

# Centro Universitário FMU

Disciplina:

**Engenharia de Software I**

**Aula 1**

**Bacharelado em Ciência da Computação**

**Tecnologia em Análise e Desenvolvimento de Sistemas**

***Prof.: Celso Eduardo Guimarães***

***celso.guimaraes@fmu.br***

**Aula 1**  
**Análise e Desenvolvimento de Sistemas**  
**Ciência da Computação**  
**Engenharia de Software I**

- I. Apresentação do professor: breve histórico curricular.
- II. Horário de aula:
  - 08h50 – 10h05 - 1ª parte da aula (chamada antes de sair para o intervalo)
  - 10h05 – 10h25 – intervalo
  - 10h25 – 11h40 – 2ª parte da aula (chamada no término da aula)Cada chamada corresponde a 1,5 períodos.
- III. Apresentação dos objetivos de aprendizagem da disciplina.
- IV. Explicação de como será o Processo Avaliativo dessa disciplina.
- V. Conceitos básicos sobre lógica de programação

## O Processo avaliativo na Disciplina de Engenharia de Software I

☐ A média final na disciplina será composta por duas partes: N1 e N2

- N1 terá peso de 40% na nota.
- N2 terá peso de 60% na nota.
- Portanto: a média final (MF) será calculada assim:
  - $MF = (N1 * 0,4) + (N2 * 0,6)$
- Para compor a N1, teremos 3 avaliações:
  - A1 = Prova individual valendo de 0 a 10.
  - A2 = Prova individual valendo de 0 a 10.
  - A3 = Atividades desenvolvidas em sala de aula ou em casa. Poderão ser em grupo ou individual, conforme orientação do professor. Todas as atividades valerão de 0 a 10 e a nota final A3 será a média aritmética de todas as atividades.
  - $N1 = A1 + A2 + A3$
- Para compor a nota N2, teremos:
  - A4 = Atividade Prática Supervisionada (APS). Será explicada em breve. Terá peso de 10% da N2.
  - A5 = Prova valendo de 0 a 10 a ser realizada dentro do calendário da instituição (entre 01/06 e 06/06)
  - A6 = Avaliação substitutiva para quem perdeu A5 ou não conseguiu atingir a média requerida (6,0).
  - $N2 = (A4 * 0,1) + (A5 * 0,9)$  ou  $(A4 * 0,1) + (A6 * 0,9)$
- Não haverá qualquer tipo de arredondamento.
- Média final mínima para ser aprovado = 6,0.

## Objetivos de Aprendizagem

- 1. Introduzir, Conceituar engenharia de software e identificar a sua importância.**
- 2. Apresentar os diferentes PAPÉIS na Engenharia de Software e profissionais envolvidos.**
- 3. Relacionar o desenvolvimento de *softwares distintos a técnicas diferentes de engenharia de software*.**
- 4. Abordar as Ferramentas CASE (Computer-Aided Software Engineering - Engenharia de Software Auxiliada por Computador) e seus tipos.**
- 5. Analisar algumas questões éticas e profissionais para engenheiros de software.**

## Literatura Indispensável:

- ✓ PRESSMAN, R., MAXIM, B. Engenharia de Software.
- ✓ SOMMERVILLE, I. Engenharia de Software.
- ✓ Pfleeger, S. L. Engenharia de Software - Teoria e Prática



Tente responder....  
O que é Engenharia de  
Software???



dicionário

**Engenharia:** aplicação de métodos científicos ou empíricos à utilização dos recursos da natureza em benefício do ser humano.

*Engenharia é a aplicação do conhecimento científico, econômico, social e prático, com o intuito de inventar, desenhar, construir, manter e melhorar estruturas, máquinas, aparelhos, sistemas, materiais e processos.*

Um **engenheiro** é um profissional de [engenharia](#), preocupado com a aplicação do conhecimento científico, matemático e da criatividade para desenvolver soluções para problemas técnicos.



