

Uma Introdução a Engenharia de Software e Sistemas

Centro de Informática - Universidade Federal de Pernambuco
Engenharia da Computação

Kiev Gama

kiev@cin.ufpe.br

Slides originais elaborados por Ian Sommerville e adaptado pelos professores Marcio Cornélio, Vinicius Garcia e Kiev Gama

O autor permite o uso e a modificação dos *slides* para fins didáticos



UNIVERSIDADE FEDERAL DE PERNAMBUCO

Engenharia de software

- As economias de TODAS as nações desenvolvidas são dependentes de software.
- Cada vez mais sistemas são controlados por software.
- A engenharia de software se dedica às teorias, métodos e ferramentas para desenvolvimento de software profissional
 - Sistemas **não-triviais**
 - Com base em um conjunto de **requisitos**

Custos de software

- Os custos de software **dominam os custos** de sistemas computacionais.
 - Em geral, software custa mais que hardware
- Manter um software custa mais que desenvolvê-lo
- A engenharia de software dedica-se ao desenvolvimento de software com custos adequados
 - **Respeitando** o cronograma acordado
 - **Satisfazendo** as necessidades dos clientes
 - **Minimizando** o custo de manutenção

Definições

- Disciplina gerencial e tecnológica que lida com a produção e manutenção sistemática de produtos de software desenvolvidos dentro de estimativas de custo e tempo
- “Methods for the Multi-person construction of multi-version software”(Parnas)

FAQs sobre engenharia de software

- O que é software?
- O que é engenharia de software?
- Qual é a diferença entre engenharia de software e ciência da computação?
- Qual é a diferença entre engenharia de software e engenharia de sistemas?
- O que é processo de software?
- O que é um modelo de processo de software?

FAQs sobre engenharia de software

- Quais são os custos da engenharia de software?
- O que é CASE (*Computer-Aided Software Engineering*)?
- Quais são os atributos de um bom software?
- Quais são os desafios-chave enfrentados pela engenharia de software?

O que é software?

- Programas de computador e artefatos associados
- Produtos de software podem ser:
 - **Genéricos** – desenvolvidos para serem vendidos para uma grande variedade de clientes (e.g., Excel e Word)
 - **Personalizados** – desenvolvidos para um único cliente de acordo com as suas especificações.
- Um software novo pode ser criado através de
 - desenvolvimento de novos programas;
 - configuração de sistemas de software genéricos; ou
 - reutilização de um software existente.

Tipos de Aplicação

- Stand-alone
- Interativas baseadas em transações
- Sistemas de controle embarcados
- Sistemas de processamento em batch
- Sistemas de entretenimento
- Sistemas de modelagem e simulação
- Sistemas de coleta de dados
- Sistemas de sistemas

O que é engenharia de software?

- Engenharia de software é uma disciplina relacionada com todos os aspectos da produção de software.
- ... e propõe ferramentas, técnicas e processos para:
 - Entender com precisão qual é o problema (as necessidades associadas ao sistema que deve ser construído/modificado)
 - Produzir uma solução adequada para esse problema (um sistema pronto para usar, levando-se em consideração as necessidades das partes interessadas)
 - Levando-se em conta restrições de desenvolvimento e recursos disponíveis

Qual é a diferença entre engenharia de software e ciência da computação?

- A ciência da computação dedica-se à teoria e aos fundamentos;
 - Engenharia de software dedica-se aos aspectos práticos de desenvolvimento e de entrega de software
 - “O que” vs. “Como”
- Teorias de ciência da computação são ainda insuficientes para atuar como uma base completa para a engenharia de software (diferente de, por exemplo, física e engenharia elétrica).
 - Em outras palavras: **não há receitas prontas!**

Qual é a diferença entre engenharia de software e engenharia de sistemas?

- Engenharia de sistemas:
 - Mais ampla
 - Muita ênfase em aspectos de hardware e infraestrutura
 - Abstração do hardware
 - Organização física das partes do sistema
 - Aspectos de comunicação
 - **Engloba** a engenharia do software
- Os engenheiros de sistema estão envolvidos em diversas atividades da engenharia de software
 - Projeto da arquitetura
 - Elicitação e especificação de requisitos

O que é processo de software?

- Um conjunto estruturado de atividades, práticas, artefatos e ferramentas necessários para o desenvolvimento de um sistema de software
 - Especificação;
 - Desenvolvimento;
 - Validação;
 - Evolução.
- Exemplos: Processo Unificado (RUP), Programação Extrema, UML Components
- Diferente da definição do livro!!!
 - “processo”, “modelo de processo”, “método” e “metodologia”

O que é processo de software?

