

# **Engenharia de Software**

**Aula: Introdução a  
Engenharia de Software e  
Conceitos Iniciais.**



Prof. Anderson Augusto Bosing

# O que é **Software**?

**Consiste em uma série de programas separados + arquivos de configuração + documentação do sistema + documentação do usuário + ,se for o caso, sites na WEB para os usuários fazerem download de informações recentes. Adaptado Sommerville (2014).**

**Pressman (2011) afirma que todo projeto de software nasce de uma necessidade de negócio, seja para adaptar algo já existente ou para criar um novo produto ou serviço.**



# O que é **Engenharia de Software**?

É uma disciplina da engenharia que se ocupa de todos os aspectos da produção de software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema, depois que ele entrou em operação.  
**Sommerville (2014).**

**Engenharia de Software** é o estabelecimento e uso de sólidos princípios de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.  
**Pressman(2011)**



Prof. Anderson Augusto Bosing

# O que é **Engenharia de Software**?

- Em suma, desenvolver software é uma atividade que não se confunde com escrever programas para computador.
- Desenvolvimento de software envolve procedimentos que exigem a abordagem simultânea e integra de aspectos técnicos e gerenciais.
- Se ocupa de todos os aspectos da produção do software, desde os estágios iniciais de especificação do sistema até a manutenção desse sistema.
- Trabalha com técnicas, teorias, métodos e ferramentas que auxiliam na produção do software.
- É uma área do conhecimento voltada para:
  - ✓ Especificação de software
  - ✓ Desenvolvimento de software
  - ✓ Validação de software
  - ✓ Evolução de software.



Prof. Anderson Augusto Bosing

# O que é Engenharia de Software?

**Em resumo é uma disciplina de engenharia que se preocupa com todos os aspectos de produção de software.**



Prof. Anderson Augusto Bosing

# Mas por que Engenharia?

Pois deve ser um processo sistêmico e regrado.



Prof. Anderson Augusto Bosing

# 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.



Prof. Anderson Augusto Bosing

# **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.**



Prof. Anderson Augusto Bosing

# 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.

- ✓ Pagamentos por uso de acordo com o uso.
- ✓ Software as a Service(SaaS).
- ✓ Sistemas baseadas na web, em vez de sistemas locais.
- ✓ WebServices e API's possibilitam que funcionalidades sejam acessadas via internet.



# **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.**



Prof. Anderson Augusto Bosing

# **Quais são os atributos essenciais de um bom software?**

## **Manutenibilidade**

**O software deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes. Esse é um atributo crítico, porque a mudança de software é um requisito inevitável de um ambiente de negócio em mudança**



Prof. Anderson Augusto Bosing

# **Quais são os atributos essenciais de um bom software?**

## **Confiança e proteção**

A confiança do software inclui uma série de características como confiabilidade, proteção e segurança. Um software confiável não deve causar prejuízos físicos ou econômicos no caso de falha de sistema. Usuários maliciosos não devem ser capazes de acessar ou prejudicar o sistema.



Prof. Anderson Augusto Bosing

# **Quais são os atributos essenciais de um bom software?**

## **Eficiência**

O software não deve desperdiçar os recursos do sistema, como memória e ciclos do processador. Portanto, eficiência inclui capacidade de resposta, tempo de processamento, uso de memória etc.



Prof. Anderson Augusto Bosing

# **Quais são os atributos essenciais de um bom software?**

## **Aceitabilidade**

**O software deve ser aceitável para o tipo de usuário para o qual foi projetado. Isso significa que deve ser compreensível, usável e compatível com outros sistemas usados por ele.**



Prof. Anderson Augusto Bosing

# Engenheiro de Software

A função de um profissional como Engenheiro de Software:

- ✓ Atuar no desenvolvimento de aplicativos para dispositivos móveis, como smartphones, tablets, jogos e softwares.
- ✓ Atuar na área de Gestão que é o gerenciamento de negócios e projetos de empresas de computação e software.
- ✓ Atuar em projetos, desenvolvimento, implantação e evolução de softwares complexos, corretos, disponíveis, seguros e tolerantes a falhas e com usabilidade.
- ✓ Desenhar, especificar, programar e testar soluções que atendam às necessidades do mercado, da sociedade, das organizações e dos indivíduos, levando em conta os impactos sociais.



