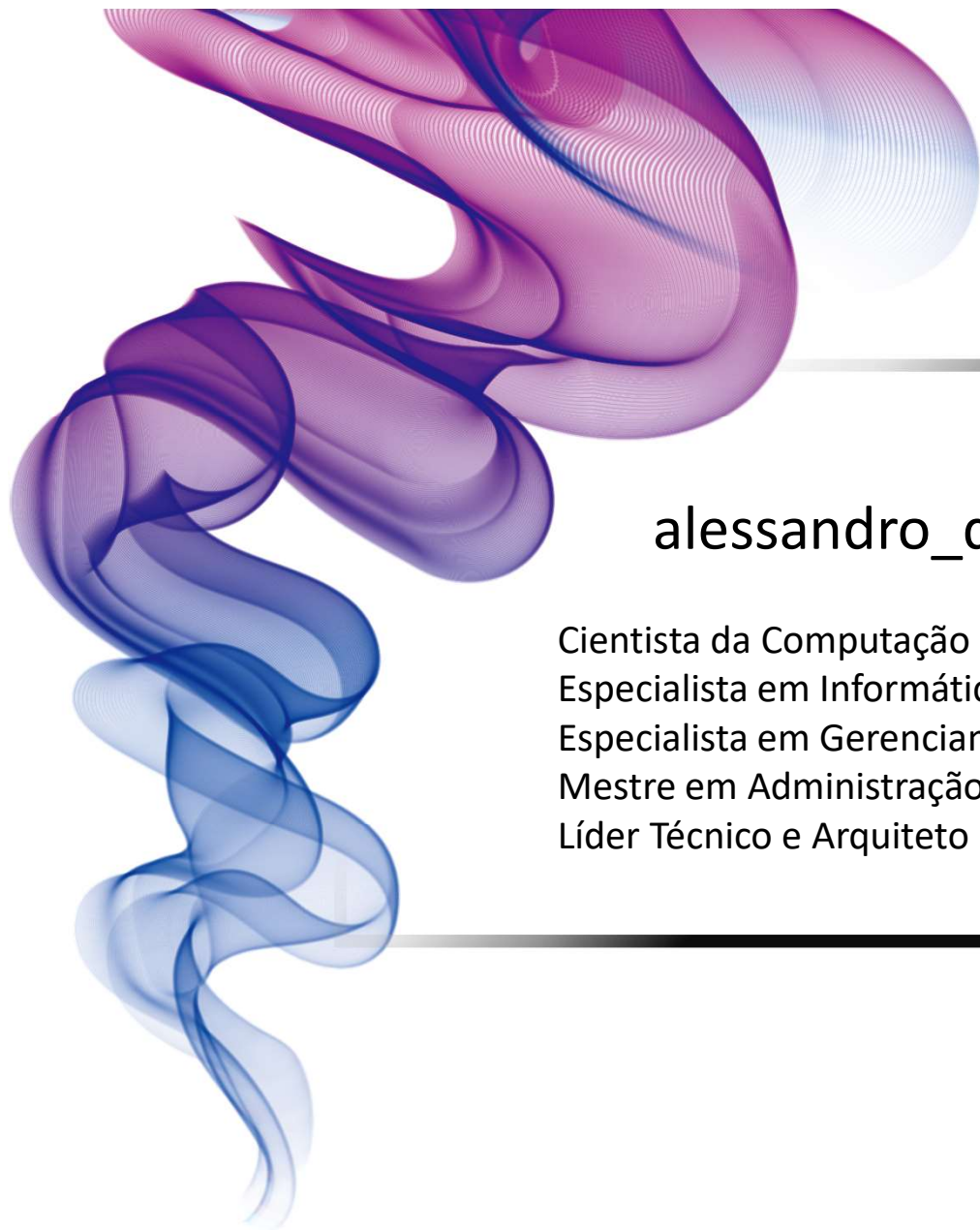




# Engenharia de Software

Material baseado em produção do professor Jean Paul Lopes



[alessandro\\_dias@uniritter.edu.br](mailto:alessandro_dias@uniritter.edu.br)

Cientista da Computação - PUCRS

Especialista em Informática na Educação - PUCRS

Especialista em Gerenciamento de Projetos de TI - PUCRS

Mestre em Administração e Negócios - PUCRS

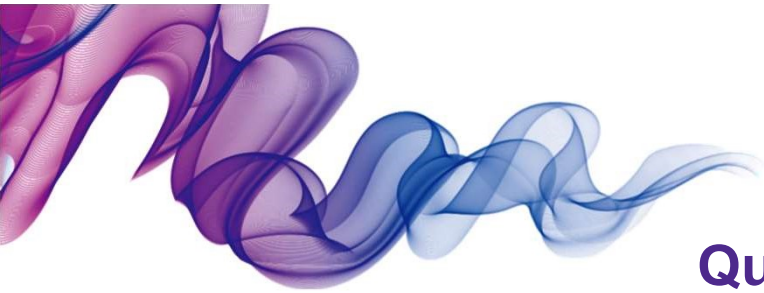
Líder Técnico e Arquiteto na e-Core

# Objetivos

- Descrever o cenário atual da indústria de software
- Compreender o processo de desenvolvimento de software
- Identificar os princípios necessários e das qualidades desejadas no processo de desenvolvimento de software.
- Analisar os aspectos envolvidos na engenharia de requisitos, no gerenciamento e na qualidade de projetos de software.
- Avaliar processos de desenvolvimentos de software, tais como modelos tradicionais e modelos de desenvolvimento ágil e avaliar sua aplicabilidade no contexto de negócio.
- Analisar os modelos de processo de desenvolvimento de software

