Engenharia de Software

Tecnologia de Software

Aula 1: Visão Geral

Fábio Levy Siqueira
levy.siqueira@usp.br

Quem sou eu?

- Formação
  - Engenheiro de Computação (Poli-USP)
  - Mestrado e Doutorado em Engenharia de Software (Poli-USP)
- Professor
  - PECE - desde 2006
  - Poli-USP (tempo parcial) - desde 2014
- Consultor
- Áreas de pesquisa
  - Engenharia de requisitos
  - Métodos ágeis
  - Engenharia dirigida por modelos

Avaliação

- Critério
  - Média = $\frac{\text{Exercícios} + \text{Trabalho}}{2}$
  - Exercícios
    - Feitos em aula (em geral, em grupo)
  - Trabalho
    - Texto individual sobre um tema/abordagem de ES
    - (Mais detalhes no AlunoWeb)

Estrutura do curso

Aula	Data	Aula	Exercício
1	24/02	Visão geral	
2	03/03	Processos de software	
3	10/03	Engenharia de requisitos	X
4	17/03	Arquitetura e projeto	X
5	24/03	Implementação e teste	X
6	31/03	Métodos ágeis	
7	07/04	Gerência de projetos	X
8	14/04	Manutenção	X
9	28/04	Qualidade de software	X
10	??/??	Dívida técnica e pesquisa em ES	

Bibliografia básica

- BOURQUE, P.; FAIRLEY, R. E (eds.). **SWEBOK: Guide to the Software Engineering Body of Knowledge**. v.3. IEEE Computer Society. 2014.
  - https://www.computer.org/education/bodies-of-knowledge/software-engineering

- PFLEEGER, S. L.; ATLEE, J. M. **Software Engineering: Theory and Practice**. 4. ed. Prentice Hall, 2010.

- PRESSMAN, R.; MAXIM, B. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. AMGH, 2016.

- SOMMERVILLE, I. **Engenharia de Software**. 10. ed. Pearson, 2019.

Visão Geral

Engenharia de Software

- O que é Engenharia de Software?
  - Engenharia (IEEE 610, 1990)

A aplicação de uma abordagem sistemática, disciplinada e quantificada para estruturas, máquinas, produtos, sistemas ou processos.

  - Software (IEEE 610, 1990)

Programas de computador, procedimentos e possivelmente a documentação associada e os dados relativos à operação de um sistema computacional.

Engenharia de Software

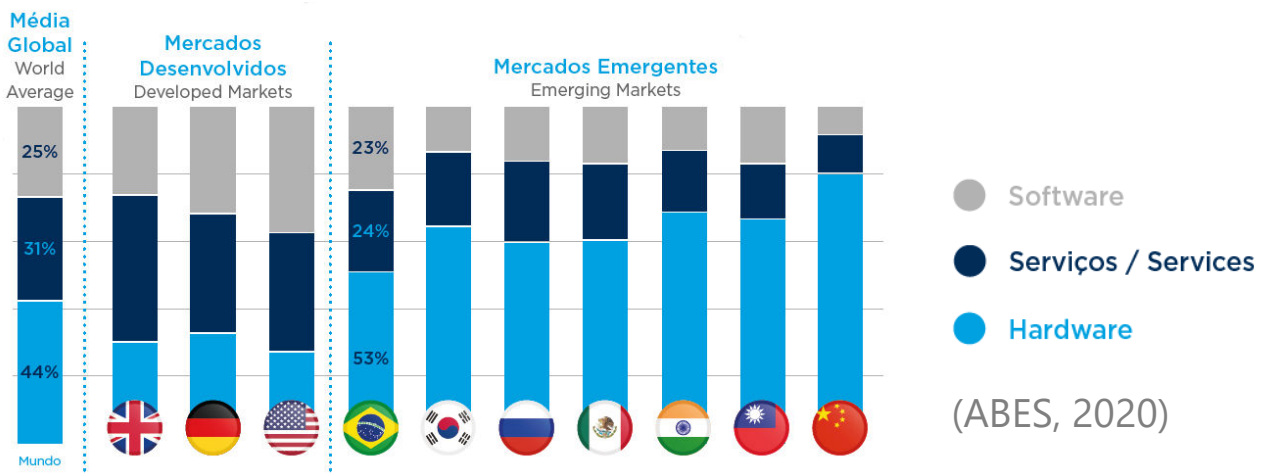
- O que é Engenharia de Software?
  - Definição (IEEE 610, 1990)

