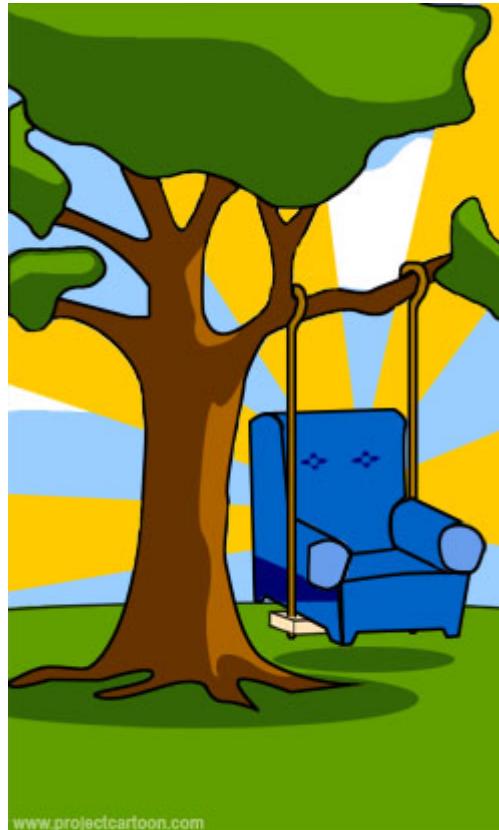
Como o analista planejou

Desenvolvimento de Software: arte ou engenharia?



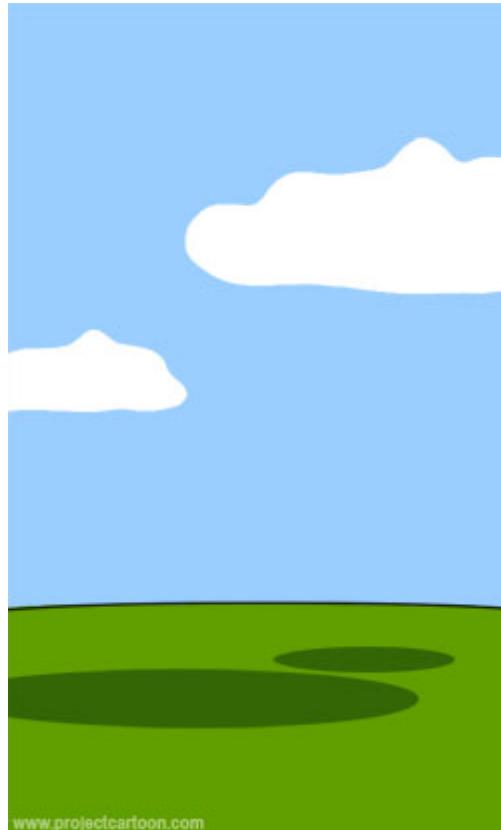
Como o programador codificou

Desenvolvimento de Software: arte ou engenharia?



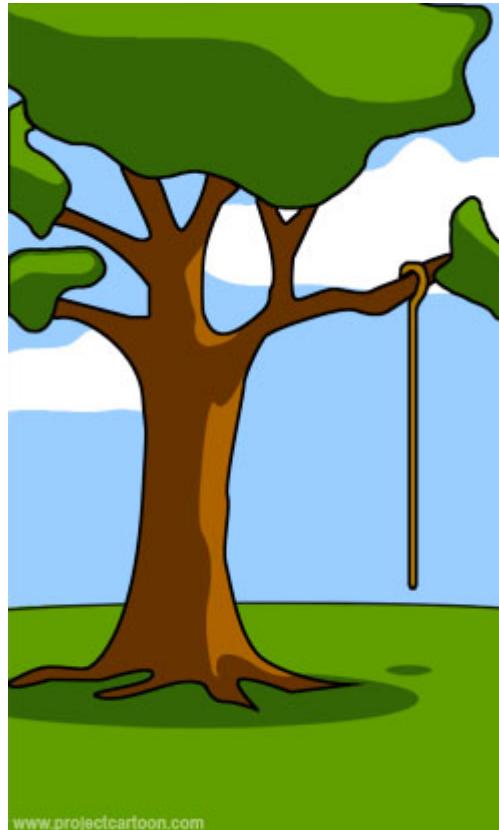
Como o consultor de negócio descreveu

Desenvolvimento de Software: arte ou engenharia?



