



USP - Universidade  
de São Paulo



IME - Instituto de  
Matemática e Estatística

## MAC0332 - Engenharia de Software



# Introdução

Prof. Marco Aurélio Gerosa

([gerosa@ime.usp.br](mailto:gerosa@ime.usp.br))

<http://www.ime.usp.br/~gerosa/>



## Diagnóstico

# Formulário de perfil da turma



# O que é Engenharia de Software?



# 1) O que é Engenharia?



# Engenharia

- Arte de aplicar os conhecimentos científicos à invenção, aperfeiçoamento ou utilização da técnica industrial em todas as suas determinações **(Dic. Michaelis)**
- Engenharia (latim ingeniu = "faculdade inventiva, talento") é a arte, a ciência e a técnica de bem conjugar os conhecimentos especializados (científicos) de uma dada área do saber com a sua viabilidade técnico-econômica, para produzir novas utilidades e/ou transformar a natureza, em conformidade com ideias bem planejadas. **(Wikipedia)**
- Arte de aplicar conhecimentos científicos e empíricos e certas habilitações específicas à criação de estruturas, dispositivos e processos que se utilizam para converter recursos naturais em formas adequadas ao atendimento das necessidades humanas. **(Dic. Aurélio)**
- Aplicação de métodos científicos ou empíricos à utilização dos recursos da natureza em benefício do ser humano **(Dic. Houaiss)**



# História da Engenharia

- **Antes da revolução industrial**
  - Practical artists and craftsmen - trial and error.
  - Tinkering + imagination -> many marvelous devices.
  - Engineer = “ingenious”
  - Leonardo da Vinci - *Ingegnere Generale*.
- **Revolução industrial**
  - Galileo - scientific approach to practical problems.
  - Traditional artisans -> professionals.
  - Mathematical analysis and controlled experiments.
  - Technical training shifted from apprenticeship to university education.
  - Meetings and journal publications. Professional societies emerged.
  - Chemical and electrical engineering developed in close collaboration with chemistry and physics

<http://www.creatingtechnology.org/history.htm>



# História da Engenharia

- **Era da informação**

- Engineering stimulated by new technologies, notably aerospace, microelectronics, computers, novel means of telecommunications
- Information revolution in which intellectual chores are increasingly alleviated by machines
- Basic sciences such as physics, chemistry, and biology continue to contribute to Engineering
- Engineering research produced not only new technologies but also bodies of powerful systematic knowledge: the engineering sciences and systems theories in information, computer, control, and communications.
- Engineering firmly established itself as a science of creating, explaining, and utilizing manmade systems.

<http://www.creatingtechnology.org/history.htm>



## 2) O que é Software?



# Software

- Em um sistema computacional, o conjunto dos componentes que não fazem parte do equipamento físico propriamente dito e que incluem as instruções e programas (e os dados a eles associados) empregados durante a utilização do sistema. (**Dic. Aurélio**)
- Software é um artefato abstrato
  - A construção de software requer inovação na engenharia, que se desenvolveu considerando eminentemente artefatos concretos
  - Qual é o custo de matéria prima no desenvolvimento de software? Como esse custo se diferencia dos custos de mão de obra?
- Projetos de software podem atingir grandes proporções:
  - A complexidade do projeto (design, construção, gerenciamento) cresce “exponencialmente” com as dimensões do projeto.
    - Desenvolvedores – pode chegar a centenas, ou pode ser um projeto aberto
    - Linhas de código – pode chegar a muitos milhões
    - Usuários – pode chegar a milhões
    - Demandas – alta confiabilidade (missão crítica), vida longa, segurança, adequação, interoperabilidade, conformidade, usabilidade, atratividade, eficiência, eficácia, manutenibilidade, portabilidade etc.



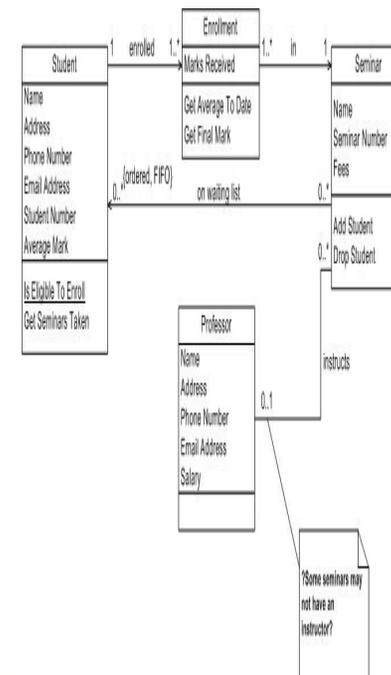
### 3) O que é Engenharia de Software?

= Desenvolvimento de software?