Prof. Anderson Augusto Bosing

# Engenheiro de Software

O profissional Engenheiro de Software produz um software que podemos dividir em:

**Produtos genéricos** - software de caixa ou de prateleira.

A especificação do que o software deve fazer é de propriedade do desenvolvedor de software e as decisões sobre as mudanças de software são feitos pelo desenvolvedor.

**Produto sob encomenda** - quando um cliente solicita a uma empresa de software que desenvolva um software específico às necessidades da empresa.

A especificação do que o software deve fazer é propriedade do cliente para o software e eles tomam decisões sobre as mudanças necessárias no software.



Prof. Anderson Augusto Bosing

# Tipos de Aplicações de Software

## Aplicações stand-alone.

Essas são as aplicações executadas em um computador local, como um PC. Elas contêm toda a funcionalidade necessária e não precisam estar conectadas a uma rede. Exemplos de tais aplicações são aplicativos de escritório em um PC, programas CAD, software de manipulação de fotos etc.

## Aplicações interativas baseadas em transações

São aplicações executadas em um computador remoto e são acessadas pelos usuários a partir dos seus próprios PCs ou terminais. Essas incluem aplicações web tais como para e-commerce.



# Tipos de Aplicações de Software

## Sistemas de controle embutidos.

São sistemas de software de controle que controlam e gerenciam dispositivos de hardware. Numericamente, provavelmente existem mais sistemas embutidos do que qualquer outro tipo de sistema.

## Sistemas de processamento de lotes

São sistemas corporativos projetados para processar dados em grandes lotes. Eles processam grande número de entradas individuais para criar as saídas correspondentes. Exemplos de sistemas de lotes incluem sistemas periódicos de cobrança, como sistemas de cobrança telefônica, e sistemas de pagamentos de salário.



Prof. Anderson Augusto Bosing

# Tipos de Aplicações de Software

## Sistemas de entretenimento.

São sistemas cuja utilização principal é pessoal e cujo objetivo é entreter o usuário. A maioria desses sistemas é de jogos de diferentes tipos. A qualidade de interação com o usuário é a característica particular mais importante dos sistemas de entretenimento.

## Sistemas para modelagem e simulação.

São sistemas que incluem vários objetos separados que interagem entre si, desenvolvidos por cientistas e engenheiros para modelar processos ou situações físicas. Esses sistemas geralmente fazem uso intensivo de recursos computacionais e requerem sistemas paralelos de alto desempenho para executar.



Prof. Anderson Augusto Bosing

# Tipos de Aplicações de Software

## Sistemas de coleta de dados.

**São sistemas que coletam dados de seu ambiente com um conjunto de sensores e enviam esses dados para outros sistemas para processamento. O software precisa interagir com sensores e frequentemente é instalado em um ambiente hostil, por exemplo, dentro de uma máquina ou em um lugar Remoto.**

## Sistemas de sistemas.

**São sistemas compostos de uma série de outros sistemas de software. Alguns deles podem ser produtos genéricos de software, como um programa de planilha eletrônica. Outros sistemas do conjunto podem ser escritos especialmente para esse ambiente.**



Prof. Anderson Augusto Bosing

# PONTOS IMPORTANTES

- A engenharia de software é uma disciplina de engenharia que se preocupa com todos os aspectos de produção de software.
- Atributos essenciais do produto de software são manutenibilidade, confiança, proteção, eficiência e aceitabilidade.
- Existem muitos tipos diferentes de sistemas e cada um requer ferramentas de engenharia de software e técnicas apropriadas para o seu desenvolvimento.



Prof. Anderson Augusto Bosing

# VAMOS PRATICAR ?

Aponte a câmera do seu  
smartphone.



Prof. Anderson Augusto Bosing

# Engenharia de Software

**Aula: Conceitos de  
Sistemas da Informação,  
Mitos da Engenharia de  
Software.**



Prof. Anderson Augusto Bosing

# Mitos da Engenharia de Software

Mito do Gerenciamento

**Mito 1.** "Se a equipe dispõe de um manual repleto de padrões e procedimentos de desenvolvimento de software, então a equipe está apta a encaminhar bem o desenvolvimento. A equipe já não tem tudo o que ela precisa saber?"

