Introdução à Engenharia de Software (Unidade 01)

Pós-graduação em Engenharia de Software

Processos Prescritivos e Ágeis para Desenvolvimento de Software

Prof. Humberto Torres Marques Neto

Março de 2019

PUC Minas Virtual

Objetivos

Os objetivos da Unidade 1 de 4 da disciplina de Processo Prescritivos e Ágeis para Desenvolvimento de Software são:

- Apresentar a área de Engenharia de Software
- Discutir a importância do software na sociedade contemporânea
- Mostrar a necessidade de se construir software de qualidade com produtividade

Referências Bibliográficas

Básica:
PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 8 ed. McGraw-Hill, 2016.

Complementar:
SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

O custo do hardware

Durante as três primeiras décadas da era da computação, o principal desafio era desenvolver um hardware que reduzisse o custo de processamento e de armazenagem de dados

Isto ocorreu?
Com qual velocidade?

Um breve histórico da Computação (1/5)

- Primeira era (195? e 196?)
  - Orientação para o processamento batch
  - Distribuição limitada de softwares
  - Softwares extremamente padronizados
  - Produção artesanal sob demanda
  - Os profissionais de informática mantinham em suas cabeças o projeto das aplicações

Um breve histórico da Computação (2/5)

- Segunda era (196? e 197?)
  - Sistemas multiusuários
  - Surgimento de *software houses* e *bureaus* de processamento
  - Surgimento dos pacotes de softwares, que passam a ser tratados como produto

Um breve histórico da Computação (2/5)

- Segunda era (196? e 197?)
  - Processamentos real-time
  - Início da utilização da tecnologia de Banco de dados
  - Surgimentos dos cursos de Ciência da Computação

Um breve histórico da Computação (3/5)

- Terceira era (197? e 198?)
  - Sistemas distribuídos, principalmente em função da evolução das tecnologias de redes de computadores
  - Inteligência embutida nos microprocessadores
  - Diminuição do custo do hardware
  - Os sistemas de computação passam a interferir no cotidiano das pessoas

Um breve histórico da Computação (4/5)

- Quarta era (198? até 200?)
  - Aumento do poder das aplicações desktop
  - Tecnologia de orientação à objetos
  - Sistemas especialistas
  - Redes neurais
  - Computação paralela
  - Arquitetura *multitier*

Um breve histórico da Computação (5/5)

- Quinta era (200? até hoje?)
  - Consolidação dos sistemas Web
  - Sistemas distribuídos de grande escala
  - Aplicações para dispositivos móveis
  - *Cloud Computing*
  - *Big Data*

Em síntese (1/2)

- A velocidade de evolução do hardware foi e está sendo muito superior a velocidade de evolução do software
- Os softwares não têm conseguido acompanhar as mudanças impostas pelos ambientes organizacionais

Em síntese (2/2)

- As organizações estão cada vez mais dependentes dos recursos advindos dos sistemas baseados em computador
- É necessário construir softwares cada vez mais confiáveis e de qualidade
- O prazo para desenvolvimento de um software é cada vez mais curto

Algumas questões para reflexão (1/2)

- Por que a finalização dos softwares é tão demorada?
- Por que os custos de desenvolvimento de software são tão elevados?
- Por que não é possível identificar os erros antes de entregar o software para o usuário?

Algumas questões para reflexão (2/2)

- Por que gastamos muito tempo e esforço mantendo os softwares existentes?
- Por que se tem dificuldades em acompanhar o progresso do desenvolvimento do software enquanto ele está sendo construído?

Alguns problemas (1/4)

- Relacionados ao **Processo**
  - Cronogramas muito otimistas
  - Gerência de risco inexistente
  - Falha de contratação de recursos
  - Planejamento insuficiente
  - Abandono do planejamento por problemas de tempo

Alguns problemas (2/4)

- Relacionados ao **Processo**
  - Gasto de tempo durante a concepção
  - Corte míope de atividades que não sejam codificação (ex: análise ou arquitetura)
  - Design inadequado
  - SQA ou SCM inexistente

Alguns problemas (3/4)

- Relacionados ao **Produto**
  - Requisitos mal formulados, ou mal entendidos
    - +/- 25% dos requisitos mudam em projetos
  - Desenvolvedores com tecnologias “folheadas em ouro”
  - Negociação “Puxa-empurra”
    - Cronograma é esticado, e mais tarefas são adicionadas

Alguns problemas (4/4)