- **Conjunto de componentes lógicos de um computador ou sistema de processamento de dados;**
- **Programa, rotina ou conjunto de instruções que controlam o funcionamento de um computador; suporte lógico.**



# Engenharia de Software

## O QUE É???



É um ramo da engenharia com foco em desenvolver sistemas de software **de Alta Qualidade e Custos Adequados**

Não há grandes limitações físicas como em outras engenharias (elétrica, mecânica, construção civil, etc)

Engenharia “teoricamente” mais simples, sem as barreiras naturais → muitas vezes, os sistemas tornam-se extremamente **complexos**

# UM POUCO DA HISTÓRIA DO COMPUTADOR



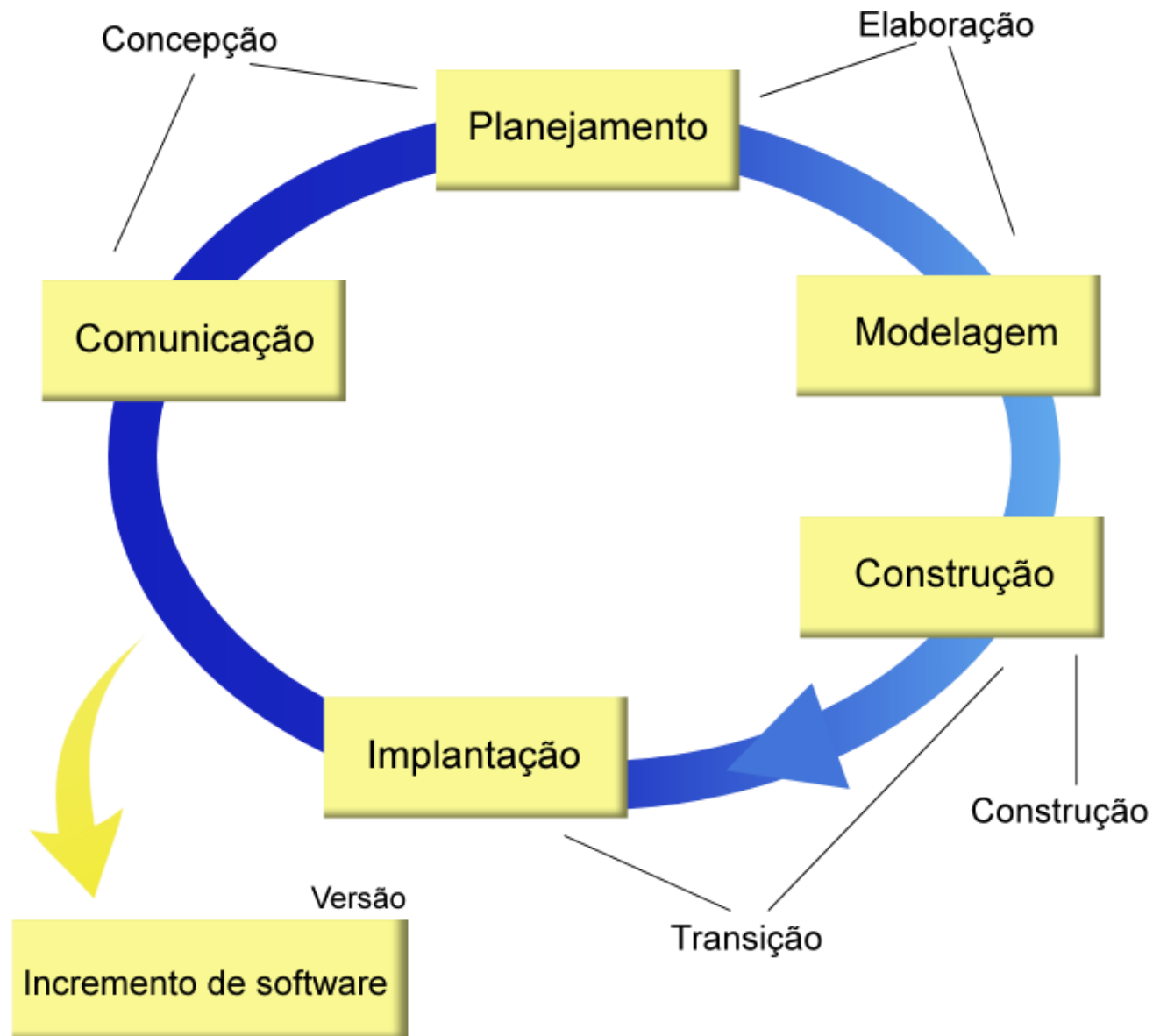
<https://www.youtube.com/watch?v=X3CXM62j9Uc>