# Objetivos

- Identificar, analisar e documentar requisitos e regras de negócio de um software.
- Identificar os vários modelos de ciclo de vida e seu efeito na prática da produção de software.
- Conhecer e saber aplicar métodos e ferramentas de especificação de sistemas de informação.
- Conhecer os conceitos de projeto de sistemas de informação e capacitar-se na utilização de seus métodos, técnicas e ferramentas.
- Identificar as etapas de implementação, teste e manutenção de sistemas de computação e ser capaz de realizá-los e/ou coordená-los.
- Conhecer e saber aplicar métodos de controle da qualidade do processo de software.
- Desenvolver produtos de software



## Qual é a definição de Engenharia?

“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. É também profissão em que se adquire e se aplicam os conhecimentos matemáticos e técnicos na criação, aperfeiçoamento e implementação de utilidades que realizem uma função ou objetivo.

Nos processos de criação, aperfeiçoamento e complementação, a engenharia conjuga os vários conhecimentos especializados no sentido de viabilizar as utilidades, tendo em conta a sociedade, a técnica, a economia e o meio ambiente.”

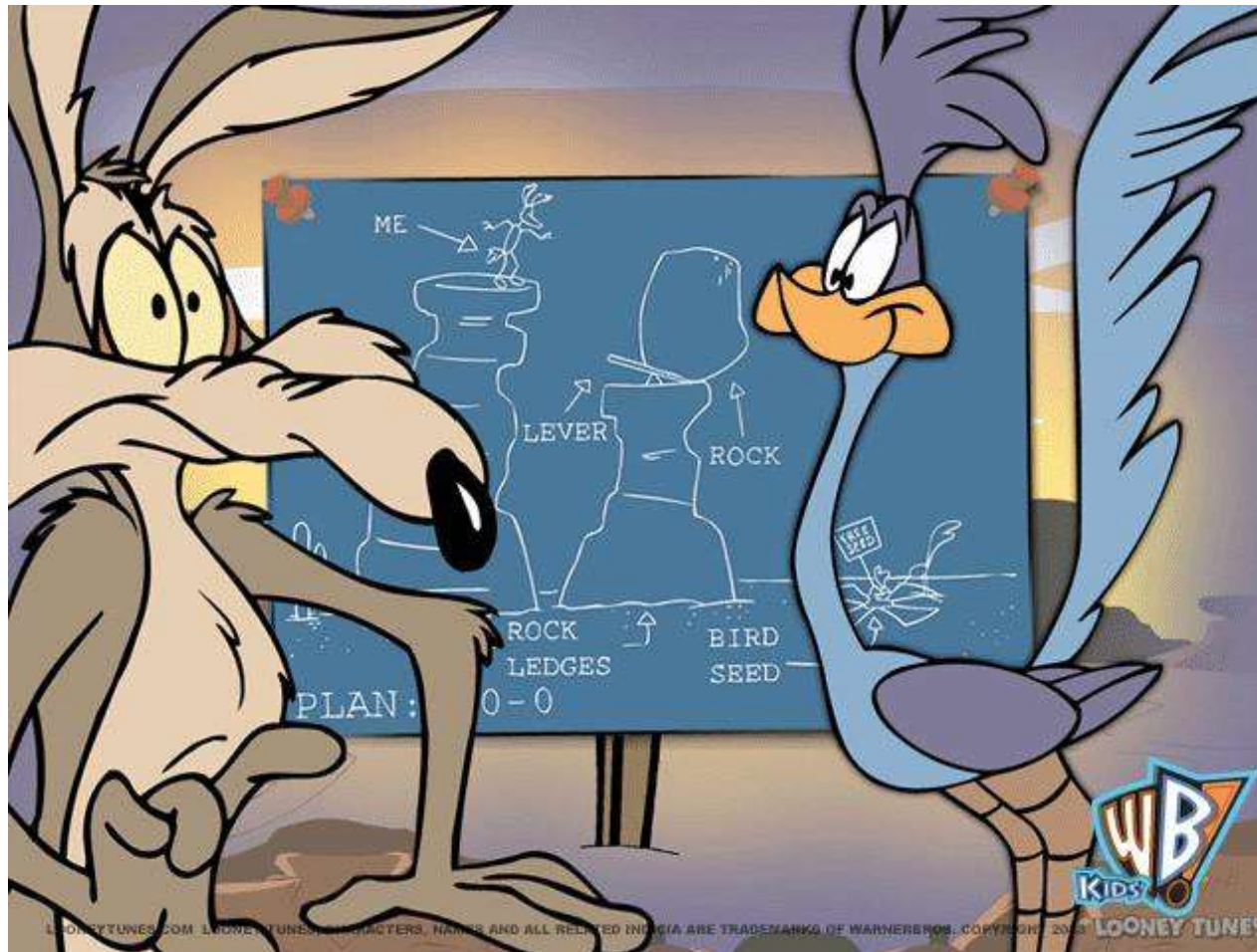
Fonte: wikipedia

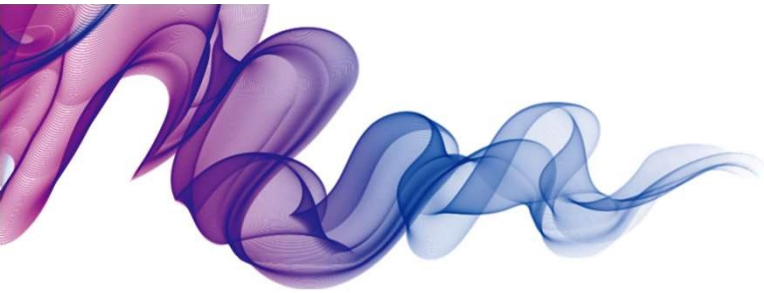


# Quebra Gelo



**Nada supera um bom planejamento  
e um bom projeto**





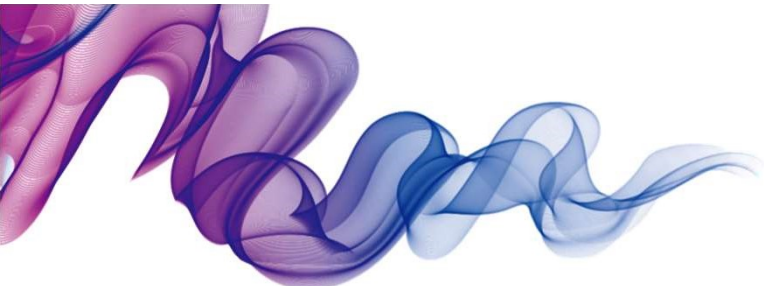
