

# ENGENHARIA DE SOFTWARE

**Carga Horária 60h**

**Prof. André Luiz de Castro Leal**

## ↪ Ementa:

### Introdução

- I Processo de desenvolvimento de software.
- II Modelos de processo de desenvolvimento de software.
- III Qualidade em software.
- IV Planejamento e Gerência de projetos de software.
- V Métricas.
- VI Análise de riscos.
- VII Engenharia de Requisitos.
- VIII Métodos e Técnicas de análise e projeto de software.
- IX Verificação, validade e teste de software.
- X Gerência de Configuração.
- XI Manutenção.
- XII Documentação.
- XIII Ferramentas e ambientes de software.

## ↪ Introdução:

As 10 Áreas da Engenharia de Software, Conforme o SWEBOK  
*Software Engineering Body of Knowledge*

[http://pt.wikipedia.org/wiki/Software\\_Engineering\\_Body\\_of\\_Knowledge](http://pt.wikipedia.org/wiki/Software_Engineering_Body_of_Knowledge)

Características de uma Engenharia (SWEBOK):

- Educação de iniciação profissional validada e legitimada pela sociedade;
- Registro da adequação à prática através de certificação voluntária ou licenciamento compulsório;
- Desenvolvimento de habilidades especializadas e educação profissional continuada;
- Suporte através de sociedades profissionais
- Compromisso a normas de conduta freqüentemente estabelecidas em um código de ética.

## ↪ Introdução:

### Origens do Corpo de Conhecimentos da Engenharia de Software

- Matemática
- Ciência da Computação
- Administração de Projetos
- Ciências Cognitivas e Fatores Humanos
- Engenharia de Computadores
- Administração e Ciências Administrativas
- Engenharia de Sistemas

## ↪ Introdução:

Conceitos Fundamentais de Engenharia de Software:

- Abstração
- Métodos e Notações de Análise e Design
- Prototipação de Interface com Usuário
- Modularidade e Arquitetura
- Ciclo de Vida e Processo de Software
- Reuso
- Métricas
- Suporte Automatizado

## ↪ Introdução:

10 Áreas da Engenharia de Software SWEBOK 2004

- 1) Gerência de Configuração de Software:** Identifica a configuração do sistema (características documentadas do hardware e software que o compõem) em pontos discretos no tempo, de modo a controlar sistematicamente suas mudanças e manter sua integridade e rastreabilidade durante o ciclo de vida do sistema.
- 2) Gerência de Engenharia de Software:** Gerencia projetos de desenvolvimento de software.
- 3) Processo de Engenharia de Software:** Define, implementa, mede, gerencia, modifica e aperfeiçoa o processo de desenvolvimento de software.
- 4) Ferramentas e Métodos:** Ferramentas de software automatizam o processo de engenharia de software. Métodos impõem estrutura sobre a atividade de desenvolvimento e manutenção de software com o objetivo de torná-la sistemática e mais propensa ao sucesso.
- 5) Qualidade de Software:** Conjunto de atividades relacionadas com garantia de qualidade de software, entre estas as atividades de verificação e validação.

## ↪ Introdução:

### 10 Áreas da Engenharia de Software

**6) Requisitos de software:** Aquisição, análise, especificação e gestão de requisitos de software.

**7) Design de software:** Transformação de requisitos (de software), tipicamente estabelecidos em termos relevantes ao domínio do problema, em uma descrição explicando como solucionar os aspectos do problema relacionados com software

**8) Construção de Software:** Construção de programas funcionais e coerentes através da codificação, autovalidação, e teste unitário.

**9) Teste de Software:** Verificação dinâmica do comportamento do programa através do uso de um conjunto finito de casos de teste – adequadamente selecionados de um domínio de execuções usualmente infinito - contra o comportamento esperado deste

**10) Manutenção de Software:** Atividades de suporte custo-efetivo a um sistema de software, que pode ocorrer antes e após a entrega do software. Após a entrega do software são feitas modificações com o objetivo de corrigir falhas, melhorar seu desempenho ou adapta-lo a um ambiente modificado. Antes da entrega do software são feitas atividades de planejamento.

## AS ÁREAS DO CONHECIMENTO DO SWEBOK

- São 10 as áreas de conhecimento do SWEBOK (knowledge areas – Kas):

Cap. 2	Requisitos de Software	Gerenciamento de Configuração de Software	Cap. 7
Cap. 3	Projeto de Software	Gerenciamento de Engenharia de Software	Cap. 8
Cap. 4	Construção de Software	Processo de Engenharia de Software	Cap. 9
Cap. 5	Teste de Software	Ferramentas e Métodos de Engenharia de Software	Cap. 10
Cap. 6	Manutenção de Software	Qualidade de Software	Cap. 11



## ➤ Introdução: O que é software:

Segundo o dicionário aurélio:

- aquilo que pode ser executado por um equipamento, no caso o hardware;
- um produto comercializado que consiste em um sistema de rotinas e funções.