    A. A aplicação de uma abordagem sistemática, disciplinada e quantificada para o desenvolvimento, a operação e a manutenção do software, isto é a **aplicação de engenharia ao software**
    B. O estudo de abordagens como definido em (A)

  - 1º uso: 1968 em uma conferência da OTAN

Por que se preocupar com a ES?

- Ubiquidade do Software
- Mercado de TI (ABES, 2020)
  - Mundial: US$ 2,368 trilhões
  - Brasil: US$ 43 bilhões (10º mercado doméstico)
    - Software: US$ 10,056 bilhões
      - 5.519 empresas de software e produção de software



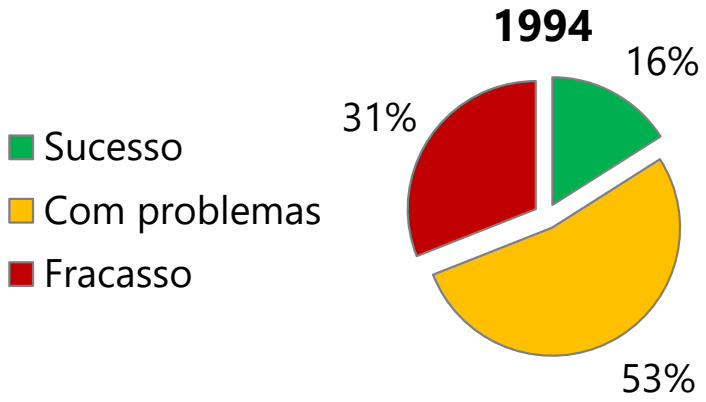
(ABES, 2020)

Por que se preocupar com a ES?

- Complexidade crescente (fevereiro 2021)
  - Kernel do Linux
  - MySQL
  - Firefox

Por que se preocupar com a ES?

- Dificuldade em desenvolver software



(Standish Group, 1994; 2015)

Por que se preocupar com a ES?

- Algumas razões para projetos de software falharem

Por que se preocupar com a ES?

- Custo da baixa qualidade de software nos EUA em 2020
  - US$ 2,08 trilhões (estimativa)
    - Falhas operacionais de software: US$ 1,56 trilhões
      - Envolve falhas de segurança
    - Projetos de software sem sucesso: US$ 0,26 trilhões

(CISC, 2021)

Por que se preocupar com a ES?

- Problemas causados por defeitos de software
  - Roubo de informações de crédito da Equifax (2017)
    - 143 milhões de registros roubados por Hackers
      - 50% dos norte americanos
  - WannaCry (2017)
    - Ransomware: 230.000 computadores em 150 países
      - Impacto de US$ 4 bilhões
    - Vulnerabilidade do Windows (já corrigida)
  - SISU (2019)
    - Instabilidade e usuários alegam acesso a dados de outros candidatos
    - 500.000 usuários simultâneos (?!?)

Por que se preocupar com a ES?

- Problemas causados por defeitos de software
  - Airbags da Nissan (2017)
    - 1 milhão de carros
    - Airbag não identifica adultos no banco de passageiro (e não infla)
    - 2 acidentes identificados
  - Boeing (2019)
    - Defeito no software de simulador de voo: 737 MAX
      - Incapaz de reproduzir algumas situações de voo
    - 300 mortes

Por que se preocupar com a ES?