- Relacionados à **Tecnologia**
  - Superestimar ganhos no uso de uma nova tecnologia
  - Trocar de ferramentas no meio do projeto
  - Falha de controle automático do código fonte (sem controle de versão)

Software e suas Aplicações

PUC Minas
Virtual

Um conceito para SOFTWARE

"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 como na virtual, descrevendo a operação e o uso dos programas." (PRESSMAN, 2016. p. 4)

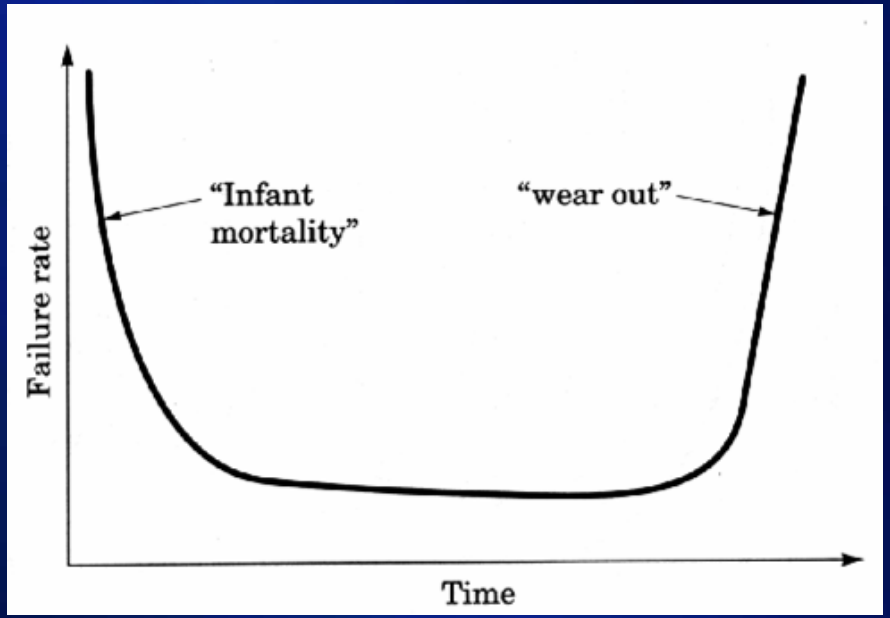
Outro conceito para SOFTWARE

"Software é um lugar onde sonhos são plantados e pesadelos são colhidos, um pântano abstrato e místico onde demônios terríveis competem com mágicas panaceias, um mundo de lobisomens e balas de prata. Por Brad J. Cox"
(PRESSMAN, 2016. p. 4)

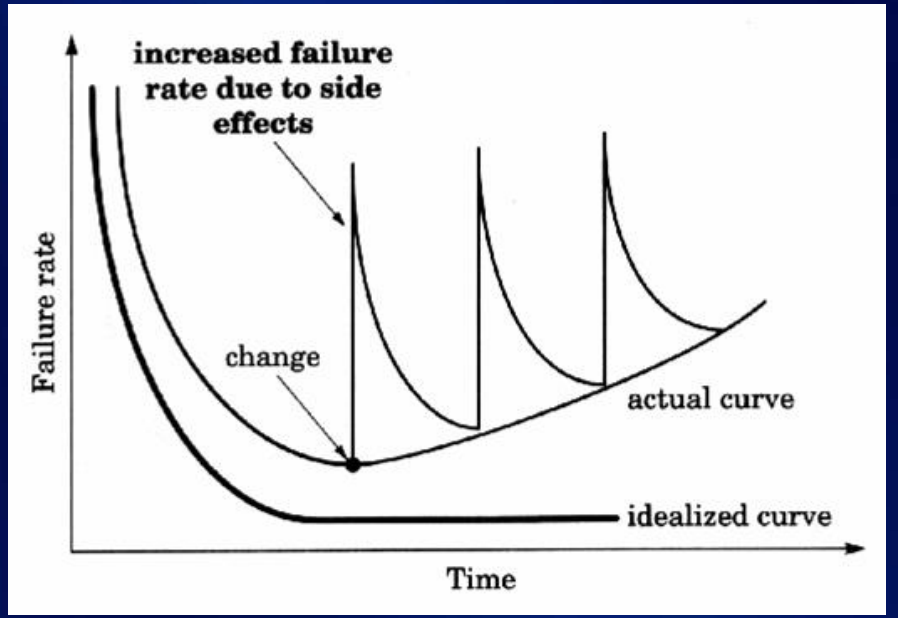
Características do Software

- O software é desenvolvido ou projetado por engenharia, e não manufaturado no sentido clássico
- O software não “se desgasta”
- Embora a indústria caminhe para a construção com base em componentes, a maioria dos softwares continua a ser construída de forma personalizada (sob encomenda)