**Realidade:** o manual é útil, confuso, muitas vezes, fica defasado rapidamente, não é tão completo quanto deveria, não é adaptável a todos os projetos ou, ainda, não busca melhorar o tempo de entrega com foco na qualidade, sendo assim, o manual apenas não é o suficiente.



Prof. Anderson Augusto Bosing

# Mitos da Engenharia de Software

Mito do Gerenciamento

**Mito 2.** "Se o desenvolvimento do software estiver atrasado, basta aumentar a equipe para honrar o prazo de desenvolvimento".

**Realidade:** Existe um custo em tempo para que um desenvolvedor passe a render, e os desenvolvedores que já estão no projeto deverão parar para auxiliar os novos colaboradores. Certas coisas exigem tempo, porém, quando bem planejadas, a inclusão de novos desenvolvedores ajudará.



Prof. Anderson Augusto Bosing

# Mitos da Engenharia de Software

Mito do Cliente

**Mito 3.** “Uma descrição breve e geral dos requisitos do software é o suficiente para iniciar o seu projeto... maiores detalhes podem ser definidos posteriormente”.

**Realidade:** quanto maior a definição dos requisitos, mais correto será o desenvolvimento e menor será o retrabalho.



Prof. Anderson Augusto Bosing

# Mitos da Engenharia de Software

Mito do Desenvolvedor

**Mito 4.** “Após a edição do programa e a sua colocação em funcionamento, o trabalho está terminado.”

**Realidade:** O que ocorre na realidade é completamente diferente disto. Segundo dados obtidos que 50 a 70% do esforço de desenvolvimento de um software é despendido após a sua entrega ao cliente (manutenção).



Prof. Anderson Augusto Bosing

# Mitos da Engenharia de Software

Mito do Desenvolvedor

**Mito 5.** “A equipe de desenvolvimento entende que, enquanto o software não estiver em uso, não é possível validar sua qualidade.”

**Realidade:** Inúmeros tipos de testes podem e devem ser feitos durante o desenvolvimento, visando encontrar a menor quantidade possível de erros durante a fase de uso.



Prof. Anderson Augusto Bosing

# Mitos da Engenharia de Software

Mito do Desenvolvedor

**Mito 6.** “A engenharia de software obriga a equipe de desenvolvimento a entregar uma documentação volumosa e trabalhosa de ser criada.”

**Realidade:** A engenharia de software visa à entrega de um produto de qualidade, e não à entrega de uma documentação volumosa.



Prof. Anderson Augusto Bosing

# O que é Sistema da Informação?

**Sistema de Informação, sigla S.I., é um conjunto de componentes inter-relacionados (pessoas, hardware, software, redes de comunicações e recursos de dados) que coletam (ou recuperam), processam, armazenam e distribuem informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização. (LAUDON; LAUDON, 2004, p. 7).**



