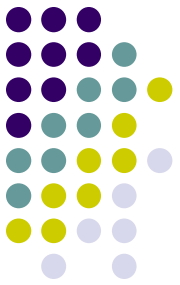


Engenharia de Software I

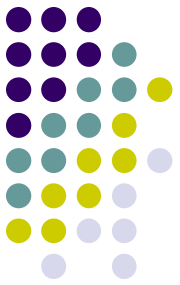
Prof. Ricardo Satoshi





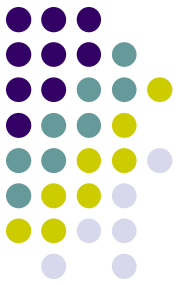
Plano de Ensino

- Justificativa
 - A disciplina visa proporcionar ao aluno conceitos básicos sobre Engenharia de Software que possibilitem aplicar um processo de desenvolvimento de software utilizando tecnologia Orientada a Objetos.



Plano de Ensino

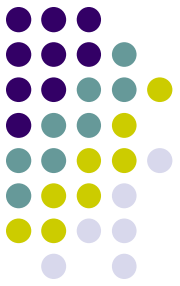
- Objetivos
 - Conhecer e utilizar os princípios da Engenharia de Software no processo de desenvolvimento de aplicações.
 - Aplicar os conceitos da Análise e Projeto em um caso prático
 - Utilizar a notação UML para a modelagem de aplicações orientadas a objetos.



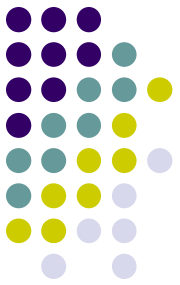
Bibliografia

- Básica:
 - BEZERRA, Eduardo, Princípios de Análise e Projeto de Sistemas com UML, Elsevier, 2003.
 - MELO, Ana Cristina, Desenvolvendo Aplicações com UML: Do Conceitual à Implementação. Rio de Janeiro; Editora Brasport., 2002.

Bibliografia

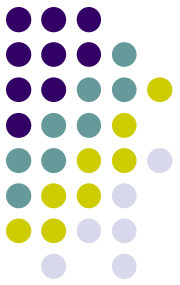


- Complementar
 - PRESSMAN, R. S. Engenharia de Software. São Paulo, Makron Books, 1995.
 - KRUCHTEN, P.; Introdução ao RUP, Ciência Moderna, 2003.
 - SOMMERVILLE.; Engenharia de Software, Person, 8º Edição
- Software de Apoio:
 - Ferramenta Case



O Produto

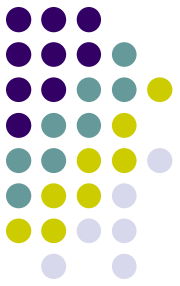
- Dois papéis do Software:
 - Produto
 - quando libera o potencial embutido no hardware
 - Veículo para liberar produto:
 - quando o produto atua:
 - (a) como uma base para o controle do computador – sistema operacional;
 - (b) para a comunicação de informação - redes;
 - (c) para criar outros produtos – Java, Delphi, VB)



Evolução do Software

(1950 - 1965)

- O hardware sofreu contínuas mudanças
- O software era uma arte "secundária" para a qual havia poucos métodos sistemáticos
- O hardware era de propósito geral
- O software era específico para cada aplicação
- Não havia documentação

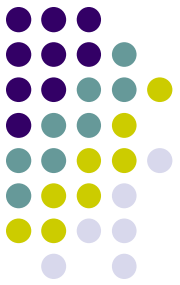


Evolução do Software

(1965 - 1975)

- Multiprogramação e sistemas multiusuários
- Sistemas de tempo real
- 1a geração de SGBD's
- Produtos de software --> software houses
- Bibliotecas de Software
- Cresce o número de sistemas baseado em computador
- Manutenção quase impossível

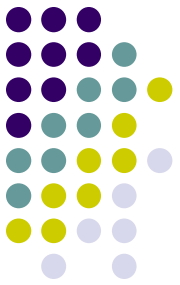
..... CRISE DE SOFTWARE



Evolução do Software

(1975 - hoje)

- Sistemas distribuídos
- Redes locais e globais
- Uso generalizado de microprocessadores - produtos inteligentes
- Hardware de baixo custo
- Impacto de consumo



Evolução do Software

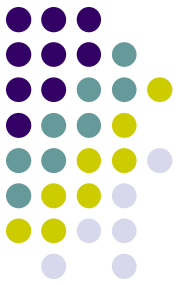
(Quarta era do software: atualidade)

- Tecnologias orientadas o objetos
- Sistemas especialistas e software de inteligência artificial usados na prática
- Computação Paralela
- Internet
- Realidade Virtual
- Programação para dispositivos móveis

Problemas que Persistem e se Intensificam



- 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

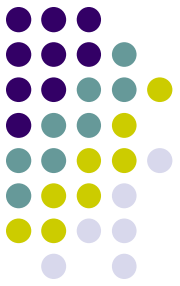


Uma Perspectiva Industrial

- 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 ?