Uma definição mais didática:

- instruções (programas de computador) que, quando executadas produzem a função e o desempenho desejados;
- estruturas de dados que possibilitam que os programas manipulem adequadamente a informação;
- documentos que descrevem a operação e o uso dos programas.

Para quê desenvolver software?

- automatizar um determinado processo;
- melhorar a qualidade das informações;
- apoiar decisões estratégicas;
- ser um instrumento de melhoria da qualidade de vida da sociedade;
- dar mais credibilidade as informações.

## ↪ Introdução: Aplicações de Software:

**Software básico:** coleção de programas escritos para apoiar outros programas. ex.: compiladores, gerenciadores de arquivos, drivers, etc...

**Software de tempo real:** software que monitora, analisa e controla eventos do mundo real. são sistemas aonde um resultado atrasado não tem valor. ex.: monitoração de batidas cardíacas, sistema de controle de tráfego aéreo, freios abs (carros), etc ...

**software comercial:** sistemas de operações comerciais e tomadas de decisões administrativas. ex.: folha de pagamentos, contas a pagar e a receber, controle de estoques, etc...

**Software científico e de engenharia:** caracterizado por algoritmos de processamento numérico. ex.: astronomia, análise de fadiga da mecânica de automóveis, manufatura automatizada, etc...

**Software embutido:** usado para controlar produtos e sistemas para os mercados industriais e de consumo. reside em memória só de leitura. ex.: controle do teclado de um microondas, controle de combustível, funções digitais de automóveis, etc...

**Software de computador pessoal:** envolve processamento de textos, planilhas eletrônicas, computação gráfica, multimedia, etc.

**Software baseado na web:** páginas web recuperados por um browser que incorporam instruções executáveis (cgi, html, java, perl, xml, etc...) e dados. ex.: hipertextos, formatos de áudio, vídeo, imagens, etc...

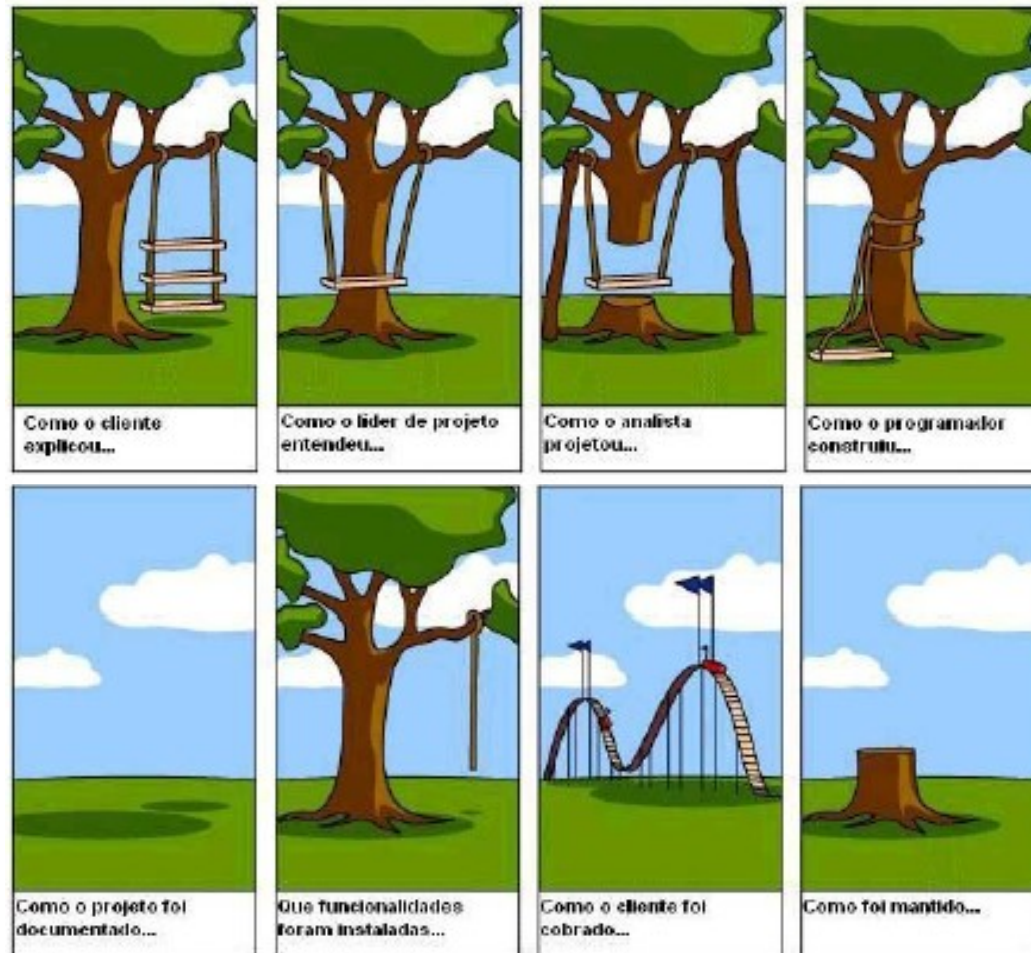
**Software de inteligência artificial:** faz uso de algoritmos não numéricos para resolver problemas de alta complexidade que não sejam favoráveis à computação ou à análise direta. ex.: sistemas especialistas, reconhecimento de padrões, redes neurais, jogos, etc...