# Conceitos de Informação

**DADOS** são fatos que ainda não foram trabalhados: Nome de Empregado e Registro de Identidade.

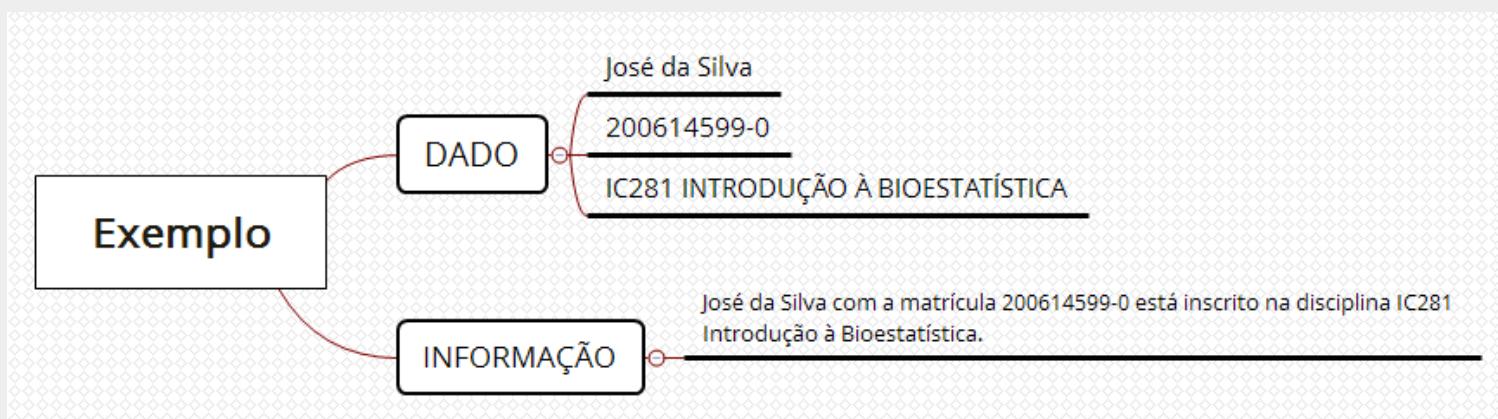


**INFORMAÇÃO** são dados organizados ou ordenados de forma significativa de modo que adquirem um valor adicional. Ex: Total de horas trabalhadas em uma semana, Estoque médio mensal, etc.

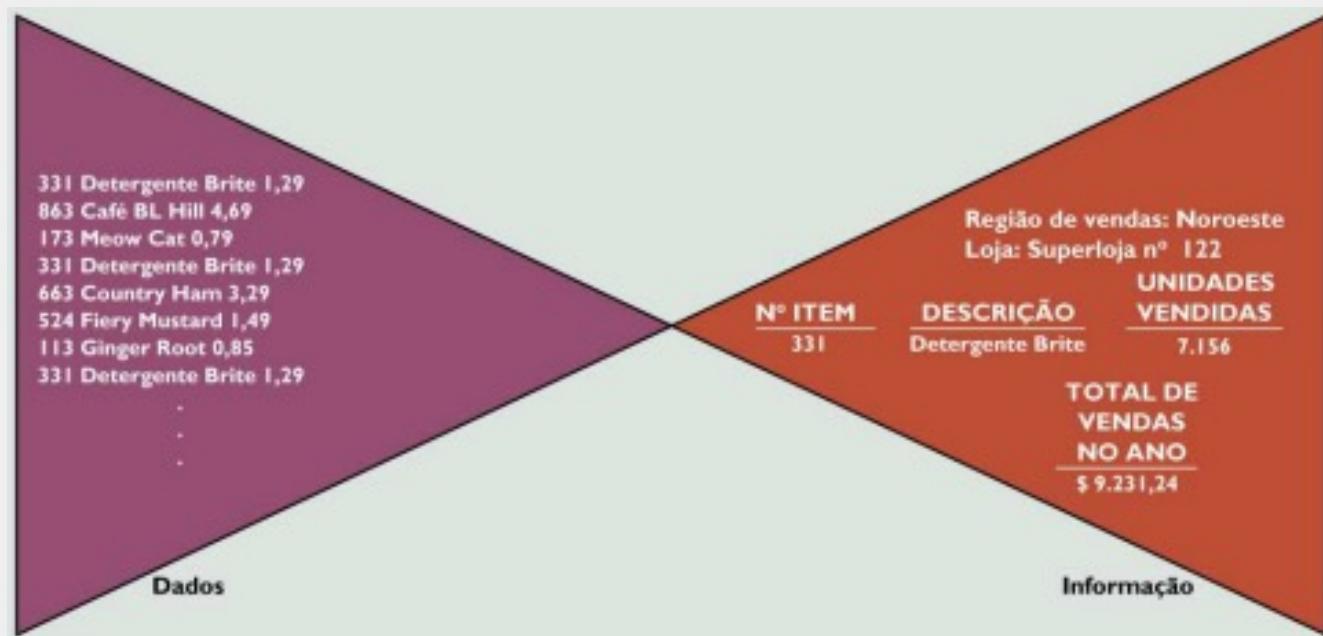
# Conceitos de Informação

A transformação de dados em informação é tarefa principal dos sistemas de informação.

A transformação de dados em informação é um processo ou uma série de tarefas logicamente relacionadas, executadas para atingir um resultado definido



# Conceitos de Informação



# Conceitos de Informação

O processo de definição das relações entre dados requer conhecimento.

- Conhecimento são as regras, diretrizes, procedimentos usados para selecionar, organizar e manipular os dados, com a finalidade de torná-los úteis para uma tarefa específica.