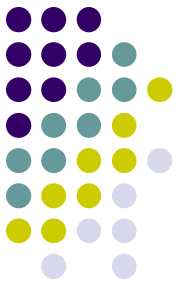
A preocupação em resolver essas questões tem levado à adoção das práticas da Engenharia de Software

Competitividade do Software

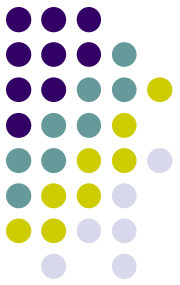


- Hoje o software é um negócio competitivo
- Os principais direcionadores que propiciarão uma intensa competição na área de software são:
 - custo
 - adequação de prazo e
 - qualidade
- Intensifica-se, portanto, uma rápida movimentação dos desenvolvedores para adotar práticas modernas de Engenharia de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

Características do Software



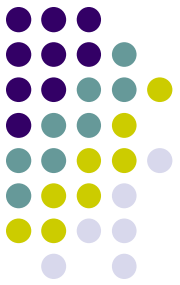
1. desenvolvido ou projetado por engenharia, não manufaturado no sentido clássico
2. não se desgasta mas se deteriora
3. a maioria é feita sob medida em vez de ser montada a partir de componentes existentes



Crise de Software

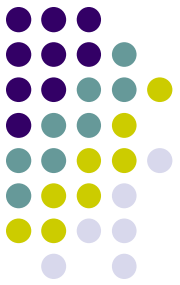
- Refere-se a um conjunto de problemas encontrados no desenvolvimento de software:
 - (1) As estimativas de prazo e de custo freqüentemente são imprecisas
 - “Não dedicamos tempo para coletar dados sobre o processo de desenvolvimento de software”
 - “Sem nenhuma indicação sólida de produtividade, não podemos avaliar com precisão a eficácia de novas ferramentas, métodos ou padrões”

Crise de Software



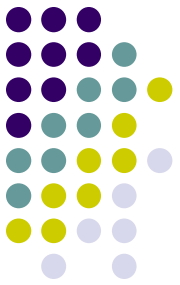
(2) A produtividade das pessoas da área de software não tem acompanhado a demanda por seus serviços

- “Os projetos de desenvolvimento de software normalmente são efetuados apenas com um vago indício das exigências do cliente”



Crise de Software

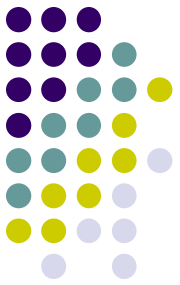
- (3) A qualidade de software às vezes é menos que adequada
- Só recentemente começaram a surgir conceitos quantitativos sólidos de garantia de qualidade de software



Crise de Software

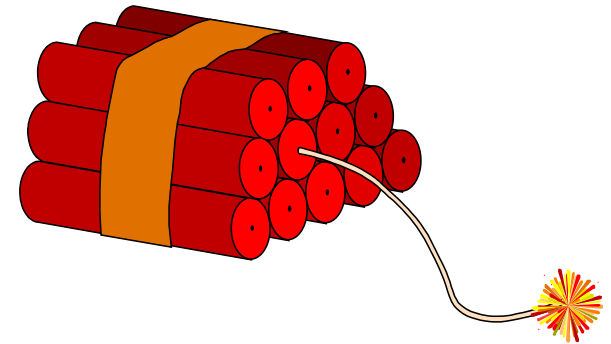
(4) O software existente é muito difícil de manter

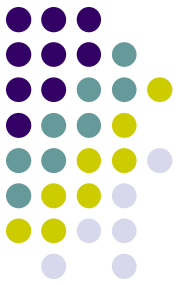
- A tarefa de manutenção devora o orçamento destinado ao software
- A facilidade de manutenção não foi enfatizada como um critério importante