# Engenharia de Software

- Software engineering.** The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. [IEEE, 1991]





# O que é Engenharia de Software?

- A Engenharia de Software é uma disciplina cujo objetivo é produzir software isento de falhas, entregue dentro do prazo e orçamento previstos, e que atenda a necessidade do cliente. **[Schach, 2009]**
- O estabelecimento de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais **[Fritz Bauer]**
- (1). Engenharia que aplica: uma abordagem sistemática, disciplinada e quantificável; os princípios da ciência da computação, design, engenharia, administração, matemática, psicologia, sociologia e outras disciplinas se necessário for; e às vezes pura invenção, para criar, desenvolver, operar e manter de forma econômica, confiável e correta, soluções de alta qualidade para problemas que envolvam software. (2). Engenharia de Software também é o estudo e a busca por abordagens para a realização das atividades (1). **[Berry, 1992]**

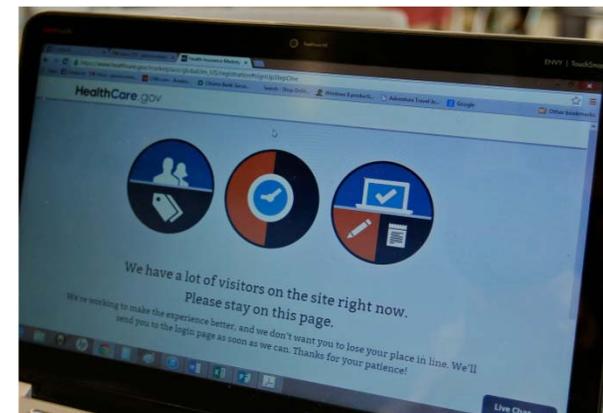


Qual a importância de se estudar  
Engenharia de Software?



# Software

YEAR	COMPANY	OUTCOME (COSTS IN US \$)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to \$33.3 million* loss.
2004-05	UK Inland Revenue	Software errors contribute to \$3.45 billion* tax-credit overpayment.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system canceled after \$54.5 million <sup>†</sup> is spent.
2004	Ford Motor Co.	Purchasing system abandoned after deployment costing approximately \$400 million.
2004	J Sainsbury PLC [UK]	Supply-chain management system abandoned after deployment costing \$527 million.*
2004	Hewlett-Packard Co.	Problems with ERP system contribute to \$160 million loss.
2003-04	AT&T Wireless	Customer relations management (CRM) upgrade problems lead to revenue loss of \$100 million.
2002	McDonald's Corp.	The Innovate information-purchasing system canceled after \$170 million is spent.
2002	Sydney Water Corp. [Australia]	Billing system canceled after \$33.2 million <sup>†</sup> is spent.
2002	CIGNA Corp.	Problems with CRM system contribute to \$445 million loss.
2001	Nike Inc.	Problems with supply-chain management system contribute to \$100 million loss.
2001	Kmart Corp.	Supply-chain management system canceled after \$130 million is spent.
2000	Washington, D.C.	City payroll system abandoned after deployment costing \$25 million.
1999	United Way	Administrative processing system canceled after \$12 million is spent.
1999	State of Mississippi	Tax system canceled after \$11.2 million is spent; state receives \$165 million damages.
1999	Hershey Foods Corp.	Problems with ERP system contribute to \$151 million loss.
1998	Snap-on Inc.	Problems with order-entry system contribute to revenue loss of \$50 million.
1997	U.S. Internal Revenue Service	Tax modernization effort canceled after \$4 billion is spent.
1997	State of Washington	Department of Motor Vehicle (DMV) system canceled after \$40 million is spent.
1997	Oxford Health Plans Inc.	Billing and claims system problems contribute to quarterly loss; stock plummets, leading to \$3.4 billion loss in corporate value.
1996	Arianespace [France]	Software specification and design errors cause \$350 million Ariane 5 rocket to explode.
1996	FoxMeyer Drug Co.	\$40 million ERP system abandoned after deployment, forcing company into bankruptcy.
1995	Toronto Stock Exchange [Canada]	Electronic trading system canceled after \$25.5 million** is spent.
1994	U.S. Federal Aviation Administration	Advanced Automation System canceled after \$2.6 billion is spent.
1994	State of California	DMV system canceled after \$44 million is spent.
1994	Chemical Bank	Software error causes a total of \$15 million to be deducted from 100 000 customer accounts.
1993	London Stock Exchange [UK]	Taurus stock settlement system canceled after \$600 million** is spent.
1993	Allstate Insurance Co.	Office automation system abandoned after deployment, costing \$100 million.
1993	London Ambulance Service [UK]	Dispatch system canceled in 1990 at \$1125 million**; second attempt abandoned after deployment, costing \$15 million.**
1993	Greyhound Lines Inc.	Bus reservation system crashes repeatedly upon introduction, contributing to revenue loss of \$61 million.
1992	Budget Rent-A-Car, Hilton Hotels, Marriott International, and AMR [American Airlines]	Travel reservation system canceled after \$165 million is spent.