- Alguns elementos de um processo:
 - Modelos de sistema:
 - Modelos gráficos que podem/devem ser produzidos e as notações que devem ser empregadas;
 - Restrições aplicadas aos modelos de sistema;
 - Recomendações de boas práticas de projeto;
 - Atividades que devem ser seguidas em determinada ordem
 - Às vezes também prescrevem ferramentas
- Um processo adere a um ou mais modelos de processo

O que é um modelo de processo de software?

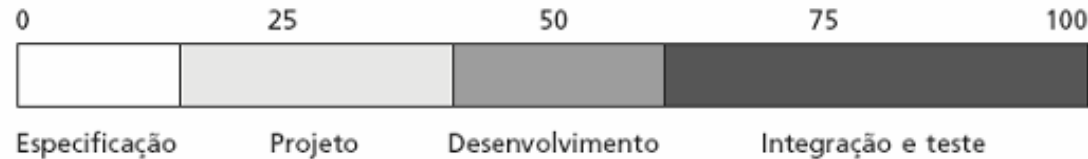
- Uma representação simplificada de um processo de software, apresentado sob uma perspectiva específica
 - Incluindo algumas atividades e sua organização de alto nível
- Modelos gerais de processo
 - Cascata;
 - Desenvolvimento iterativo;
 - Engenharia de software baseada em componentes.
- Dois modelos não são necessariamente **mutuamente excludentes!**
- Representações de modelos de processo:
 - Modelo de *workflow* – seqüência de atividades;
 - Modelo de fluxo de dados – fluxo de informações;
 - Modelo de papel/ação – quem faz o quê.

Quais são os custos da engenharia de software?

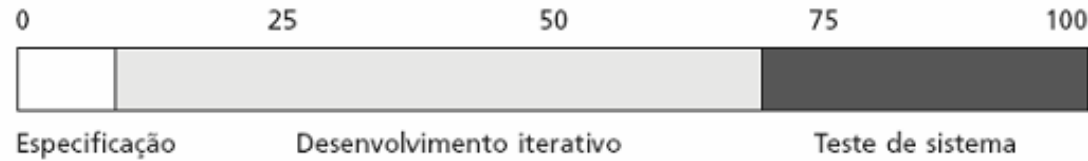
- 60% dos custos são custos de desenvolvimento
- 40% são custos de testes
- Para software sob encomenda, os custos de evolução normalmente excedem os de desenvolvimento.
- Os custos variam dependendo do tipo de sistema que está sendo desenvolvido e dos requisitos do sistema, tais como **desempenho** e **confiabilidade**
- A distribuição de custos depende do modelo de desenvolvimento que é usado.

Distribuição de custos nas atividades

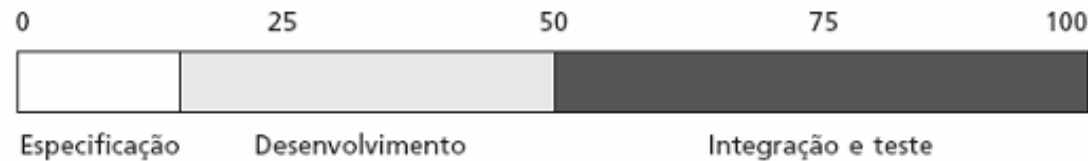
Modelo cascata



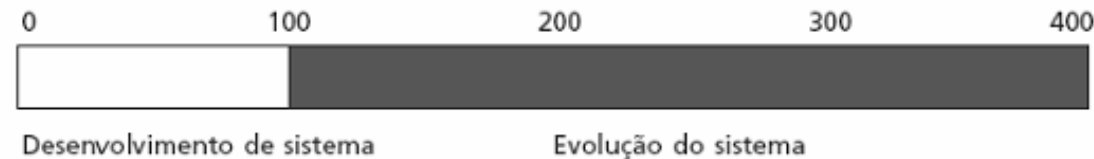
Desenvolvimento iterativo



Engenharia de software baseada em componentes



Custos de desenvolvimento e evolução ao longo da vida do software



O que é CASE (*Computer-Aided Software Engineering*)?