Crise de Software

- Estimativas de prazo e de custo ↑
- Produtividade das pessoas ↓
- Qualidade de software ↓
- Software difícil de manter ↑

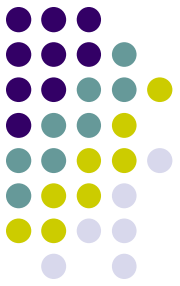




O Processo

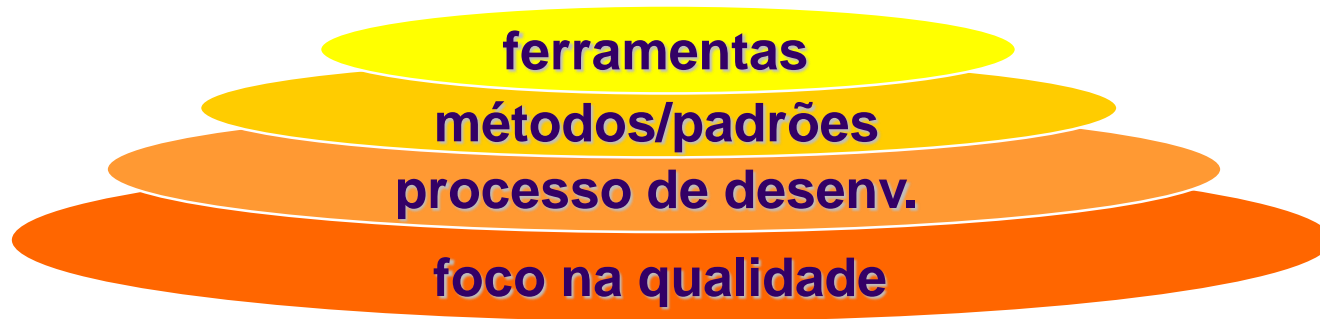
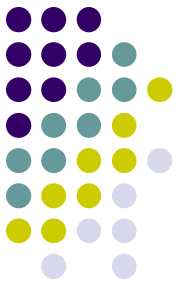
- É um conjunto das tarefas requeridas para construir software de alta qualidade
- “Processo” = Engenharia de Software ?
- Processo define a abordagem (Estruturada, Orientado a Objetos etc.)
- Engenharia de Software engloba também as tecnologias, como métodos e ferramentas

Engenharia de Software: Uma tecnologia em Camadas

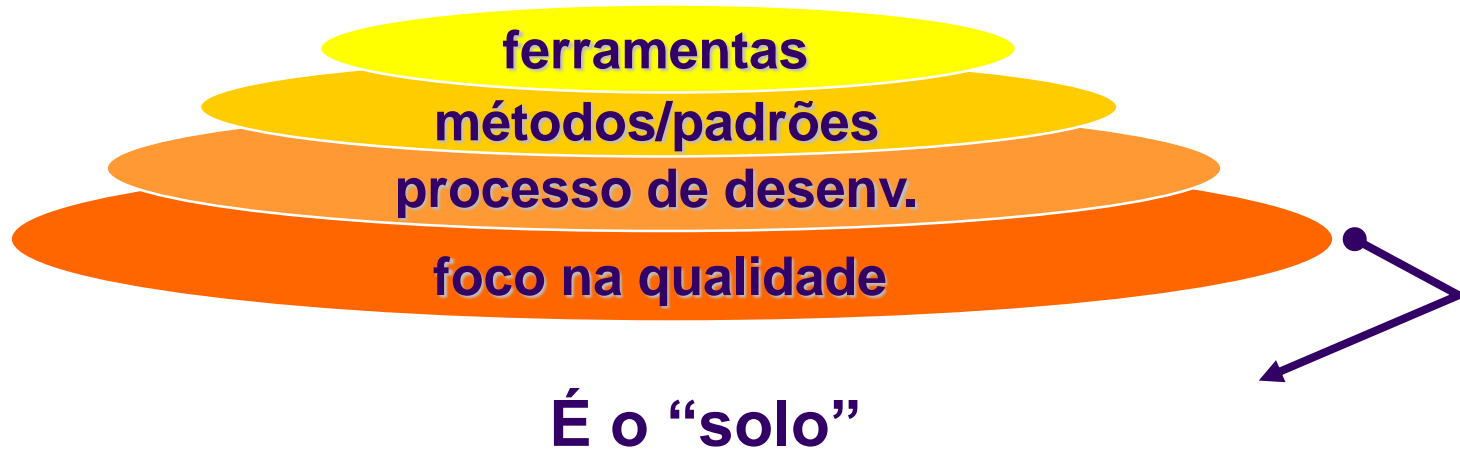
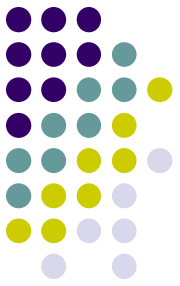