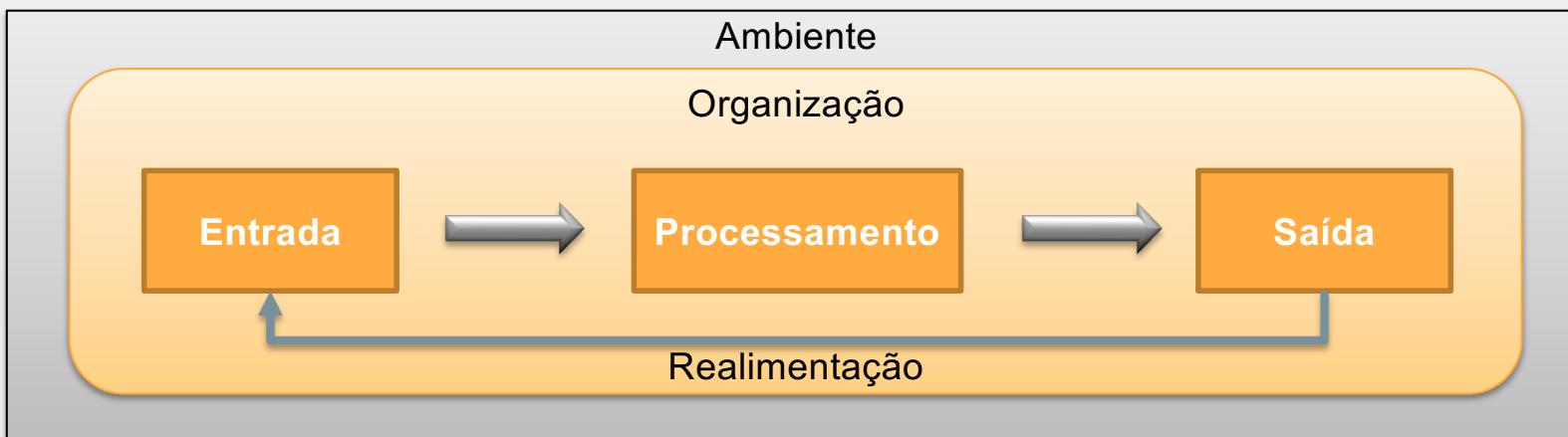


Através da informação as pessoas adquirem conhecimento.



Prof. Anderson Augusto Bosing

# Conceitos de Informação

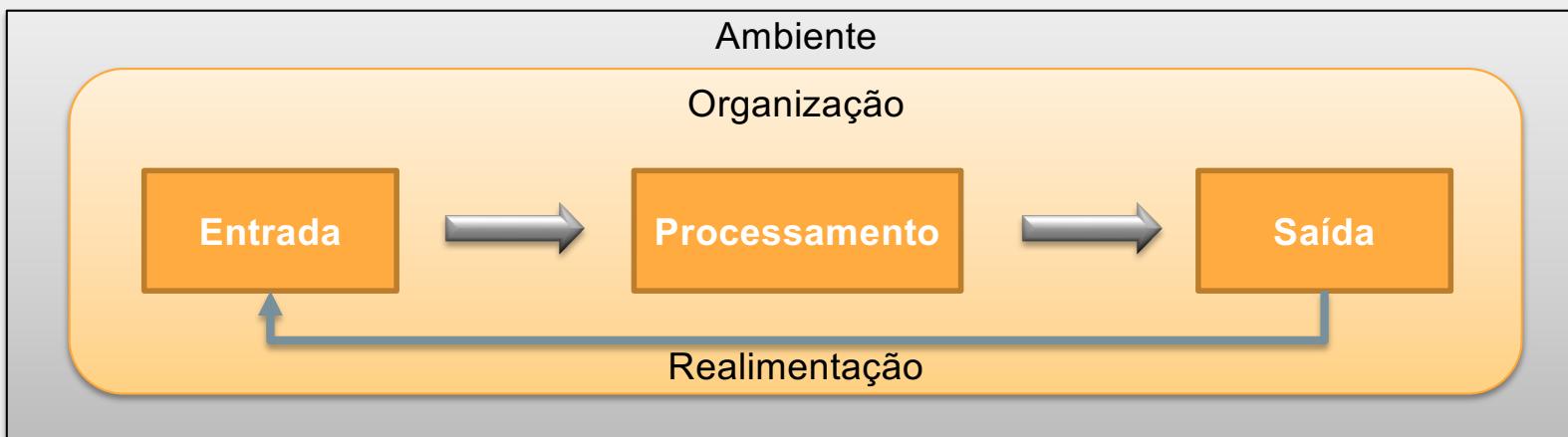


**Entrada:** envolve captação e reunião de elementos que entram no sistema para serem processados. Ex. matéria-prima, energia, dados e esforço humano.

