

Uma Introdução à Engenharia de Software

Objetivos

- Apresentar a engenharia de software e explicar a sua importância
- Dirigir as respostas às questões-chave sobre engenharia de software
- Apresentar questões éticas e profissionais e explicar por que elas são assunto para engenheiros de software

Tópicos abordados

- FAQs sobre engenharia de software
- Responsabilidade profissional e ética

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
- Os dispêndios com software representam uma fração significativa do PIB em todos os países desenvolvidos.

Custos de software

- Os custos de software dominam os custos de sistemas computacionais. Em um PC, os custos de software são freqüentemente maiores que o custo do hardware.
- Manter um software custa mais que desenvolvê-lo. Para sistemas com uma longa vida, os custos de manutenção podem ser muito maiores que os custos de desenvolvimento.
- 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

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?
- Quais são os métodos 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, por exemplo, softwares para PC, tais como Excel e Word.
 - Personalizados – desenvolvidos para um único cliente de acordo com as suas especificações.
- Um software novo pode ser criado através do desenvolvimento de novos programas, da configuração de sistemas de software genéricos ou da reutilização de um software existente.

O que é engenharia de software?

- Engenharia de software é uma disciplina de engenharia relacionada com todos os aspectos da produção de software.
- Engenheiros de software devem - dependendo do problema a ser resolvido, das **restrições de desenvolvimento** e dos **recursos disponíveis** - adotar uma abordagem sistemática e organizada para seu trabalho, além de usar **ferramentas** e **técnicas** apropriadas.

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

- A ciência da computação dedica-se à teoria e aos fundamentos; já a engenharia de software dedica-se aos aspectos práticos de desenvolvimento e de entrega de software para o uso.
- 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?

- A engenharia de sistemas dedica-se aos aspectos de desenvolvimento de sistemas baseados em computador, incluindo:
 - a engenharia de hardware
 - de software e
 - de processo.
- A engenharia de software é a parte desse processo que se dedica ao desenvolvimento da infra-estrutura do software
- Os engenheiros de sistema estão envolvidos na especificação, no projeto de arquitetura e na integração e implantação do sistema.

O que é processo de software?

- É um conjunto de atividades cuja meta é o desenvolvimento ou evolução de sistemas de software.
- As atividades comuns à maioria dos processos de software são:
 - Especificação – o que o sistema deve fazer e suas restrições de desenvolvimento.
 - Desenvolvimento – produção do sistema de software.
 - Validação – o software é o que o cliente deseja?
 - Evolução – mudança do software em resposta às demandas de mudança.
- Há uma confusão generalizada entre os termos "**processo**", "**modelo de processo**", "**método**" e "**metodologia**".
 - Ex: Processo Unificado (RUP)
 - Usaremos a definição que está nos slides da terceira aula (**Capítulo 4**)!

O que é um modelo de processo de software?

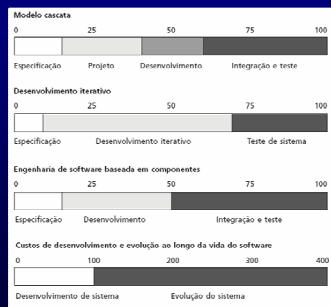
- Uma representação simplificada de um processo de software, apresentado sob uma perspectiva específica.
- Exemplos de modelos de processo são:
 - Modelo de *workflow* – sequência de atividades;
 - Modelo de fluxo de dados – fluxo de informações;
 - Modelo de papel/ação – quem faz o quê.
- Modelos gerais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

Quais são os custos da engenharia de software?

- Aproximadamente 60% dos custos são custos de desenvolvimento e 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

Figura 1.1
Distribuição de custos nas atividades de engenharia de software



Quais são os métodos de engenharia de software?

- Abordagens estruturadas para desenvolvimento de software que incluem modelos de sistema, notações, regras, recomendações de projeto e guias de processo.
- Descrições de modelo de sistema:
 - Descrições de modelos gráficos que podem/devem ser produzidos;
- Regras
 - Restrições aplicadas aos modelos de sistema;
- Recomendações:
 - Recomendações de boas práticas de projeto;
- Guia de processo:
 - Quais atividades devem ser seguidas
- Às vezes também prescrevem ferramentas

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

- Sistemas de software que se destinam a fornecer apoio automatizado para as atividades de processo de software.
- Sistemas CASE são usados frequentemente para apoio ao método.
- *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 ser fácil de manter, confiável e aceitável.
- Facilidade de manutenção
 - Deve ser fácil e barato fazer com que o sistema, depois de implantado, evolua para atender às necessidades dos clientes;
- Confiança
 - O software deve funcionar conforme sua especificação;
- Eficiência
 - O software deve usar o mínimo de recursos e prover máxima funcionalidade
- Usabilidade
 - O software deve ser compreensível, usável e compatível com outros sistemas.