# Ciclo de Vida de Projeto de Software

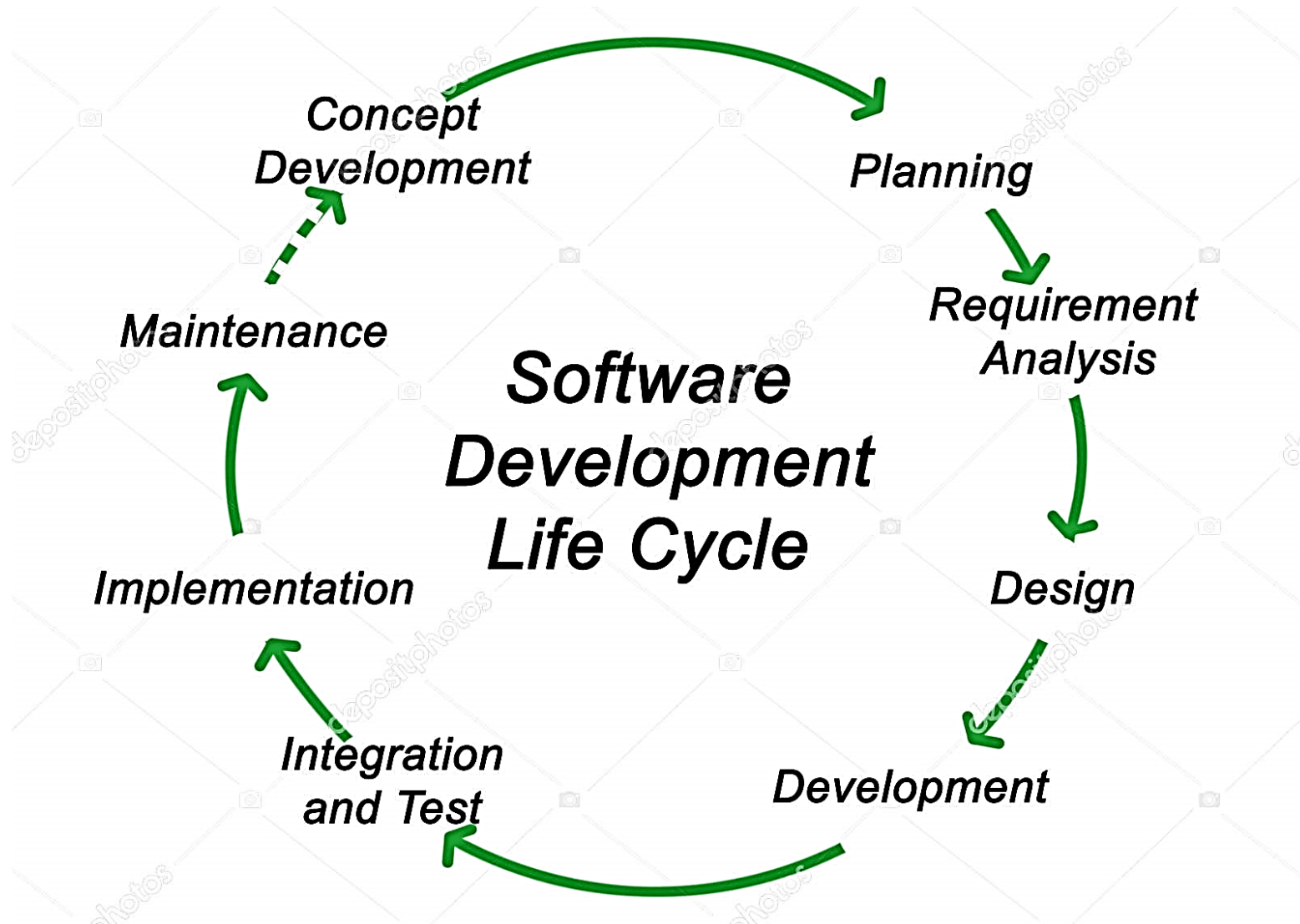


A engenharia de software está relacionada com todas as fases e aspectos de desenvolvimento e manutenção do software, incluindo o gerenciamento de projeto, ferramentas de apoio, metodologia para produção de software.