**Processamento:** envolve processos de transformação que converte insumo (entrada) em produto. Ex. controle de estoque, respiração humana ou cálculos matemáticos.

**Saída:** envolve a transferência de elementos produzidos por um processo de transformação até o seu destino final. Ex. produtos acabados, serviços, informações gerenciais.

# Conceitos de Informação



**Ambiente:** é o conjunto de elementos que não pertencem ao sistema, mas qualquer alteração no sistema pode mudar ou alterar os seus elementos e qualquer alteração nos seus elementos pode mudar ou alterar o sistema.

**Realimentação:** pode ser considerado como a reintrodução de uma saída sob forma de informação. Ou ainda, é uma saída usada para fazer ajustes ou modificações nas atividades de entrada ou processamento.

# Exemplo: Folha de Pagamento

## Entrada:

- As horas trabalhadas pelos funcionários devem ser levantadas (captadas) através dos cartões de horas do funcionário.

## Processamento:

- As horas trabalhadas de cada funcionário devem ser convertidas em pagamento líquido.
- Multiplicação das horas trabalhadas pela taxa de pagamento por hora do empregado, para se obter o pagamento bruto.
- Se o pagamento de horas semanais trabalhadas superar 40 horas, o pagamento de horas extras também pode ser determinado.
- Então as deduções são subtraídas do pagamento bruto para se obter o pagamento líquido.

## Saída:

- cheques de pagamentos a empregados, relatórios para gerentes e informações fornecidas para acionistas, bancos, agências governamentais e outros grupos.



Prof. Anderson Augusto Bosing

# VAMOS PRATICAR ?

## Desenvolvendo um SI para uma ONG

- Você está em contato com o coordenador de uma Organização Não Governamental chamada Parceiros Voluntários;
- A Idéia desta ONG é cadastrar profissionais voluntários (parceiros) e também instituições que necessitam dos serviços destes profissionais;
- Dessa forma, os profissionais cadastrados informariam sua especialidade, tempo e horário disponível, e as instituições a serem favorecidas pelos trabalhos voluntários, repassariam as informações necessárias para o encaminhamento de tais profissionais;
- O diretor da organização solicita o trabalho de sua equipe para desenvolver um Sistema de Informação para a ONG;
- Primeiramente descreva este sistema em termos de ENTRADAS – PROCESSAMENTOS e SAÍDAS, em seguida, identifique quais dados precisariam ser armazenados pelo sistema.



Prof. Anderson Augusto Bosing

# Engenharia de Software

**Aula: Processo e Ciclo de  
Vida de software.**



Prof. Anderson Augusto Bosing

# Qual é o valor da informação para uma empresa?



# Qual é o valor da informação para uma empresa?

Está surgindo um novo tipo de bem econômico: a **informação**.

A empresa que dispõe de mais informações sobre seu processo de negócio está em **vantagem**.

Em consequência, surgiu a necessidade de gerenciar informações de uma forma **adequada e eficiente**.



# **Qual é o valor da informação para uma empresa?**

O **objetivo** principal e final da construção de um sistema de informações é a adição de valor à empresa.

Isso implica que a produtividade nos processos da empresa deve aumentar de modo significativo.

**O sistema deve ser economicamente justificável.**



# Qual é o valor da informação para uma empresa?

- <https://adssettings.google.com/authenticated>



# **Qual é o valor da informação para uma empresa?**

**Uma característica é a complexidade de seu desenvolvimento, que aumenta à medida que o tamanho do sistema cresce.**



# **Qual é o valor da informação para uma empresa?**

Para a construção de sistemas de software mais complexos, é necessário um planejamento inicial. Equivalente ao projeto das plantas da engenharia civil também deve ser realizado. Essa necessidade leva ao conceito de modelo, tão importante no desenvolvimento de sistemas.