HEALTH CARE

## Traffic Didn't Crash the Obamacare Site Alone. Bad Coding Did Too.

By Michael Scherer @michaelscherer | Oct. 24, 2013 | 252 Comments

Health insurance

### The Obamacare software mess

Signing up for Obamacare is still ridiculously hard

## Obamacare Website Programmers Complained About Unrealistic Deadlines

By JACK GILLUM and JULIE PACE 10/22/13 08:04 AM ET EDT AP





# Falhas

- Algumas falhas notáveis
  - 1979 – O sistema de defesa americano disparou um alarme sobre um ataque de mísseis da União Soviética
  - 1985 – Devido a uma falha no software de um equipamento médico, pacientes morreram devido a overdose de radiação
  - 1991 – Na Guerra do Golfo, a bateria anti-míssil Patriot ficou operando por mais de 100 horas, causando uma falha em um acumulador. Um míssil Scud atingiu um acampamento militar
  - 2003 – Uma falha no sistema de pagamento de aposentadoria nos EUA causou o envio de milhares de cheques inválidos
- O que falha mais: um software ou uma ponte?
  - 1940 – Cai ponte em Washington (EUA)  
<https://www.youtube.com/watch?v=XggxeuFDaDU>
  - 2004 – Ponte entre Alemanha e Suíça – ao conectar as duas partes havia uma diferença de altura devido a diferença de interpretação entre “nível do mar”

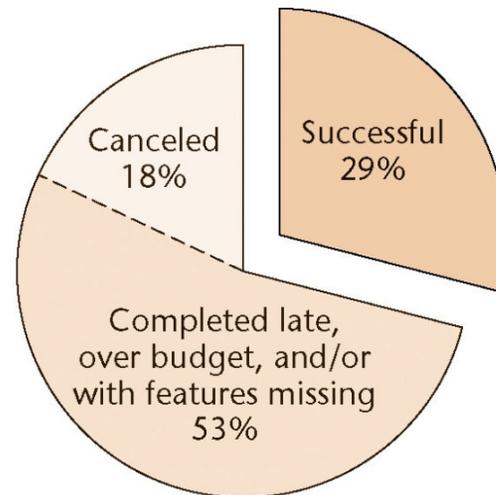


# Engenharia de Software é uma disciplina madura?



## Alguns dados

- Pesquisa Stanish Group (2004) com 9.236 projetos



- Software é entregue:
  - depois do prazo
  - acima do orçamento,
  - com falhas
  - não atende a necessidade do cliente



## Outros dados

- Cutter Consortium (2002):
  - 78% das empresas de TI se envolveram em disputas judiciais por conta de software entregue
  - 67% dos casos o software não entregava o pedido
  - 56% as datas prometidas não foram cumpridas
  - 45% apresentavam falhas graves
- Crise do software (ou Depressão do Software?)



*“Programar é divertido, porém desenvolver software de qualidade é difícil.”*

*Craig Larman*



# Por que é difícil desenvolver software de qualidade?

<http://www.youtube.com/watch?v=ZRUiJXrl8Yg>

Qual a similaridade com Engenharia de Software?



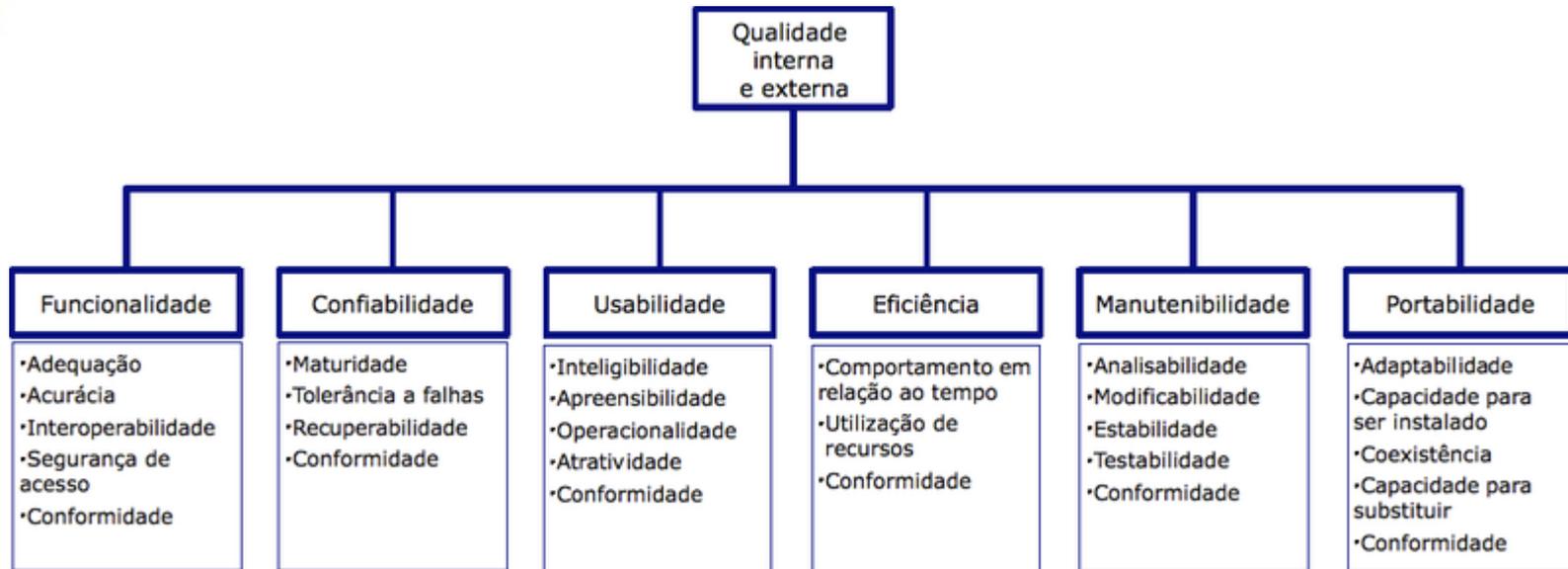
# O que é uma solução boa?

