

# 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 frequentemente 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.

## 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 documentação associada, tais como requisitos, modelos de projetos e manuais de usuário.
- Produtos de *software* podem ser desenvolvidos para um cliente particular ou para um mercado geral.
- 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

## 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 parte desse processo que se dedica ao desenvolvimento da infra-estrutura do *software*, controle, aplicações e banco de dados no sistema.
- 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 *software*.
- As atividades genéricas em todos os processos de *software* são:
  - Especificação – o que o sistema deve fazer e suas restrições de desenvolvimento.
  - Desenvolvimento – produção do sistema de *software*.
  - Validação – verificação de que o *software* é o que o cliente deseja.
  - Evolução – mudança do *software* em resposta às demandas de mudança.

## O que é um modelo de processo de *software*?

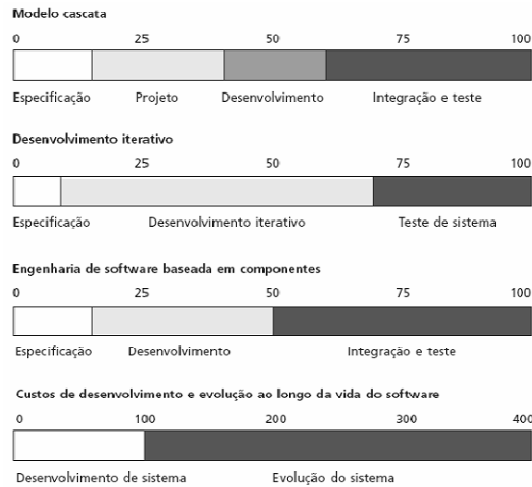
- Uma representação simplificada de um processo de *software*, apresentado sob uma perspectiva específica.
- Exemplos de modelos de processo são:
  - Modelo de *workflow* – sequência de atividades;
  - Modelo de fluxo de dados – fluxo de informações;
  - Modelo de papel/ação – quem faz o quê.
- Modelos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

## 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 de desenvolvimento.
- Os custos variam dependendo do tipo de sistema que está sendo desenvolvido e dos requisitos de atributos de sistema, tais como desempenho e confiabilidade.
- A distribuição de custos depende do modelo de desenvolvimento que é usado.

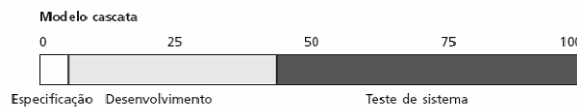
## Distribuição de custos nas atividades

**Figura 1.1**  
Distribuição de custos nas atividades de engenharia de software



## Custos de desenvolvimento de produto

**Figura 1.2**  
Custos de desenvolvimento do produto



## 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 guia de processo.
- Descrições de modelo de sistema:
  - Descrições de modelos gráficos que 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.

## 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 manutenível, confiável e aceitável.

Facilidade de manutenção

- O *software* deve evoluir para atender às necessidades de mudança;

Confiança

- O *software* deve ser confiável;

Eficiência

- O *software* não deve desperdiçar os recursos do sistema;

Usabilidade

- O *software* deve ser aceito pelos usuários para o qual foi projetado.
- Isso significa que ele deve ser compreensível, usável e compatível com outros sistemas.

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

- Heterogeneidade, entrega e confiança.
- Heterogeneidade
  - Técnicas de desenvolvimento para construção de *software* que podem lidar com plataformas heterogêneas e ambientes de execução;
- Entrega
  - Técnicas de desenvolvimento para conduzir a entrega mais rápida de *software*;
- Confiança
  - Técnicas de desenvolvimento que mostram que o *software* pode ter a confiança dos seus usuários.

## 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.

## 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 uma disciplina de engenharia relacionada com todos os aspectos de produção de *software*.
- Os produtos de *software* consistem em programas desenvolvidos e documentação associada.
  - Os atributos essenciais do produto são: manutenibilidade, confiança, eficiência e aceitabilidade.
- 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*.
  - incluem sugestões para o processo a ser seguido, as notações a serem usadas, modelos de sistemas a serem 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 e rastreabilidade de testes de programa realizados.
- 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.