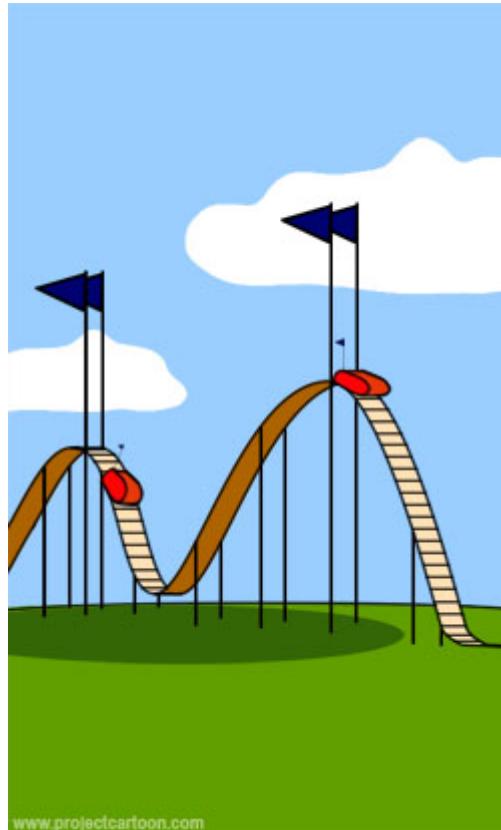
Como o projeto foi documentado

Desenvolvimento de Software: arte ou engenharia?



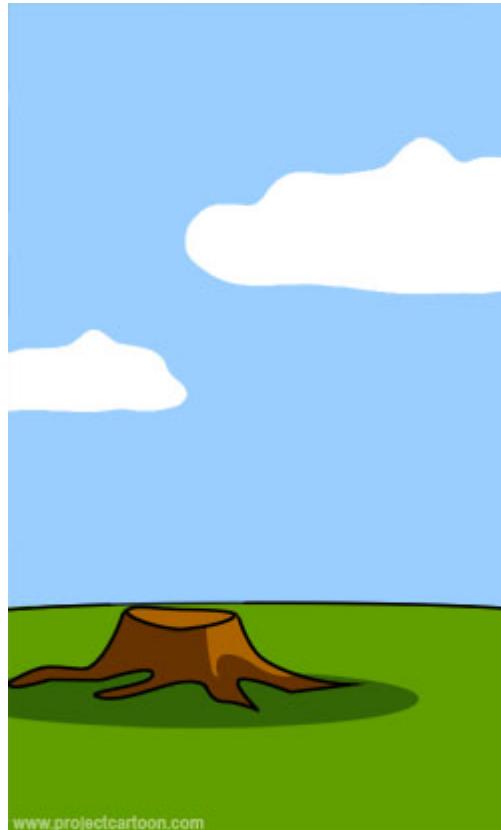
O que a assistência técnica instalou

Desenvolvimento de Software: arte ou engenharia?



Como o cliente foi cobrado

Desenvolvimento de Software: arte ou engenharia?



Como é suportado

Desenvolvimento de Software: arte ou engenharia?

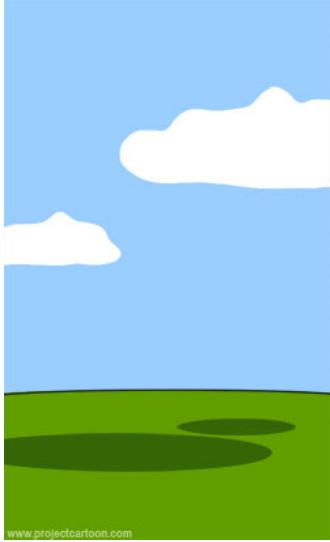


O que o cliente realmente necessitava

Desenvolvimento de Software: arte ou engenharia?