- Atende aos requisitos
- Relação custo x benefício
- Evitar o overkill





# Qualidade de Software



ISO 9126



# Diferentes tipos de desenvolvimento

- Desenvolvimento interno por demanda
- Desenvolvimento por contrato
- Desenvolvimento de COTS
- Desenvolvimento de linha de produto
- Desenvolvimento de F/OSS (Free and Open Source Software)
  - “Com um bom número de olhos, todos os bugs são superficiais”  
(Raymond, 2000) => Lance o produto logo e frequentemente.
- Desenvolvimento Web
- Desenvolvimento sistemas críticos
- Desenvolvimento de sistemas de tempo real
- Desenvolvimento de sistema embarcado
- Desenvolvimento de sistemas científicos
- etc.



# Desenvolvimento de software

- O engenheiro de software constrói soluções computacionais para problemas dos usuários – quase sempre não são problemas da área de informática.
- Necessidade de criação e invenção contínua
- A **experiência**, a **criatividade** e a **perspicácia** são fundamentais
- Um programador bom usando bloco de notas é melhor do que um programador mediano mais equipado?



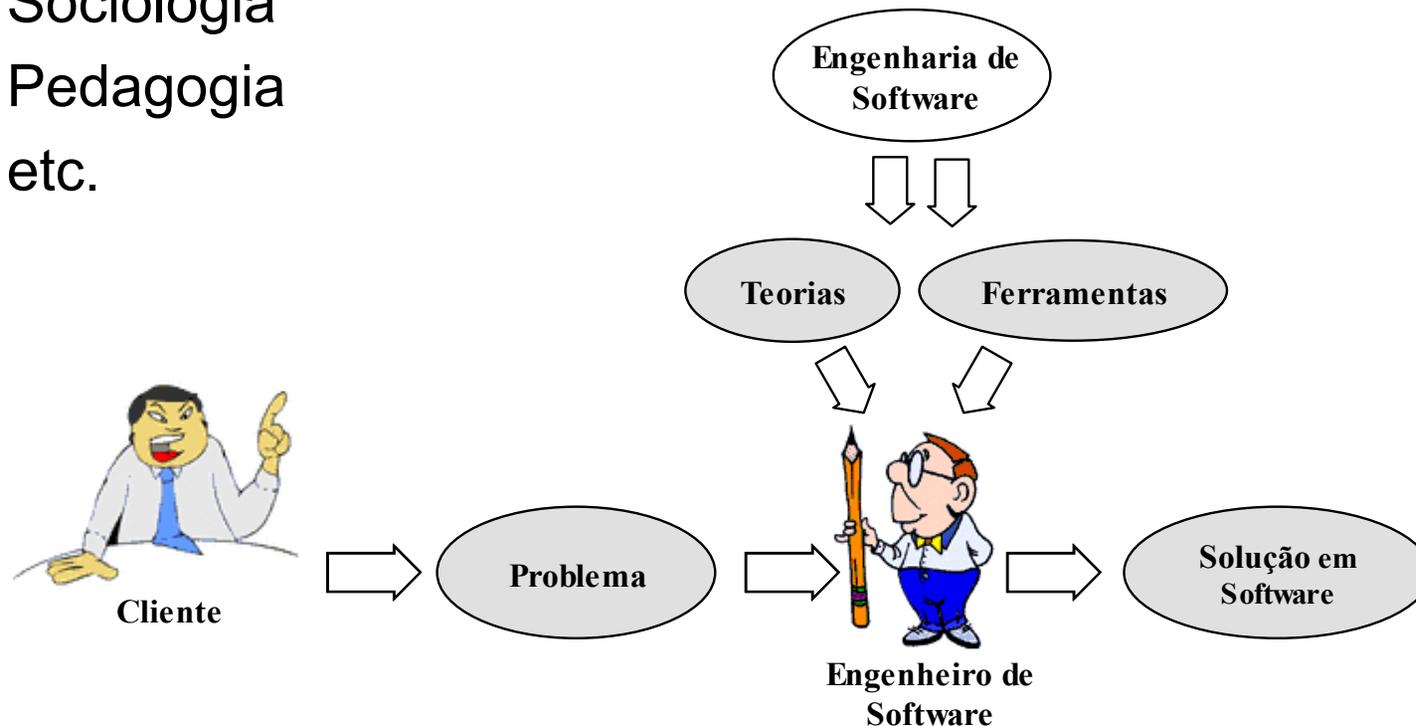
O que faz parte do escopo de estudo da Engenharia de Software?

Quais atividades fazem parte da Engenharia de Software?



# Competências envolvidas na ES

- Matemática e ciência da computação
- Economia
- Administração
- Psicologia
- Sociologia
- Pedagogia
- etc.