# Ciclo de Vida de Projeto de Software



# Ciclo de Vida de Software



## Camadas da Engenharia de Software



# Engenharia de Software

Conceito proposto em 1968 devido a uma **crise** nos projetos de sistemas.

→ Hardware evoluindo possibilitando operações até então irrealizáveis

→ Isso resultou em softwares complexos, ou seja, projetos complexos (altos custos, atrasos...)

→ Desenvolvimento informal não era mais aceitável (viável)

# Engenharia de Software



*ibm.com*

Custos de hardware  
estavam caindo



Mas os custos de  
software  
aumentavam!!!



Surgiu a necessidade de adoção de novas técnicas e métodos para controlar melhor os grandes projetos de software.



# Engenharia de Software

O **objetivo** dessa disciplina nesse curso é:

Desenvolver o estudo do **papel da Engenharia de Software** para determinação de um **processo adequado de desenvolvimento**.

para isso...

Vamos apresentar a Engenharia de Software, conceituando os modelos de processos existentes, de modo a permitir a identificação e determinação das atividades essenciais em um processo de desenvolvimento.



## Quem realiza?

Os engenheiros de software **criam** e **dão suporte** a ele e, praticamente, todos do mundo industrializado o utilizam, direta ou indiretamente.

## Por que ele é importante?

Software é importante porque **afeta** a quase todos os aspectos de **nossas vidas** e tornou-se pervasivo (incorporado) no comércio, na cultura e em nossas atividades cotidianas. A engenharia de software é importante porque ela nos **capacita** para o **desenvolvimento de sistemas complexos** dentro do **prazo** e com alta **qualidade**.



**Por que precisamos da Engenharia de Software?**

## Por que precisamos da Engenharia de Software?



Como o cliente explicou



Como o líder de projeto entendeu



Como o analista planejou



Como o programador codificou



O que os beta testers receberam



Como o consultor de negócios descreveu



Valor que o cliente pagou



Como o projeto foi documentado



O que a assistência técnica instalou



Como foi suportado



Quando foi entregue



O que o cliente realmente necessitava

# Engenharia de Software

*retomando...*

A Engenharia de Software está relacionada a todos os **aspectos do software**.

O software envolve os **programas** de computador e a **documentação**

A diferença para Ciência da Computação é que a engenharia está relacionada com a parte **prática** de desenvolvimento de software



# Definindo Software

- (1) instruções (programas de computador) que, quando executadas, fornecem características, funções e desempenho desejados;
- (2) estruturas de dados que possibilitam aos programas manipular informações adequadamente; e
- (3) informação descritiva, tanto na forma impressa como na virtual, descrevendo a operação e o uso dos programas.

*Pressman*

***Software é desenvolvido ou passa por um processo de engenharia; ele não é fabricado no sentido clássico.***

Software não é apenas o pgm do computador, mas também toda a documentação associada e os dados de configuração necessários para fazer com que esses programas operem corretamente.

*Sommerville*

# Ferramentas CASE

**Ferramentas CASE** (do inglês Computer-Aided Software Engineering  
“Engenharia de Software Auxiliada por Computador”)

→ é uma classificação que abrange todas **ferramentas** baseadas em computadores que auxiliam atividades de engenharia de software, desde análise de requisitos e modelagem até programação e testes.

# FERRAMENTA



**Qualquer instrumento que se usa para a realização de um trabalho.**

**Ferramenta é algo que realiza uma função, implementando e dando suporte a metodologias.**



## FERRAMENTAS CASE

- Ferramentas automatizadas que tem como objetivo auxiliar o **desenvolvedor** de sistemas em uma ou várias etapas do ciclo de desenvolvimento de software.
- Para simplificar, podemos dizer que as ferramentas CASE auxiliam os engenheiros de software e os profissionais de processamento de dados a especificar e projetar o software.(FISHER, 1990)

# Ferramentas CASE

- ❑ **FERRAMENTAS CASE** podem ser utilizadas para **gerar um esboço do programa**, a partir de um **projeto**.
- ❑ → Em alguns casos o programador precisa somente acrescentar detalhes da operação de cada componente do programa.