Como o cliente
explicou



Como o projeto
foi documentado



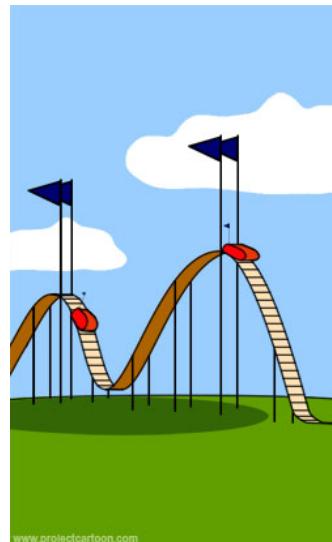
Como o líder de
projeto entendeu



O que a assistência
técnica instalou



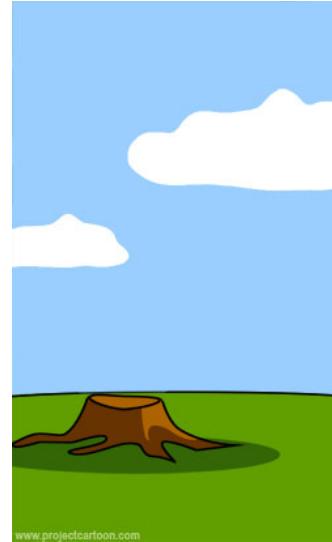
Como o analista
planejou



Como o cliente foi
cobrado



Como o programador
codificou



Como é
suportado



Como o consultor de
negócio descreveu



O que o cliente
realmente necessitava⁵⁵

Crise do software

- Refere-se a um conjunto de problemas encontrados no desenvolvimento de software.

Crise do software

Fatores

- 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 do software

Fatores

- A produtividade das pessoas da área de software não tem acompanhado a demanda por seus serviços
 - “Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente”
- 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 do software

Fatores

- 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

Crise do software

Fatores



Crise do software

Causa dos problemas

1. Próprio caráter do Software

- O software é um elemento de sistema lógico e não físico.
- Consequentemente, o sucesso é medido pela qualidade de uma única entidade e não pela qualidade de muitas entidades manufaturadas
- O software não se desgasta, mas se deteriora!!!

Crise do software

Causa dos problemas

2. Falhas das pessoas responsáveis pelo desenvolvimento de Software

- Gerentes sem nenhum *background* em software
- Os profissionais da área de software têm recebido pouco treinamento formal em novas técnicas para o desenvolvimento de software
- Resistência a mudanças

Crise do software

Causa dos problemas

3. Mitos do Software

- Propagaram desinformação e confusão
 - Administrativos
 - Cliente
 - Profissional

Mitos do software

Administrativo

Mito

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 Administrativo

Mito

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

Administrativo

Mito

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

Do cliente

Mito

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.
- É fundamental uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação.

Mitos do software

Do cliente

Mito

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

Realidade

- Uma mudança, quando solicitada tardeamente num projeto, pode ser de uma ordem de magnitude mais dispendiosa do que se a mesma fosse solicitada nas fases iniciais.

Mitos do software

Do profissional

Mito

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

Do profissional

Mito

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.

Considerações finais

- Pode-se dizer que atualmente o software é um item indispensável nas atividades humanas.
- O objetivo da Engenharia de Software é produzir um software de qualidade.
- Como atingir a qualidade?

Referências

- PRESSMAN, R. S. Engenharia de Software – Makron Books, 1995, 3.ed. São Paulo, pgs 03 – 30.
- REZENDE, D. A. Engenharia de Software e Sistema de Informação, Brasport, 2005, 3^a ed., Rio de Janeiro.
- SOMMERVILLE, I. Engenharia de Software, Adisson Wesley, 2003, 6^a.ed., São Paulo.