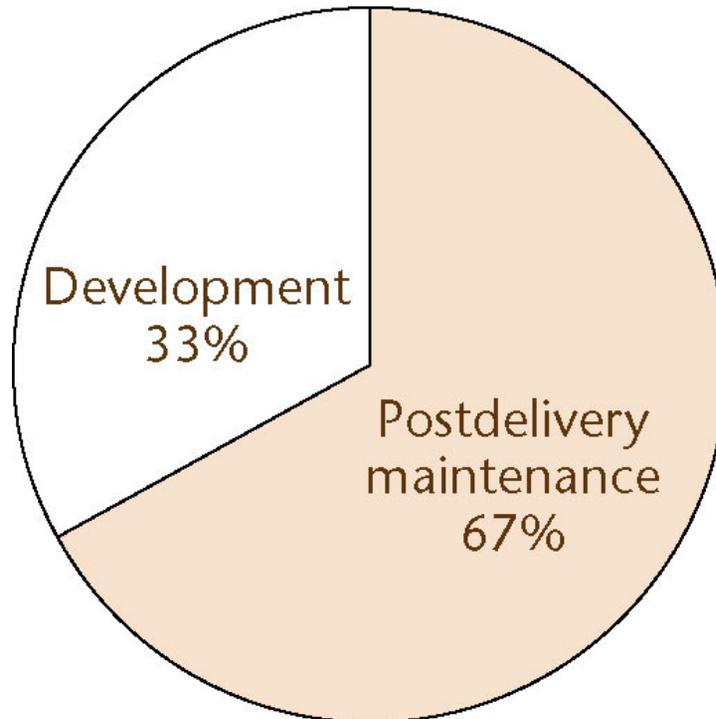




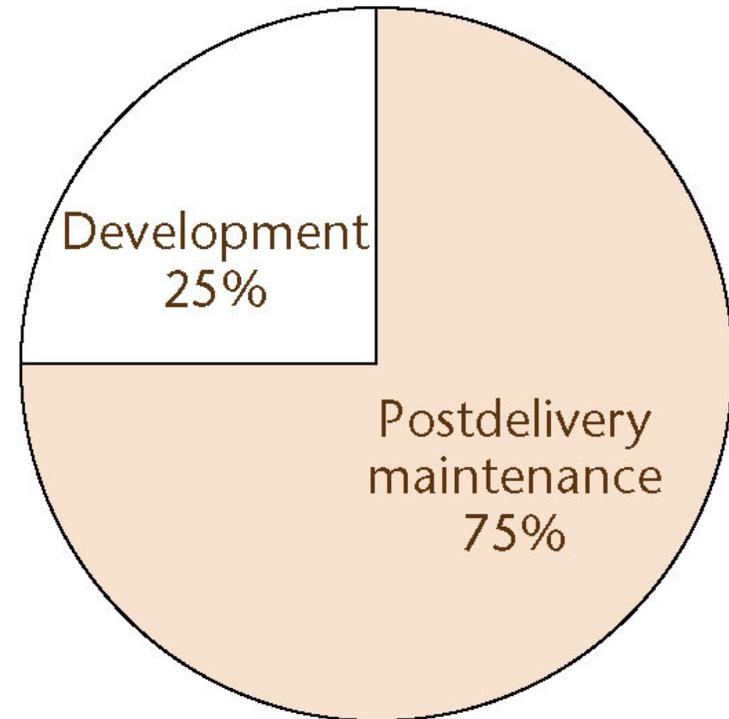
Qual a atividade de maior custo na Engenharia de Software?



# Desenvolvimento x Manutenção



(a)  
1976-1981



(b)  
1992-1998

[Schach, 2009]