## ↳ **Desenvolvimento de Software:**

Associada ao desenvolvimento do software devido a um conjunto de problemas que tornam esta atividade em uma atividade árdua e subjetiva. entre eles podemos citar:

- as estimativas de custo e prazo frequentemente são imprecisas;
- a produtividade das pessoas da área de software não tem acompanhado a demanda por seus serviços;
- a qualidade do software às vezes é menos adequada;
- não dedicamos tempo para coletar dados sobre o processo de desenvolvimento;
- poucos dados históricos para guiar estimativas;
- a comunicação entre analista e cliente frequentemente é muito pobre;
- falta ou deficiência em técnicas de teste;
- a tarefa de manutenção devora a maioria do orçamento destinado ao software.

## ➤ Expectativa do Usuário X Entendimento Analistas:



## ↳ Expectativa do Usuário X Entendimento Analistas:

**mais quais são as causas destes problemas?**

- o caráter do software;
- falhas humanas;
- gerentes sem nenhum *background* em software;
- os profissionais da área de software têm recebido pouco treinamento formal em novas técnicas para o desenvolvimento de software;
- resistência à mudança.

## ↳ Engenharia de Software:

- engenharia é a aplicação sistemática de conhecimentos científicos na criação e construção de soluções com um bom custo benefício para resolução de problemas práticos da sociedade.
- engenharia de software é uma forma de engenharia que aplica os princípios da ciência da computação e matemática para alcançar soluções com um bom custo benefício para resolução de problemas do software.
- a engenharia de software compreende de um conjunto de etapas que envolve ferramentas, métodos, processos e tem como alicerce a tudo isto o controle da qualidade.

## ↳ Engenharia de Software: Ferramentas, Métodos, Processos



- métodos: proporcionam os detalhes de "**como fazer**" para construir o software;
- ferramentas: fornecem suporte automatizado ou semi-automatizados aos métodos;
- processos: é a fundação da engenharia de software, provendo a sustentação e relacionamentos entre as camadas;
- foco na qualidade: garante toda qualidade do software gerado.

## ↳ Engenharia de Software: Desmitificando de Software

**Mitos Administrativos.** Advém de gerentes sobre pressão de **orçamento** e **tempo**.

**Mitos do Cliente.** Advém de falsas expectativas e insatisfação com o desenvolvedor

**Mitos do Profissional de Desenvolvimento.** Advém de se considerar o software como uma forma de arte. Será que o software é uma arte ou uma engenharia?



## ↳ Engenharia de Software: Desmitificando de Software

### Mito do Manual de Práticas

**Mito:** Já temos um manual repleto de padrões e procedimentos para a construção de software. Isso não oferecerá ao meu pessoal tudo o que eles precisam saber?

**Realidade:** O manual de padrões pode muito bem existir, mas será que ele é usado? Os profissionais de software têm conhecimento de sua existência? Ele reflete a moderna prática de desenvolvimento de software? É completo? Em muitos casos, a resposta a todas estas perguntas é “não”.

### Mito do Computador Moderno

**Mito:** Meu pessoal tem ferramentas de desenvolvimento de software de última geração; afinal de contas lhes compramos os mais novos computadores.