- Definição [IEEE]:
 - (1) aplicação de uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software, ou seja, a aplicação da engenharia ao software
 - (2) o estudo de abordagens do tipo declarado em (1)

Engenharia de Software: Uma tecnologia em Camadas

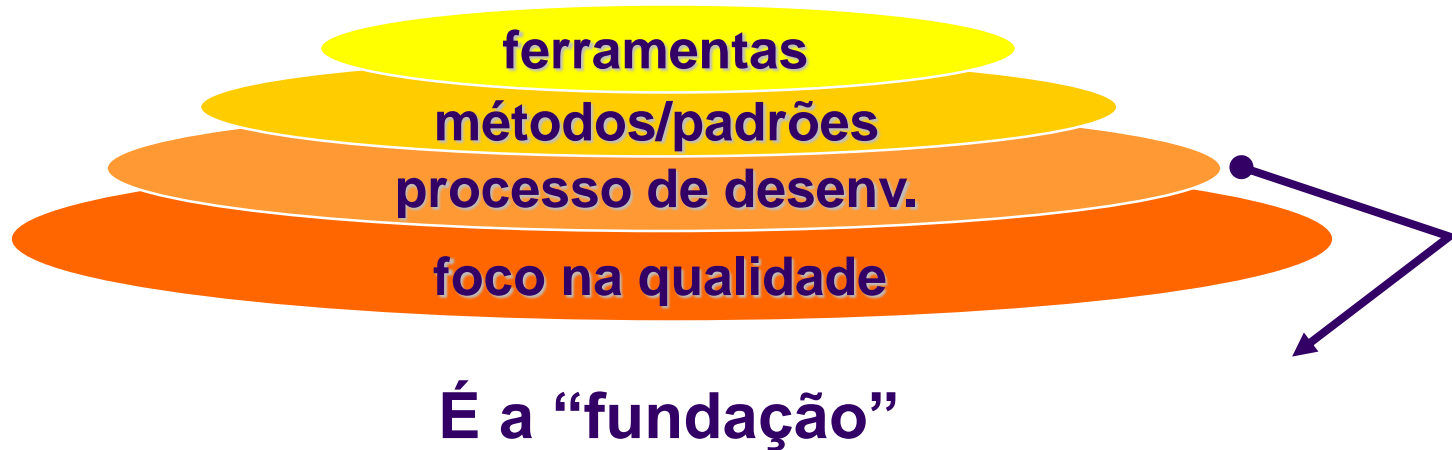
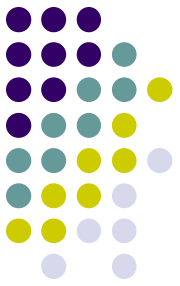


Engenharia de Software: Uma tecnologia em Camadas



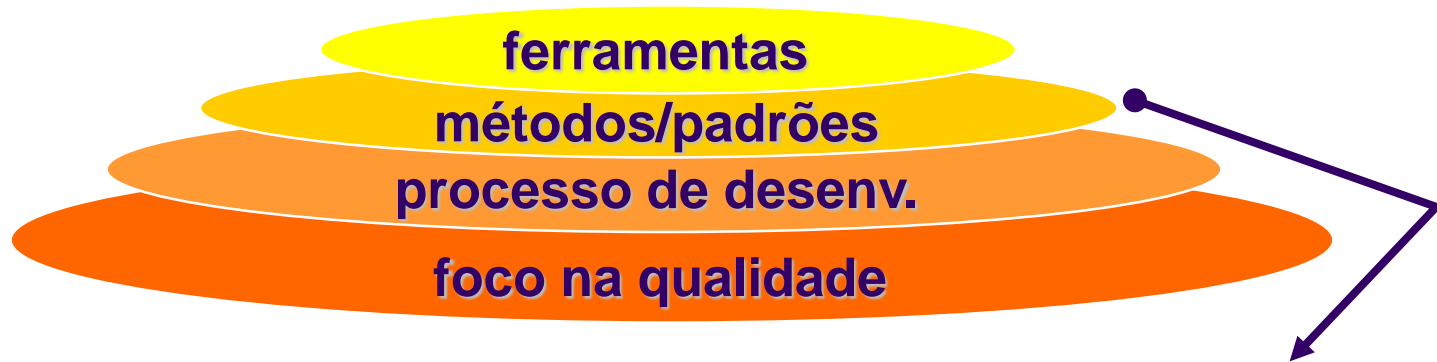
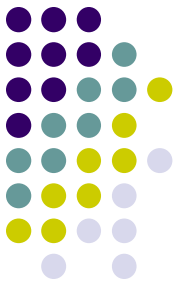
Gerenciamento da Qualidade Total e filosofias similares produzem uma mudança cultural que permite o desenvolvimento crescente de abordagens mais maduras para a Engenharia de Software