- Importância da manutenção
  - 49% a 75% do custo do software é manutenção

    (Grubb e Takang, 2003)
  - Alguns problemas
    - Equipe diferente da de desenvolvimento (em geral)
      - *Turnover* da equipe de manutenção
    - Complexidade do Software
  - *Exemplo*: Bug do milênio
    - Custo de correção estimado em US$ 400 bilhões (corrigido)

      (http://www.computerworld.com/s/article/9142555)

Por que se preocupar com a ES?

- Custo de corrigir defeitos aumenta com o tempo
  - Média do custo da indústria para corrigir um defeito

(Pressman e Maxim, 2016)

  - Abordagens de ES tendem a melhorar essa curva

Particularidades

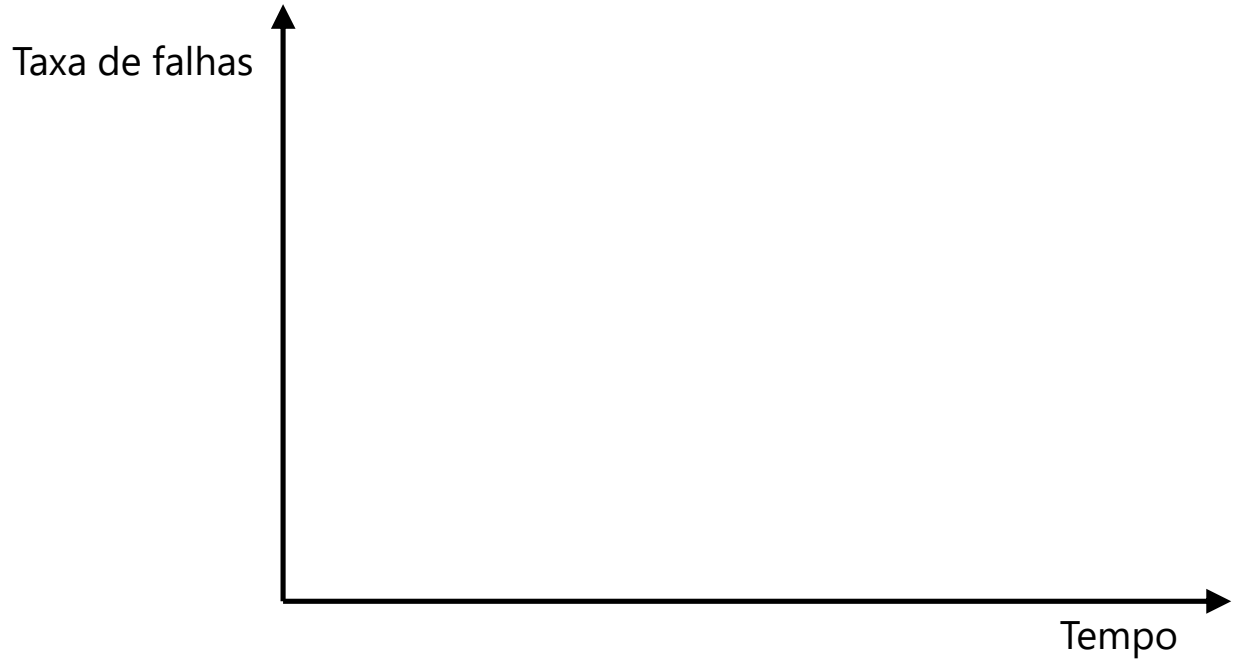
- O que tem de diferente no software?
  - Não é um produto como outro?
  - O projeto de software não é similar/igual a outros projetos?

Particularidades

- Produto intangível
  - Não desgasta!
    - Leis físicas e propriedade dos materiais não são relevantes
  - Hardware

Particularidades

- Produto intangível
  - Software



Particularidades

- Desenvolvido e não manufaturado
  - Necessidade de criatividade e disciplina
    - Projeto e implementação tem uma arte envolvida
  - E as fábricas de software?