Estão para o Engenheiro de Software assim como o CAD está para a Engenharia Civil

# Ferramentas CASE

- Uma ferramenta CASE é um **aplicativo** que auxilia os profissionais envolvidos na tarefa de produzir sistemas.
- O tipo de “ajuda” que a ferramenta fornece, depende exclusivamente da proposta do fabricante.
- Um dos componentes indispensáveis de uma ferramenta CASE é a **modelagem visual**, ou seja, a possibilidade de representar, através de **modelos gráficos**, o que está sendo definido.
  - ✓ Algumas são baseadas na Notação UML.

Demonstração de um software da IBM da sua suíte Rational para Gerenciamento de Requisitos de Software:

<https://www.youtube.com/watch?v=BaQOu-7hC88>

## CATEGORIAS

- Não há um padrão definido para a categorização das CASE
- No entanto são definidas pelas terminologias a seguir

## CATEGORIAS DAS CASE

- **Front End ou Upper CASE:** apoia as etapas iniciais de criação dos sistemas: as fases de planejamento, análise e projeto do programa ou aplicação.
- **Back End ou Lower CASE:** dão apoio à parte física, isto é, a codificação testes e manutenção da aplicação.
- **I-CASE ou Integrated CASE:** classifica os produtos que cobrem todo o ciclo de vida do software, desde os requisitos do sistema até o controle final da qualidade.



**Tabela 1.1**

Perguntas frequentes sobre software

Pergunta	Resposta
O que é software?	Softwares são programas de computador e documentação associada. Produtos de software podem ser desenvolvidos para um cliente específico ou para o mercado em geral.
Quais são os atributos de um bom software?	Um bom software deve prover a funcionalidade e o desempenho requeridos pelo usuário; além disso, deve ser confiável e fácil de manter e usar.
O que é engenharia de software?	É uma disciplina de engenharia que se preocupa com todos os aspectos de produção de software.
Quais são as principais atividades da engenharia de software?	Especificação de software, desenvolvimento de software, validação de software e evolução de software.
Qual a diferença entre engenharia de software e ciência da computação?	Ciência da computação foca a teoria e os fundamentos; engenharia de software preocupa-se com o lado prático do desenvolvimento e entrega de softwares úteis.
Qual a diferença entre engenharia de software e engenharia de sistemas?	Engenharia de sistemas se preocupa com todos os aspectos do desenvolvimento de sistemas computacionais, incluindo engenharia de hardware, software e processo. Engenharia de software é uma parte específica desse processo mais genérico.
Quais são os principais desafios da engenharia de software?	Lidar com o aumento de diversidade, demandas pela diminuição do tempo para entrega e desenvolvimento de software confiável.
Quais são os custos da engenharia de software?	Aproximadamente 60% dos custos de software são de desenvolvimento; 40% são custos de testes. Para software customizado, os custos de evolução frequentemente superam os custos de desenvolvimento.
Quais são as melhores técnicas e métodos da engenharia de software?	Enquanto todos os projetos de software devem ser gerenciados e desenvolvidos profissionalmente, técnicas diferentes são adequadas para tipos de sistemas diferentes. Por exemplo, jogos devem ser sempre desenvolvidos usando uma série de protótipos, enquanto sistemas de controle críticos de segurança requerem uma especificação analisável e completa. Portanto, não se pode dizer que um método é melhor que outro.
Quais diferenças foram feitas pela Internet na engenharia de software?	A Internet tornou serviços de software disponíveis e possibilitou o desenvolvimento de sistemas altamente distribuídos baseados em serviços. O desenvolvimento de sistemas baseados em Web gerou importantes avanços nas linguagens de programação e reuso de software.