Curva de falha (PRESSMAN, 2016. p. 5-6)



Hardware



Software

Natureza do Software (1/2)

- Software é um produto
  - produz, gerencia, adquire, modifica, exibe ou transmite informação

Natureza do Software (2/2)

- Software é um veículo para entrega de um produto
  - controla outros software (Sistemas Operacionais)
  - viabiliza a comunicação de dados (Redes)
  - facilita a construção de outros softwares

Aplicações do Software (1/11)

- Software de sistemas
  - programas que apoiam o funcionamento de outros programas
  - forte interação com o hardware

Aplicações do Software (2/11)

- Software de tempo real
  - monitora, analisa e controle eventos do mundo real
  - Tempo real é diferente de interativo ou time-sharing

Aplicações do Software (3/11)

- Software de aplicação
  - amplamente difundido
  - estruturam os dados de forma a facilitar a gestão das organizações e a vida das pessoas

Aplicações do Software (4/11)

- Software científico e de engenharia
  - vão desde a astronomia até a vulcanologia
  - trabalham e processam números
  - CAD

Aplicações do Software (5/11)

- Software embutido (*embedded software*)
  - reside na memória só de leitura (*read only*)
  - controla produtos e sistemas no mercado industrial

Aplicações do Software (6/11)

- Software para linhas de produto
  - textos, planilhas, gerenciamento de dados, aplicações financeiras de cunho pessoal
  - interface sempre inovadora

Aplicações do Software (7/11)

- Software que usa recursos da Web
  - Características das *WebApps*:
    - Uso intensivo de redes
    - Simultaneidade
    - Carga não previsível
    - Desempenho
    - Disponibilidade
    - Orientação a dados
    - Sensibilidade no conteúdo
    - Evolução contínua
    - Imediatismo
    - Segurança
    - Estética

Aplicações do Software (8/11)

- Software de Inteligência Artificial
  - faz uso de algoritmos não numéricos para resolver problemas complexos que não sejam favoráveis à computação
  - sistemas especialistas baseados no conhecimento
  - redes neurais artificiais

Aplicações do Software (9/11)

- *Netsourcing*
  - softwares simples e sofisticados que funcionam a partir dos recursos da Internet

Aplicações do Software (10/11)

- Software Livre
  - possuem código fonte auto descritivo que facilita a sua modificação e evolução

Aplicações do Software (11/11)

- Computação Ubíqua
  - softwares que permitem pequenos dispositivos e computadores pessoais se comunicarem em qualquer ambiente criando um contexto *always-on*

Software Legado (1/2)

- Os softwares precisam estar adaptados aos novos ambientes e às novas tecnologias
- Os softwares crescem para atender os novos requisitos

Software Legado (2/2)

- Os softwares precisam estender a sua interoperabilidade
- Os softwares precisam ser rearquitetados para os novos ambientes de rede

Engenharia de Software

Alguns fatos reais

- Entender o problema antes de desenvolver uma solução inovadora de software
  - Contudo, o software não pode complicar a vida do usuário
- Projetar é uma atividade fundamental
- Um software deve ter uma qualidade elevada
- O software deve ser fácil de ser mantido