- Maioria do software é personalizado
  - Dificuldade de reuso
  - *Frameworks* e bibliotecas
  - Software de linha de produto
    - (Forma mais geral do *software de prateleira*)

Particularidades

- Falta de uma "Teoria fundamental"
  - Entendimento da relação entre processo e produto
    - Como desenvolver um determinado tipo de software?
    - Tipos de Software (Pressman e Maxim, 2016)
      - Software de sistemas
        - *Ex.*: drivers, SO
      - Software de aplicação
        - *Ex.*: automação
      - Software científico
        - *Ex.*: análise de DNA
      - Software embarcado
        - *Ex.*: impressora e TV
      - Software de linha de produto
        - *Ex.*: pacote Office, ERPs
      - Aplicações Web / Mobile
        - *Ex.*: loja virtual
      - Software de IA
        - *Ex.*: jogos, reconhecimento de padrões

Particularidades

- Falta de uma "Teoria fundamental"
  - Diversidade de processos e métodos
  - Dificuldade de fazer previsões antes de construir o software
  - Iniciativas
    - Essence - Kernel and Language for Software Engineering Methods (SEMAT)
    - SWEBOK (IEEE)
    - Padrões (ISO e IEEE)
    - Código de Ética (ACM)
    - Certificação (IREB, IIBA, Scrum Alliance, SAFe etc.)
    - Currículos de computação (ACM)
    - Diversas conferências e periódicos

Particularidades

- Evolução rápida de tecnologias
  - Necessidade de treinamento constante
  - Dificuldade de consolidar o corpo de conhecimento
  - Distância entre teoria e prática

- Aparente facilidade de fazer mudanças
  - Necessidade de mudanças!

Particularidades

- Falta de fronteiras
  - Dificuldade de distribuir o trabalho
  - Dificuldade de comunicação
  - Problemas para gerenciar a equipe

Mitos

- O *único* produto é o software
  - Necessidade de manuais, modelos, planos, etc.
    - ...manutenção...
- Engenharia de Software é burocrática
  - ...agilidade ≠ cada um faz o que quer...
- Não é possível avaliar o software antes de construí-lo
  - Controle e garantia da qualidade

Mitos

- "Bala de prata" da Engenharia de Software
  - Novo processo / Ferramenta / Método resolve tudo!
  - Não há uma bala de prata!

- Aumentar a equipe ajuda a terminar o projeto
  - Lei de Brooks

Adicionar pessoas a um projeto de software atrasado torna-o mais atrasado.

Mitos

- O trabalho acaba quando se entrega o software
  - E a manutenção?

- Terceirizar acaba com a preocupação
  - Acompanhamento é ainda necessário
  - ...nem sempre é mais barato terceirizar...

Desenvolvimento de software

- Na prática: como desenvolver software?
  - É só definir os requisitos e implementar?
    - "Método ágil"

- Por que não? O que impede / dificulta isso?

Desenvolvimento de software

- Alguns fatores

Referências

- ABES. **Mercado Brasileiro de Software: Panorama e Tendências**. ABES, 2020.
- CISQ. **The Cost of Poor Software Quality in the US: A 2020 Report**. 2021.
- FORBES TECHNOLOGY COUNCIL. **14 Common Reasons Software Projects Fail (And How To Avoid Them)**. Disponível em: https://www.forbes.com/sites/forbestechcouncil/2020/03/31/14-common-reasons-software-projects-fail-and-how-to-avoid-them. 2020.
- GRUBB, P.; TAKANG, A. A**. Software Maintenance: Concepts and Practice**. 2. ed. World Scientific Pub Co, 2003.
- IEEE. **IEEE Standard Computer Dictionary**. IEEE 610. IEEE, 1990.
- KRUCHTEN, P. **Putting the "engineering" into "software engineering"**. Anais da Australian Software Engineering Conference. IEEE, 2004.
- STANDISH GROUP. **The Chaos Report**. Standish Group, 1994.
- STANDISH GROUP. **The Chaos Report**. Standish Group, 2015.