Fonte:  
Sommerville

# Ética na engenharia de software

1. *Confidencialidade.* Você deve respeitar naturalmente a confidencialidade de seus empregadores ou clientes, independentemente de ter sido ou não assinado um acordo formal de confidencialidade.
2. *Competência.* Você não deve deturpar seu nível de competência. Você não deve aceitar conscientemente um trabalho que esteja fora de sua competência.
3. *Direitos de propriedade intelectual.* Você deve ter conhecimento das leis locais a respeito da propriedade intelectual, como patentes e copyright. Você deve ter cuidado para garantir que a propriedade intelectual dos empregadores e clientes seja protegida.
4. *Mau uso do computador.* Você não deve usar suas habilidades técnicas para fazer mau uso de computadores de outras pessoas. Esse mau uso varia de relativamente trivial (jogar videogames em uma máquina do empregador, por exemplo) até extremamente sério (disseminar vírus ou outros *malwares*).

**Fonte:**  
**Sommerville**



## **ALGUNS MITOS QUE ENVOLVEM A ENGENHARIA DE SOFTWARE**

**1. “Se o projeto estiver atrasado, é só contratar mais programadores”**

Mesmo que exista o conceito de *Fábrica de Softwares*, o processo de desenvolvimento não é como uma linha de produção.

Ao inserir um desenvolvedor em um projeto, ele levará um tempo para se familiarizar com o código e com o que está sendo feito, para então produzir.

Portanto, alocar programadores para tentar resolver um programa de cronograma pode surgir efeitos contrários.

**2. “Faça um bom manual, com regras e padrões, assim a equipe terá tudo que precisa saber”**

Na engenharia de software, o desenvolvimento de cada projeto é único, uma vez que desenvolver não é somente seguir uma receita de bolo.

Os clientes são muito diferentes, assim como os programadores. Além disso, as prioridades dependem do projeto.

O planejamento é essencial para levar os requisitos e trabalhar em um novo projeto.



### 3. “Terceirizar o trabalho é garantia de tranquilidade”

Quando o trabalho é muito complexo, ou requer um know-how que sua equipe não tem, ou quando o prazo está apertado, muitos pensam que terceirizar vai ser a garantia de tranquilidade e nenhum trabalho.

Porém, ao terceirizar um trabalho, você continuará com a responsabilidade sobre ele. Além disso, a maioria das empresas, terceiriza o serviço, mas ao comprar o código, fica responsável pelas manutenções. E esse pode ser um grande problema.

Será que sua equipe saberá lidar com esse código? Por isso, é importante pensar bem antes de terceirizar um serviço que terá continuidade no futuro.

### 4. “Um software pode ser construído somente olhando o propósito geral, os detalhes podem ser levados em conta posteriormente”

Muitos clientes procuram as empresas de engenharia de software para fazer pequenos “ajustes”, porém isso pode gerar um retrabalho para os desenvolvedores.

Por isso, ao pensar em um software, é preciso mapear ao máximo todas as funcionalidades a serem desempenhadas.

## 5. “Se o programa funciona, nosso trabalho está completo”

Todo processo de desenvolvimento precisa buscar a qualidade, e somente funcionar não garante que o trabalho teve fim.

É preciso ter bem estabelecido que o código foi bem comentado, assim como a tecnologia precisa ser adequada, e o banco de dados precisa estar otimizado.

Além disso, a infraestrutura do cliente deve comportar o que está sendo desenvolvido.

Pensar em questões como a quantidade de acessos para fazer um sistema que suporte grande tráfego também é essencial.

As empresas de engenharia de software não vendem somente o que é executável. Mas sim, todo o processo.

## 6. “Entregarei ao cliente somente o código executável”

O produto *palpável* que o cliente recebe é somente o executável em alguns casos, mas também pode ser enviado o código fonte e toda a documentação.

Mas é preciso pensar que o que é vendido é todo o processo de desenvolvimento e não somente um código executável.

# Processo de Software

Na elaboração de um produto ou sistema, é importante seguir uma série de passos previsíveis — um roteiro que ajude a criar um resultado de alta qualidade e dentro do prazo estabelecido. O roteiro é denominado...

processo de software

(Pressman)

# Processo de Software

- Cria uma base para o controle gerencial de projetos;
- Estabelece o contexto no qual os métodos e as técnicas são aplicados;
- São produzidos os artefatos (produtos) de trabalho;
- São estabelecidos os marcos;
- A qualidade é assegurada;
- As mudanças são gerenciadas.

mas...

não é muita utopia querer controlar tudo?

isso não atrapalha a liberdade criativa do engenheiro de software?