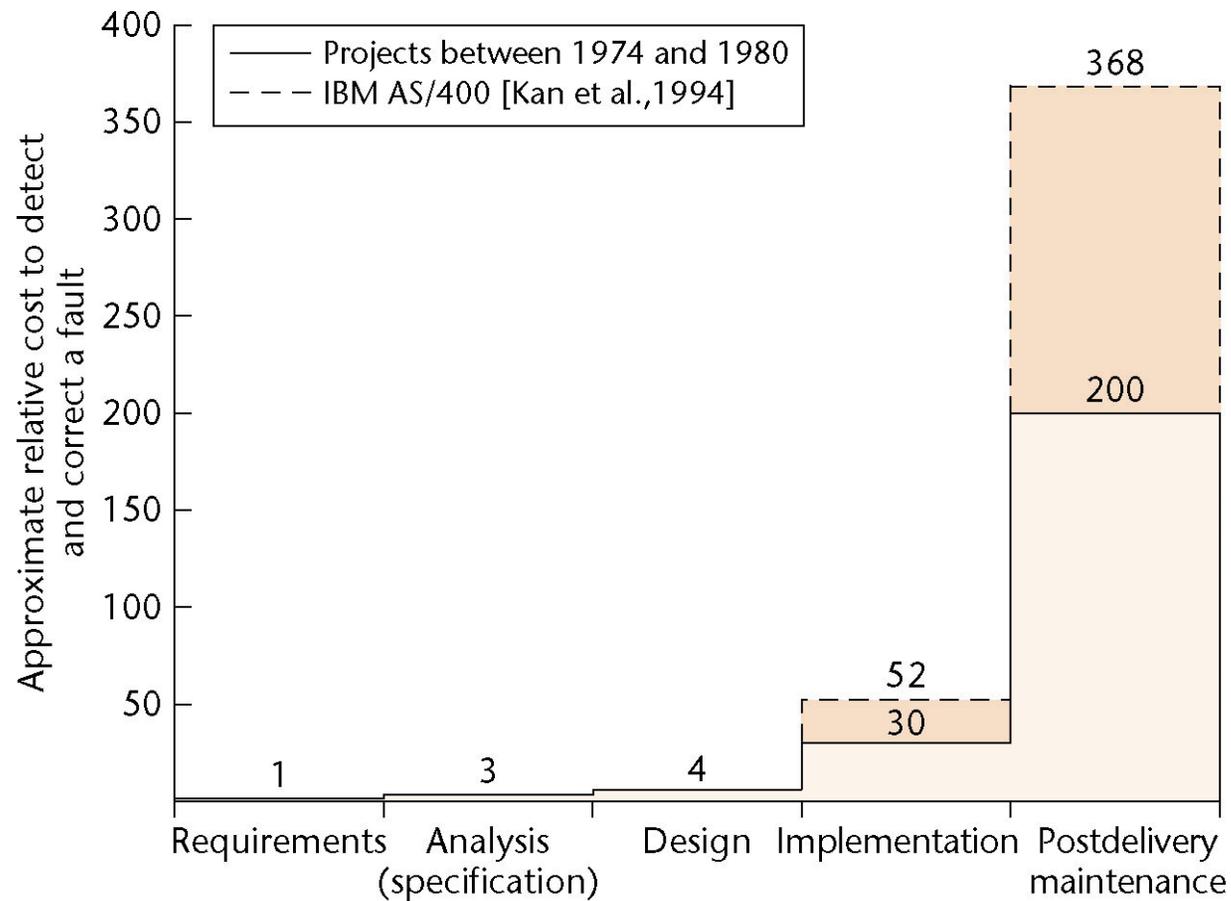
Para onde direcionar os esforços?



# Custo de uma falha

Projetos na IBM

[Boehm, 1981]



Para onde direcionar esforços?



Há quantos anos existe a Engenharia de Software?



# História

- A indústria de software começou no final dos anos 50
- Computer Usage Corporation (CUC) fundada em 1955 por dois ex-funcionários da IBM – primeira empresa de desenvolvimento de software
  - Em 1967 tinha 700 funcionários em 12 escritórios, com uma receita de mais de US\$ 13 milhões
- Estima-se que em 1967 havia 2.800 empresas de software nos EUA
- Um grupo de estudos da OTAN cunhou o termo Engenharia de Software em 1967
- NATO Conference 1968

Software History Museum  
<http://www.softwarehistory.org>



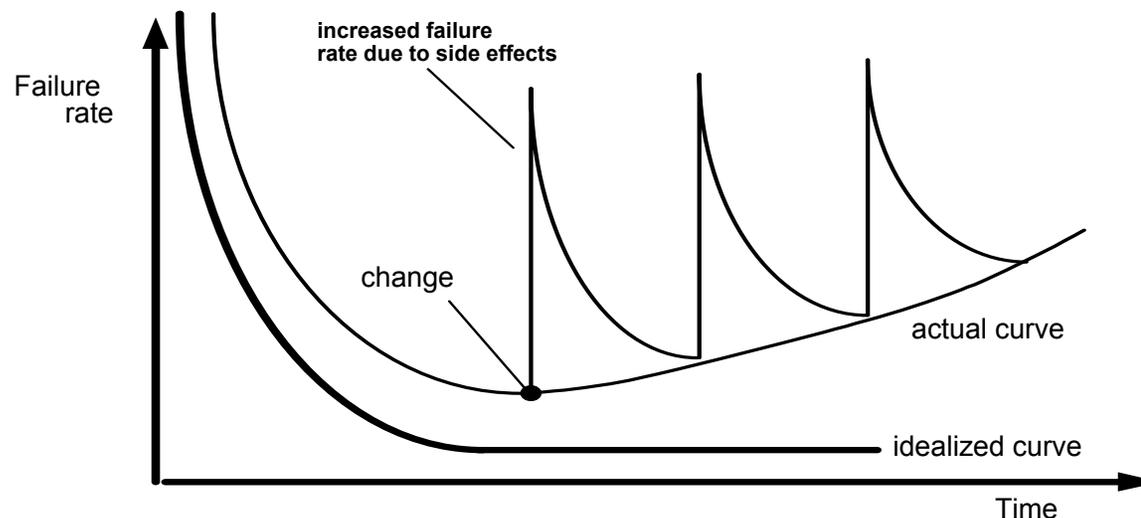
## Comparações

- Engenharia de Software x Engenharia de algoritmos
- Engenharia de Software x Engenharia de hardware
- Quais as particularidades da Engenharia de Software com relação às demais engenharias? Ela poderia ser uma das engenharias da Poli?
- Engenharia de Software do ponto de vista da Ciência da Computação x Sistemas de Informação



# Desenvolvimento de sw x hardware

- Custos do software são concentrados na engenharia
- Software não desgasta, mas se deteriora
- Fase de fabricação de um hardware pode produzir problemas de qualidade
- Quando um hardware falha, pode substituir por um sobressalente



[Pressman, 2006]

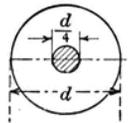
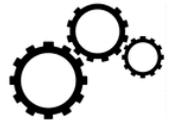


# ES é mesmo uma Engenharia?

- Atualmente alguns pesquisadores começam a contestar essa apropriação.