# **Gerenciamento de Complexidade**

**Pode haver diversos modelos de um mesmo sistema,  
cada qual descrevendo uma perspectiva.**

**Exemplo:** Um Carro pode ter um modelo para representar sua parte elétrica, outro para a parte aerodinâmica etc.



# **Gerenciamento de Complexidade**

**Por meio de modelos podemos fazer estudos e prever comportamentos do sistema em desenvolvimento.**

**Modelos revelam as características essenciais de um sistema. Detalhes não relevantes e que só aumentariam a complexidade do problema podem ser ignorados.**



# **Comunicação entre as Pessoas Envolvidas**

**O desenvolvimento de um sistema envolve a execução de uma grande quantidade de atividades.**

**Os modelos de um sistema servem para promover a difusão de informações relativas ao sistema entre os indivíduos envolvidos em sua construção.**



# **Redução dos Custos no Desenvolvimento**

**Seres humanos estão sujeitos a cometer erros.**

**Certamente a correção desses erros é menos custosa quando detectada é realizada ainda nos modelos do sistema.**

**O que é melhor?**

**Corrigir um erro na projeto de um carro ou recolher todos os carros depois de fabricados.**



# **Previsão do Comportamento Futuro do Sistema**

O comportamento do sistema pode ser discutido mediante uma análise dos seus modelos.

Os modelos servem como um “laboratório”, em que diferentes soluções para um problema relacionado à construção do sistema podem ser experimentadas.



# Processo de software

**Sommerville (2014) afirma que um “processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software”.**

**Pressman (2011) complementa dizendo que “processo é uma metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade”.**



# Processo de software

- É uma **série de passos** (um **ROTEIRO**).
- Para criar **EM TEMPO** um **SOFTWARE** de **ALTA QUALIDADE**, sem estourar o **ORÇAMENTO**.
- Como “escolher” um processo?
- ✓ As **CARACTERÍSTICAS DA APLICAÇÃO** (domínio do problema, tamanho, complexidade etc);
- ✓ A **TECNOLOGIA** a ser adotada na sua construção (paradigma de desenvolvimento, linguagem de programação, mecanismo de persistência etc), a organização;
- ✓ **ONDE** o produto será desenvolvido;
- ✓ **O PERFIL DA EQUIPE** de desenvolvimento.



Prof. Anderson Augusto Bosing

# Engenharia de Software - Ciclo de vida

- Independente da metodologia que será utilizada para desenvolvimento do software sempre existirá um ciclo de vida existente em todas as metodologias.
- Ciclo de vida pode ser definido como uma “estrutura contendo processos, atividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produto de software, abrangendo a vida do sistema desde a definição de seus requisitos até o término de seu uso”.



Prof. Anderson Augusto Bosing

# Atividade

**Em geral, um ciclo de vida de software envolve as etapas:**

- ✓ **Planejamento**
- ✓ **Análise e Especificação de Requisitos**
- ✓ **Projeto**
- ✓ **Implementação**
- ✓ **Testes**
- ✓ **Entrega e Implantação**
- ✓ **Operação**
- ✓ **Manutenção**

**Realizar pesquisa sobre cada uma das etapas acima:**

**Quais as principais atividades da etapa.**

**Quais são os profissionais de tecnologia da informação envolvidos nessa etapa.**

**Quais atividades são realizadas nesta etapa.**

**A etapa gera algum produto de trabalho como insumo para etapas posteriores.**



Prof. Anderson Augusto Bosing

# Ciclo de vida - Planejamento

- Fornece uma estrutura que possibilita ao gerente fazer estimativas iniciais de recursos, custos e prazos;
- O escopo do software é estabelecido;
- Um plano de projeto deve ser elaborado configurando o processo a ser utilizado;
- Esta atividade faz parte da **gerência de projeto**.



Prof. Anderson Augusto Bosing

# Ciclo de vida - Análise e Especificação de Requisitos

- O escopo do software é refinado;
- Nessa fase, a interação entre quem desenvolverá e o cliente é muito grande, contudo não se discute como será feito o software, mas sim o que ele deve fazer.
- Devem ser analisados o domínio do problema e o domínio da solução.
- São usadas as histórias de usuário (User Stories), que são pequenas histórias escritas para auxiliar na definição do requisito entre quem desenvolve e o cliente.
- Todo esse material é compilado e gera um relatório que servirá para os tomadores de decisão definirem, ou não, a continuidade e o desenvolvimento do software.