## **Para que serve a UTOPIA?**

**Utopia** é a ideia de civilização ideal, fantástica, imaginária. É um sistema ou plano que parece irrealizável, é uma fantasia, um devaneio, uma ilusão, um sonho. Do grego “ou+topos” que significa “lugar que não existe”

*www.significados.com.br*

<https://www.youtube.com/watch?v=9iqi1oaKvzs>

## **Para que serve a UTOPIA?**

**Utopia** é a ideia de civilização ideal, fantástica, imaginária. É um sistema ou plano que parece irrealizável, é uma fantasia, um devaneio, uma ilusão, um sonho. Do grego “ou+topos” que significa “lugar que não existe”

*[www.significados.com.br](http://www.significados.com.br)*

<https://www.youtube.com/watch?v=9iqi1oaKvzs>

# Processo de Software

O objetivo principal é colocar ordem no caos existente na área de desenvolvimento de software.

*“**Todo** o diálogo e a interação entre usuários, projetistas, ferramentas de desenvolvimento e tecnologias devem ser **transformados num processo.**”*

Isso pode parecer utópico – mas é um caminho a ser perseguido – cabe ao profissional de software buscar o equilíbrio entre a ordem e o caos – entre o método rígido e a liberdade criativa.



# Processo de Software

Processo de Software = Framework (Arcabouço)

## **Arcabouço** (*Dicionário*)

esqueleto; estrutura óssea que sustenta um corpo (humano ou animal)

*Os modelos de processo de desenvolvimento de software são representações abstratas de um processo de software.*



Que possui um certo grau de generalização

# Processo de Software

A qualidade não é garantida com a simples utilização de um determinado processo de software.

O processo a ser seguido deve ser escolhido e customizado com base em critérios essenciais com a finalidade de se obter um desenvolvimento com sucesso.

Devem ser levados em consideração os aspectos que envolvem o projeto a ser desenvolvido – circunstâncias no ambiente interno e externo à organização que produzirá o software.

Quanto maior maturidade tiver a organização, maior será a eficácia e a eficiência na aplicação de determinado modelo de processo de software.

# Engenharia de Software

Qual é o método mais **apropriado** para desenvolvimento de software?

A Engenharia de Software procura selecionar o(s) método(s) mais apropriado de acordo com as circunstâncias que envolvem o projeto → **a mais eficiente e eficaz.**

# Engenharia de Software

## Áreas de Conhecimento

- ☐ Requisitos de software
- ☐ Projeto de software
- ☐ Construção de software
- ☐ Teste de software
- ☐ Manutenção de software
- ☐ Gerência de configuração de software
- ☐ Gerência de engenharia de software
- ☐ Processos de Engenharia de Software
- ☐ Ferramentas e Métodos de Engenharia de Software
- ☐ Qualidade de software

## **Leitura complementar para casa – artigo:**

**Invista em você! Saiba como a DevMedia pode ajudar sua carreira no dia-dia: 10 dicas fundamentais para equipes de software**

<https://www.devmedia.com.br/engenharia-de-software-no-dia-dia-10-dicas-fundamentais-para-equipes-de-software/26950>

## ATIVIDADE 1 – em classe – 10/02/2020

### Em grupos de até 4 alunos

Considerem o seguinte cenário:

**Vocês fazem parte de equipes de desenvolvimento de software e vão iniciar um novo projeto de desenvolvimento de software para uma escola de idiomas que acabaram de contratar vocês. Esse software deverá:**

- **Apoiar o gerenciamento dessa escola de idiomas.**
- **Gerenciar os cursos, as turmas, os alunos, os professores, os horários das aulas e o controle financeiro da instituição.**
- **Gerenciar também os pagamentos de mensalidades e a venda de livros referentes aos diferentes cursos.**
- **A atividade consiste em:**
  - **Definir uma estratégia para a equipe desenvolver esse projeto.**
  - **Definir os papéis e etapas que vocês acreditam serem necessárias.**
  - **Cada grupo deverá eleger um representante que apresentará suas estratégias, papeis definidos e etapas que foram planejados pelos grupos.**
  - **Tudo deverá ser anotado e entregue para o professor identificando os alunos.**