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

- Heterogeneidade
 - Sistemas de software devem ser capazes 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

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 desvirtuar o seu nível de competência. Eles 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.

Código de Ética do ACM/IEEE

- As sociedades profissionais nos EUA têm cooperado para produzir uma prática de código de ética.
- Os membros destas organizações assinam o código de prática quando eles se inscrevem como membros.
- O Código contém oito Princípios relacionados ao comportamento e às decisões tomadas pelos engenheiros de software profissionais, incluindo praticantes, educadores, gerentes, supervisores e responsáveis pela criação de políticas, assim como estagiários e estudantes da profissão.

Código de ética - preâmbulo

- Preâmbulo
 - A versão resumida do código apresenta as aspirações no nível alto de abstração: as cláusulas que estão incluídas na versão completa fornecem exemplos e detalhes de como essas aspirações mudam a maneira que agimos como profissionais de engenharia de software. Sem essas aspirações, os detalhes podem se tornar formais e tediosos; sem os detalhes, as aspirações podem se tornar aparentemente importantes, mas vazias; juntos, aspirações e detalhes formam um código coeso.
 - Os engenheiros de software devem se comprometer a fazer da análise, especificação, desenvolvimento, teste e manutenção de software uma profissão benéfica e respeitada. De acordo com o seu comprometimento com a saúde, segurança e bem-estar do público, os engenheiros de software devem aderir aos seguintes Oito Princípios:

Código de ética - princípios

- PÚBLICO
 - Os engenheiros de software devem agir consistentemente com o interesse público.
- CLIENTE E EMPREGADOR
 - Os engenheiros de software devem agir dentro dos melhores interesses do seu cliente e empregador, de forma consistente com o interesse público.
- PRODUTO
 - Os engenheiros de software devem assegurar que seus produtos e as modificações a eles relacionadas atendam aos mais altos padrões profissionais possíveis.

Código de ética - princípios

- JULGAMENTO
 - Os engenheiros de software devem manter integridade e independência no seu julgamento profissional.
- GERENCIAMENTO
 - Os gerentes e líderes de engenharia de software devem contribuir e promover uma abordagem ética para o gerenciamento de desenvolvimento e manutenção de software.
- PROFISSÃO
 - Os engenheiros de software devem promover a integridade e a reputação da profissão de forma consistente com o interesse público.

Código de ética - princípios

- COLEGAS
 - Os engenheiros de software devem se honestos e colaborativos com seus colegas.
- INDIVÍDUO
 - Os engenheiros de software devem participar, ao longo da vida, aprendendo, respeitando e promovendo uma abordagem ética na prática da profissão.

Dilemas éticos

- Discordância, em princípio, das políticas da gerência sênior.
- Seu 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.

Pontos-chave

- A engenharia de software é está relacionada a todos os aspectos da produção de software.
- Os produtos de software consistem em programas desenvolvidos e documentação associada.
- O processo de software compreende todas as atividades envolvidas no desenvolvimento de produtos de software.
 - As atividades básicas são especificação, desenvolvimento, validação e evolução de software.
- Métodos são meios organizados de produção de software. Eles incluem sugestões para o processo a ser seguido, as notações a ser usadas, modelos de sistemas a ser desenvolvidos, regras que regem estes modelos e diretrizes para o projeto.

Pontos-chave

- Ferramentas CASE são sistemas de software projetados para apoiar as atividades rotineiras no processo de software, tais como:
 - edição de diagramas de projeto
 - verificação da consistência de diagramas
 - automatização de testes
 - depuração
- Engenheiros de software têm responsabilidades com a profissão de engenharia e a sociedade. Eles não devem se preocupar apenas com assuntos técnicos.
- Sociedades profissionais publicam códigos de conduta que definem os padrões de comportamento esperados de seus membros.