**Realidade:** É preciso muito mais do que o último modelo de computador para se fazer um desenvolvimento de software de alta qualidade. As ferramentas de engenharia de software auxiliadas por computador (CASE) são mais importantes do que o hw para se conseguir boa qualidade e produtividade; contudo, a maioria dos desenvolvedores de software não as usa plenamente.

## ↳ Engenharia de Software: Desmitificando de Software

### Mito das Hordas de Mongóis

**Mito:** Se nós estamos atrasados nos prazos, podemos adicionar mais programadores e tirar o atraso.

**Realidade:** O desenvolvimento de software não é um processo mecânico igual à manufatura. Acrescentar pessoas em um projeto de software atrasado torna-o ainda mais atrasado. Gasta-se tempo formando os recém-chegados, o que reduz o tempo de desenvolvimento produtivo. Pessoas podem ser acrescentadas, mas somente de uma forma planejada e bem coordenada.

### Mitos do Cliente: Especificação

**Mito:** Uma declaração geral dos objetivos é suficiente para se começar a escrever programas – podemos preencher os detalhes mais tarde.

**Realidade:** Uma definição inicial ruim é a principal causa de fracasso dos esforços de desenvolvimento de software. Uma descrição formal e detalhada do domínio da informação, função, desempenho, interfaces, restrições de projeto e critérios de validação é fundamental. Essas características podem ser determinadas somente depois de cuidadosa comunicação entre o cliente e o desenvolvedor.

## ↪ Engenharia de Software: Desmitificando de Software

### O Pior Mito do Cliente

**Mito:** Os requisitos de projeto modificam-se continuamente, mas as mudanças podem ser facilmente acomodadas, porque o software é flexível (Cuidado como o “Já que...”).

**Realidade:** É verdade que os requisitos de software se modificam, mas o impacto da mudança varia de acordo com o tempo em que ela é introduzida.

## ↳ Engenharia de Software: Custo de Reparos em Software



- **Custo aumenta com o tempo de descoberta do erro**
  - Custo de reparo
  - Custo de perda de oportunidades
  - Custo de perda de clientes
- O custo de **1 problema é 200 vezes maior se reparado após a implantação.**
- Os **erros mais caros** são aqueles cometidos na Análise de requisitos e descobertos pelo usuário!

## ↳ Engenharia de Software: Desmitificando de Software

### Mitos do Profissional: Terminar Mais Cedo

**Mito:** Assim que escrevemos o programa e o colocamos em funcionamento, nosso trabalho estará completo.

**Realidade:** Alguém disse certa vez que “quanto mais cedo se começa a ‘escrever o código’, mais tempo demora para que se consiga terminá-lo”. Os dados da indústria indicam que entre 50 e 70% de todo o esforço gasto num programa serão despendidos depois que ele for entregue pela primeira vez ao cliente.

## ↳ Engenharia de Software: Desmitificando de Software

### Mito da Qualidade

**Mito:** Assim que escrevemos o programa e o colocarmos em funcionamento, nosso trabalho estará completo.

**Realidade:** Um dos mecanismos mais efetivos de garantia de qualidade de software pode ser aplicado desde o começo de um projeto – a revisão técnica formal. As revisões de software são um “filtro da qualidade” que têm sido consideradas mais eficientes do que a realização de testes para a descoberta de defeitos.

## ↳ Engenharia de Software: Desmitificando de Software

### Mito do Executável

**Mito:** A única coisa a ser entregue em um projeto bem-sucedido é o programa funcionando.

**Realidade:** Um programa funcionando é somente uma parte de uma configuração de software que inclui vários outros elementos. A documentação forma os alicerces para um desenvolvimento bem-sucedido e fornece um guia para a tarefa de manutenção do software.

## ↪ Material de apoio:

### Bibliografia Básica

PRESSMAN, R. Engenharia de software. Rio de Janeiro: MacGraw-Hill, 2006.  
SOMMERVILLE, I. Engenharia de software. 8. ed. São Paulo: Addison Wesley, 2007.

### Bibliografia Complementar

PFLEEGER, S.L., et al, “Software Engineering”, Prentice Hall, 2005, 3rd edition.

IEEE Computer Society Real-World Software problems: A Self-Study Guide for Today’s Software Professional, Wiley-IEEE Computer Society Press, 2006.  
Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004. Disponível em <http://swebok.org>.



## ↪ Material de apoio:

### Referências Bibliográficas

FALBO, Ricardo de Almeida. Engenharia de Software – Notas de Aula. Universidade Federal do Espírito Santo (UFES). 2005.

SWEEBOK. *Software Engineering Body of Knowledge*. 2004.  
<http://www.computer.org/portal/web/swebok>