Engenharia tradicional	Engenharia de software
Escopo mais restrito	Incontáveis domínios de aplicação
Soluções restritas por leis físicas	Poucas limitações tecnológicas
Soluções similares	Pluralidade de soluções
Aplicação de técnicas de forma determinística	Criação e invenção contínua
Soluções para problemas específicos	Software modela processos abstratos do mundo real
Uso intenso da matemática	Uso restrito da matemática em algumas etapas





## ES é mesmo uma Engenharia?

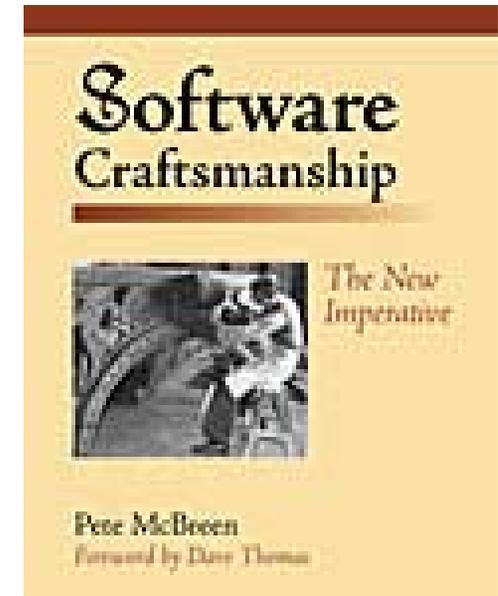
- *“Desenvolvimento de software é um jogo cooperativo de invenção e de comunicação. Nunca foi engenharia, apesar de toda a propaganda neste sentido. Desenvolvimento de software consiste em nada mais do que idéias, concretizadas. Consiste em pessoas inventando e se comunicando, trabalhando em um problema que ainda não entendem, e que não pára de mudar, criando uma solução que ainda não entendem, e que não pára de mudar, expressando suas idéias usando linguagens restritas, que quase não entendem, para um interpretador que não perdoa erros.”*

[Alistair Cockburn, 2001]



# Como se compara com outras analogias?

- Arquitetura
- Arte
- Carpintaria
- Literatura





# Treinamento do desenvolvedor

- Lidar com incerteza e indefinição
- Criatividade e inovação
- Capacidade de avaliar processos, métodos e ferramentas
- Capacidade de adaptar processos, métodos e ferramentas para cenários específicos
- Trabalho em grupo
- Comunicação
  
- Volta ao passado? [Teles, 2004]

## **Sociedade Agrícola**

Trabalho em casa

Ofício

Aprendizagem assíncrono

## **Revolução industrial**

Local de trabalho

Divisão de trabalho

Aprendizagem síncrono

- Aprendizagem de uma manufatura?



## Mitos ou verdades

- Software bem feito não sofre manutenção
- Se nos atrasarmos no cronograma, podemos adicionar mais programadores
- A tecnologia X resolve os problema do desenvolvimento de software (bala de prata)

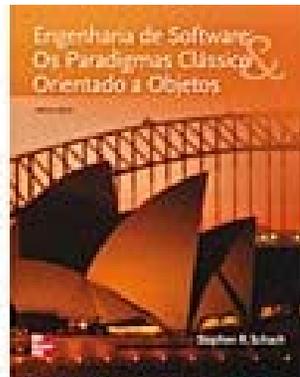


"silver bullet" refers to any **straightforward** solution perceived to have extreme effectiveness

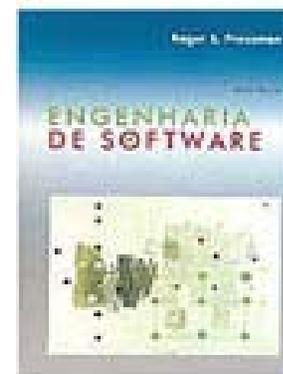
[http://en.wikipedia.org/wiki/Silver\\_bullet](http://en.wikipedia.org/wiki/Silver_bullet)