- Sistemas de software que se destinam a fornecer apoio automatizado para as atividades de desenvolvimento de software.
- Sistemas CASE são usados freqüentemente para apoiar um método específico
- *Upper-CASE*
 - Ferramentas para apoiar as atividades iniciais de processo de requisitos e de projeto;
- *Lower-CASE*
 - Ferramentas para apoiar as atividades finais tais como programação, *debugging* e teste.

Quais são os atributos de um bom software?

- O software deve fornecer a funcionalidade e o desempenho requeridos para o usuário e deve apresentar certas características
- Facilidade de manutenção
 - Deve ser fácil e barato fazer com que o sistema, depois de implantado, evolua para atender às necessidades dos clientes
- Confiabilidade
 - O software deve funcionar de maneira que o usuário possa depender dele. No caso de falha, não traz danos físicos ou econômicos
- Eficiência
 - O software deve usar o mínimo de recursos e prover máxima funcionalidade
- Usabilidade
 - O software deve ser compreensível e fácil de usar

Quais são os desafios-chave enfrentados pela engenharia de software?

- Heterogeneidade
 - Sistemas de software devem ser capaz de lidar com diferentes plataformas de hardware e ambientes de execução;
- Entrega
 - O sistema deve ser entregue ao cliente no menor tempo possível, com o menor custo possível;
- Confiança
 - O usuário deve poder justificadamente depositar sua confiança no sistema
- Escala
 - O sistema deve funcionar adequadamente mesmo quando um grande número de usuários o está usando

Diversidade da Engenharia de Software

- Há vários tipos de sistemas de software, e não existe um conjunto universal de regras que seja aplicável para todos eles
- Os métodos e ferramentas de E.S. a serem utilizados dependem do tipo de aplicação a ser desenvolvida, dos requisitos do cliente e da experiência da equipe de desenvolvimento.

Fundamentos da Engenharia de Software

- Alguns princípios fundamentais se aplicam a todos os tipos de sistemas de software, independente das técnicas a serem empregadas:
 - Sistemas devem ser desenvolvidos utilizando um processo de desenvolvimento gerenciado e compreendido. Obviamente, diferentes processos são usados para diferentes tipos de software.
 - “Dependabilidade” (Dependability) e performance são importantes para todos os tipos de sistema.
 - Entender e gerenciar a especificação do software e requisitos (o que o sistema deve fazer) é importante.
 - Quando adequado, deve-se reutilizar software que já foi desenvolvido ao invés de escrever um novo software.

Responsabilidade profissional e ética

- A engenharia de software envolve responsabilidades mais amplas do que simplesmente a aplicação de habilidades técnicas.
- Os engenheiros de software devem se comportar de modo honesto e eticamente responsável para serem respeitados como profissionais.
- O comportamento ético é mais do que simplesmente a sustentação de leis.

Questões de responsabilidade profissional

- Confidencialidade
 - Os engenheiros de software devem normalmente respeitar a confidencialidade de seus funcionários ou clientes, independentemente de ter ou não assinado um acordo formal.
 - Caso não aceitem essas condições, devem deixar isso explícito para seus contratantes
- Competência
 - Os engenheiros não devem conscientemente aceitar um trabalho que esteja fora de sua competência.

Questões de responsabilidade profissional

- Direitos sobre propriedade intelectual
 - Desenvolvedores devem estar cientes das leis locais que regem o uso de propriedade intelectual, tais como patentes, direitos autorais, etc.
 - Eles devem tomar cuidado para assegurar que a propriedade intelectual dos funcionários e clientes seja protegida.

Dilemas éticos

- Discordância, em princípio, das políticas da gerência sênior.
- Um funcionário age de uma forma não ética e libera um sistema de segurança crítico sem finalizar o teste do sistema.
- Participação no desenvolvimento de sistemas de **armamentos militares** ou de **sistemas nucleares**

Software Engineering Body of Knowledge

- SWEBOK – Padrão internacional publicado pelo IEEE - www.swebok.org/
- Guia contendo “tudo” o que é aceito como parte do “corpo de conhecimento” necessário para a engenharia de software
- Dividido em áreas de Conhecimento (Knowledge Areas – KA)
 - Software requirements
 - Software design
 - Software construction
 - Software testing
 - Software maintenance
 - Software configuration management
 - Software engineering management (Engineering management)
 - Software engineering process
 - Software engineering models and methods
 - Software quality
 - Software engineering professional practice
 - Software engineering economics
 - Computing foundations
 - Mathematical foundations
 - Engineering foundations

Bibliografia

- Sommerville, Ian. Engenharia de Software, 9ª edição. Pearson Education