Prof. Anderson Augusto Bosing

# Ciclo de vida – Projeto

- Utiliza a fase anterior como insumo.
- Essa fase é voltada ao programador do software.
- Definição de como seria a interface que o usuário opera, quais cores seriam utilizadas na interface, como seria o fluxo de funcionamento de cada botão existente na interface, em qual banco de dados ficariam os dados gerados.
- O projeto, diferente da fase anterior, define como as coisas acontecerão.



Prof. Anderson Augusto Bosing

# Ciclo de vida – Implementação

- Essa fase deve traduzir o projeto em um software, utilizando ferramentas e linguagens adequadas.
- Dadas as inúmeras metodologias de desenvolvimento, cada programador pode desenvolver o código do sistema de uma forma diferente.



Prof. Anderson Augusto Bosing

# Ciclo de vida – Testes

- **Teste de unidade:** cada componente é testado individualmente, sem conexão com outros componentes.
  - ✓ Exemplo: testar apenas o método que faz a soma dos valores das vendas, dados dois números, ele deve retornar a soma dos dois.
- **Teste de módulo:** um módulo é um agrupamento de pequenos componentes, que tem uma função específica.
  - ✓ Exemplo: testar a geração do relatório de vendas de um produto.
- **Teste de subsistemas:** são testados módulos integrados que controlam as interfaces do sistema.
  - ✓ Exemplo: testar as interfaces do sistema de compra e venda.



Prof. Anderson Augusto Bosing

# Ciclo de vida – Testes

- **Teste de sistema:** testa a integração dos subsistemas que formam o sistema principal.
  - ✓ Exemplo: realizar login, operações e geração de relatórios em um sistema.
- **Teste de aceitação:** testes realizados com dados reais fornecidos pelos clientes. Último teste antes de colocar o sistema em operação.
  - ✓ Exemplo: quando há muitos usuários na base e o login não acontece de forma instantânea.



Prof. Anderson Augusto Bosing

# Ciclo de vida – Entrega e Implantação

- O software deve ser instalado em ambiente produção.
- Envolve:
  - ✓ Treinamento de usuários;
  - ✓ Configuração do ambiente de produção;
  - ✓ Conversão bases de dados (se necessário).



Prof. Anderson Augusto Bosing

# Ciclo de vida – Operação

**Após os testes, entrega e implantação, o software passa a ser utilizado de fato em um ambiente de produção.**



Prof. Anderson Augusto Bosing

# Ciclo de vida – Manutenção

- **Manutenção corretiva:**
  - ✓ Correção dos erros encontrados pelo cliente e que não foram detectados nas fases de testes anteriores.
- **Manutenção adaptativa:**
  - ✓ Adaptação do software relacionado as mudanças do ambiente externo.
- **Manutenção evolutiva:**
  - ✓ Mudanças não previstas nos requisitos originais, visando a melhorias de desempenho e novas funcionalidades.
- **Manutenção preventiva:**
  - ✓ A iniciativa parte da equipe de desenvolvedores, visando evitar futuros problemas e melhorias em futuras manutenções.

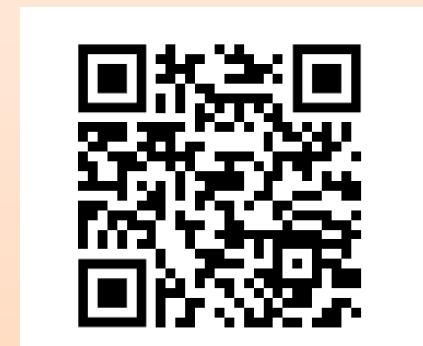


Prof. Anderson Augusto Bosing

# Engenharia de Software - Ciclo de vida

➤ Em geral, um ciclo de vida envolve as etapas:

- ✓ Planejamento
- ✓ Análise e Especificação de Requisitos
- ✓ Projeto
- ✓ Implementação
- ✓ Testes
- ✓ Entrega e Implantação
- ✓ Operação
- ✓ Manutenção



Prof. Anderson Augusto Bosing