# Referências desta aula



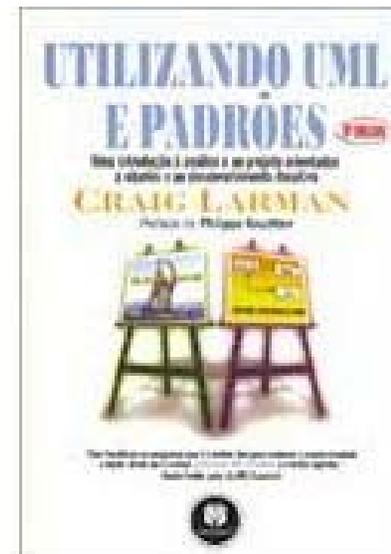
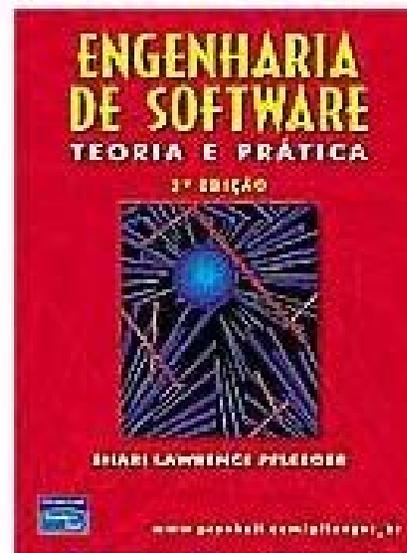
Schach, Cap 01



Pressman, Cap 01



# Outras referências





# A disciplina



# Sobre a disciplina

- **Objetivo:**
  - Compreensão das atividades que compõem o processo de desenvolvimento de software e seus propósitos. Estudo de aplicação de princípios de gerenciamento das atividades e seu impacto no andamento do projeto e no produto final.
  - Apresentar os conceitos básicos da Engenharia de Software e levar os alunos a vivenciarem algumas de suas práticas.
- **Perfil do aluno:**
  - São desejáveis bons conhecimentos e experiência com programação
  - É imprescindível tempo extraclasse para se dedicar às atividades do curso.
- **Ementa:**
  - Gerenciamento de projeto. Estimação de custos. Análise e especificação de requisitos. Especificações formais. Interface com o usuário. Modelagem de dados. Técnicas e modelagens para projeto e implementação: arquitetura de projeto, projeto estruturado, projeto orientado a objetos. Gerenciamento de versões e configurações. Verificação: testes, revisões e inspeções. Validação e certificação de qualidade. Manutenção. Documentação



## Abordagem pedagógica

- *“Escutei e esqueci,  
li e entendi,  
fiz e aprendi.”*
  - Implementação
  - Várias atividades
- Atividades em grupo realizadas de forma colaborativa e individual
  - Aprendizagem side-wide
- Experiência prática enquanto ainda está na universidade
- *O mérito do sucesso é 10% do professor, 10% dos colegas e 80% de si próprio.*



# Avaliação

- Provas
- Participação em sala
- Trabalhos
- Tarefas
- Peer-review
  
- Atividade extra
  
- Monitor: Gustavo Oliva (golivax@gmail.com)



## Atividade extra

### VIBu: Virtual Teams in International Business



- **October 7 & 21, 2014 (Tuesdays) ou October 24 & November 7, 2014**
- **Each student needs to commit to study the pre-session materials and being online during the simulation sessions for approximately 5-6 hours / simulation day and. This 6 hour working can be split into several time slots.**
- *Limitada a 10 participantes*



# Cronograma tentativo

- Não haverá aula
  - 08 e 10 de setembro (semana da pátria)
  - 13 e 15 de outubro (break)
  - 17 e 19 de novembro (break)
  - 27 de outubro (recesso)
  - 01 e 08 de outubro (*talvez*)
- Provas
  - Prova 1: 03/09
  - Prova 2: 01/12
- Trabalhos
  - Projeto 1: 20/10
  - Projeto 2: 26/11
- “Tarefas de casa”
  - Haverá atividades extraclasse para quase todas as aulas



# Tarefas

- Cadastrar-se no Moodle (Paca)
- Ler “Mythical Man-Month” (Frederick Brooks) e fazer um resumo de uma página com as principais ideias do texto
  - <http://www.cs.virginia.edu/~evans/greatworks/mythical.pdf>
  - Entrega: antes da próxima aula pelo Paca
- Definir projetos e grupos
- Atenção (válida para todas atividades da disciplina):
  - Não serão aceitas tarefas atrasadas
  - Projetos atrasados perderão 1 ponto por dia
  - Não será aceito plágio (“cola”):
    - “O Departamento de Ciência da Computação considera uma infração disciplinar inadmissível qualquer forma de plágio. Na ocorrência de tais casos, o Departamento recomenda a reprovação do aluno na disciplina, e que o ocorrido seja relatado à Comissão de Graduação para as devidas providências.” <http://www.ime.usp.br/dcc/grad/plagio>
    - Tanto quem forneceu quanto quem colou serão punidos, com a anulação da entrega, com a perda de ponto e/ou com reprovação.



# Projeto

- Tarefa: Definir os projetos e grupos
- **Prazo: próxima segunda-feira**
- Formato:
  - Título
  - Descrição de um parágrafo do sistema
  - Usuários (quem seriam?)
  - Cliente (terá um cliente real?)
- Grupo de 2 a 3 alunos
- Requisito: deverá ser um sistema web
- Até o final do curso será implementado um protótipo deste sistema



# Revisão

<http://kahoot.it>  
(use o game-pin fornecido)

<http://goo.gl/NTMiHB>