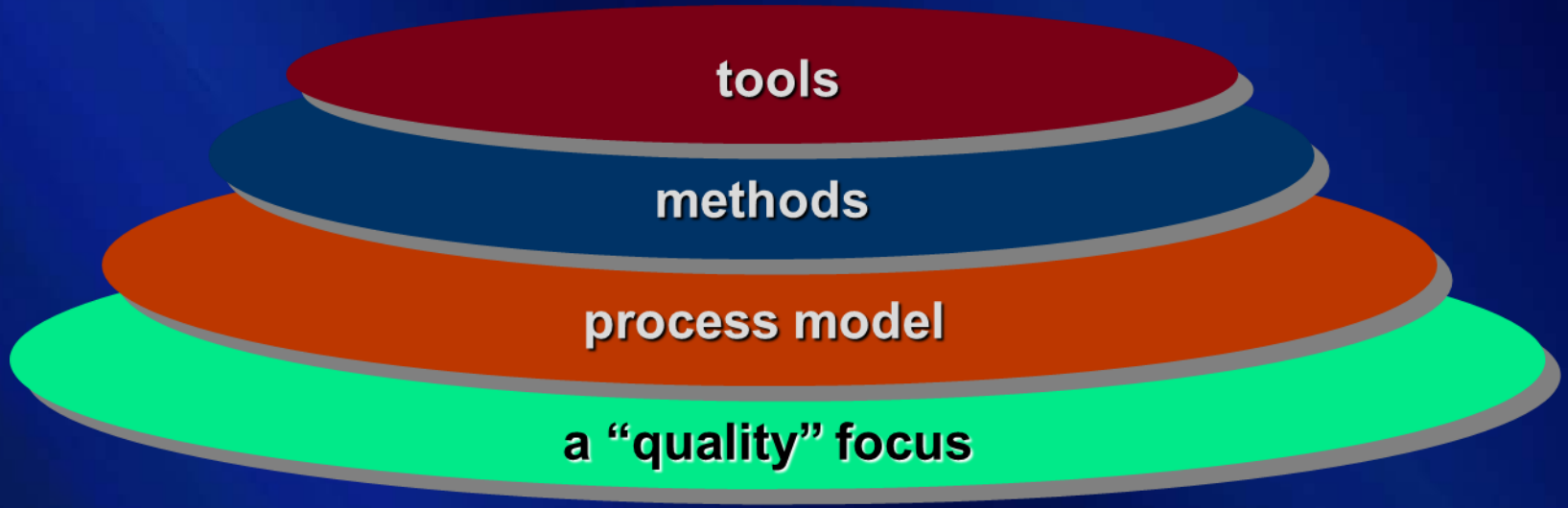
Uma definição

[Engenharia de software é] o estabelecimento e o emprego de sólidos princípios de engenharia de modo a obter software de maneira econômica, que seja confiável e funcione de forma eficiente em máquinas reais

A definição do IEEE

Engenharia de software: (1) A aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software; isto é, a aplicação de engenharia ao software. (2) O estudo de abordagens como definido em (1).

A tecnologia em camadas



(PRESSMAN, 2016. p. 16)

Questões para a Engenharia de Software

- Qual problema tem que ser resolvido?
- Quais características do software são utilizadas para resolver o problema?
- Como o software será construído?
- Como os erros serão identificados?
- Como o software será mantido?

A prática da Engenharia de Software

- Em 1945, George Polya apresentou em linhas gerais a essência da solução de problemas em seu livro *How to Solve It*
- Esses princípios são válidos até hoje!

A essência da prática (1/4)

- Compreenda o problema
  - Quem tem interesse na solução do problema?
  - Quais são as incógnitas?
  - O problema pode ser compartimentalizado?
  - O problema pode ser representado graficamente?

A essência da prática (2/4)

- Planeje a solução
  - Você já viu problemas similares?
  - Algum problema similar já foi resolvido?
  - É possível definir subproblemas?
  - É possível representar uma solução de maneira que conduza a uma implementação efetiva?

A essência da prática (3/4)

- Execute/leve adiante o plano
  - A solução se adéqua ao plano?
  - Cada uma das partes componentes da solução está provavelmente correta?

A essência da prática (4/4)

- Examine o resultado
  - É possível testar cada parte componente da solução?
  - A solução produz resultados que adéquam aos dados, às funções e características necessários?

Mitos Relativos ao Software

Ao contrário dos mitos antigos, que ofereciam lições humanas, os mitos relacionados ao software propagam desinformação e confusão.

Mitos de Gerenciamento (1/2)

- Já temos um livro que está cheio de padrões e procedimentos para desenvolver software; ele não supre meu pessoal com tudo que eles precisam saber?

Mitos de Gerenciamento (2/2)

- Se o cronograma atrasar, poderemos acrescentar mais programadores e ficarmos em dia.
- Se eu decidir terceirizar o projeto de software, posso simplesmente relaxar e deixar essa empresa realizá-lo.

Mitos dos Clientes

- Uma definição geral dos objetivos é suficiente para começar a escrever os programas — podemos preencher detalhes posteriormente.
- Os requisitos de software mudam continuamente, mas as mudanças podem ser facilmente assimiladas, pois o software é flexível.

Mitos dos Profissionais da Área (1/2)

- Uma vez feito um programa e o colocado em uso, nosso trabalho está terminado.
- Até que o programa entre em funcionamento, não há maneira de avaliar sua qualidade.

Mitos dos Profissionais da Área (2/2)

- O único produto passível de entrega é o programa em funcionamento.
- A engenharia de software nos fará criar documentação volumosa e desnecessária e, invariavelmente, irá nos retardar.