Engenharia de Software: Uma tecnologia em Camadas



É a cola que gruda as camadas de tecnologias e permite um desenvolvimento de software racional e em tempo; define um conjunto de áreas chave do processo (KPA) que deve ser estabelecido para um uso efetivo da Engenharia de Software

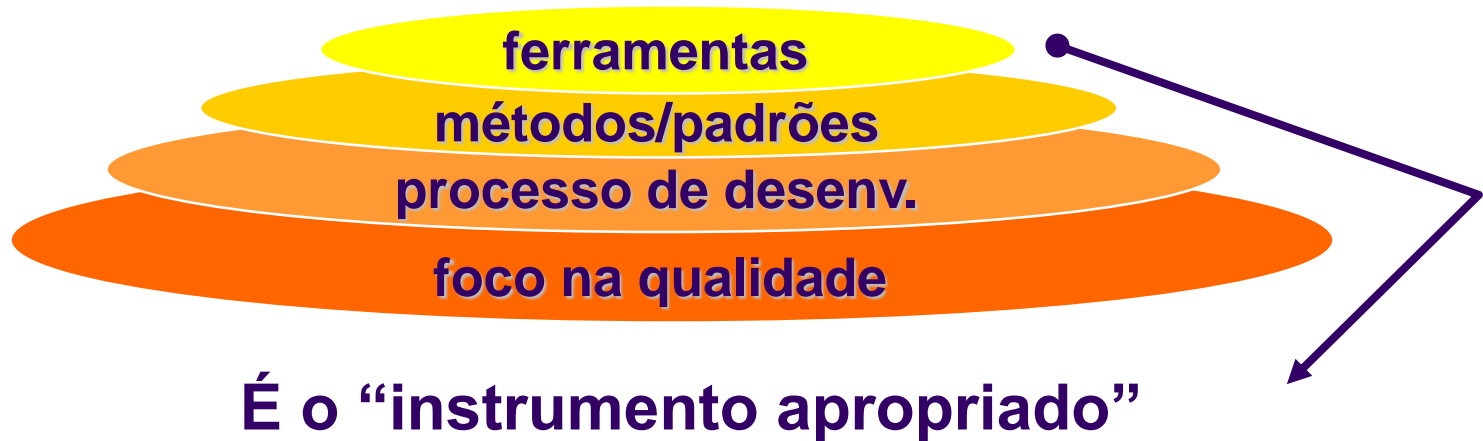
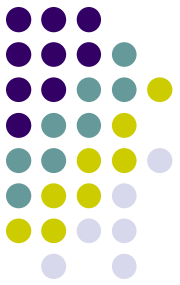
Engenharia de Software: Uma tecnologia em Camadas



É o “como fazer”

Englobam um conjunto de tarefas que inclui análise de requisitos, projeto, construção de programas, teste e manutenção

Engenharia de Software: Uma tecnologia em Camadas



Dão suporte automatizado ou semi-automatizado ao processo e aos métodos; quando as ferramentas se integram tem-se um sistema denominado CASE (Computer Aided Software Engineering)

Engenharia de Software



Uma Visão Genérica: 3 Fases

- **Definição - “o que”**

- Engenharia do Sistema
- Análise de Requisitos

- **Desenvolvimento - “como”**

- Projeto
- Geração do Código
- Teste

- **Manutenção**

- **Atividades Guarda-Chuva**

- Controle e Rastreamento do Projeto
- Revisões Técnicas Formais
- Garantia de Qualidade
- Gerenciamento de Configuração (controle de versões)
- Produção e Preparação de Documentos
- Gerenciamento de Reusabilidade
- Medição