### **A IEEE define engenharia de software como**

“the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.”

“o uso de uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, operação e manutenção de software; ou seja, a aplicação de engenharia ao software.”

SWEBOK 3.0, 2004





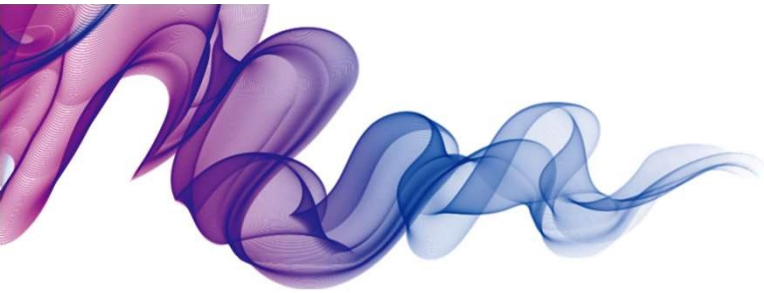
## mais definições:

“Engenharia de Software é uma disciplina de engenharia cujo o foco está em todos os aspectos da produção de software, desde os estágios iniciais da especificação do sistema até sua manutenção, quando o sistema já está sendo usado.”

“A engenharia de software tem por objetivo apoiar o desenvolvimento profissional de software. ... Quando falamos em ES não se trata somente do programa em si, mas de toda a documentação associada e dados de configurações necessários para fazer esse programa operar corretamente.”

“ES tem a ver com obter resultados de qualidade requeridos dentro do cronograma e do orçamento. Isto frequentemente envolve compromissos – **engenheiros não podem ser perfeccionistas.**”

SOMMERVILLE, 2011



## **Qual a diferença entre Engenharia de Software e Ciência da Computação ?**

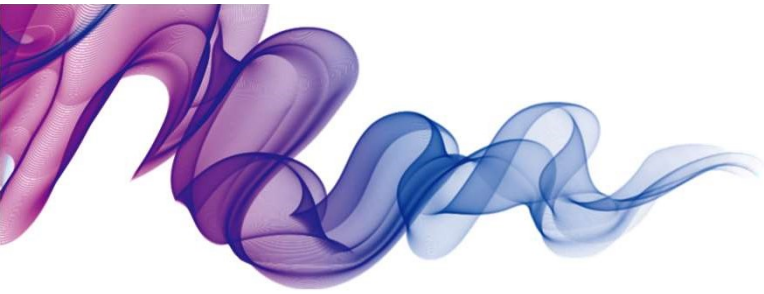
**A ciência da computação foca na teoria e nos fundamentos.**

**A engenharia de software preocupa-se com o lado prático do desenvolvimento e entrega de softwares práticos.**



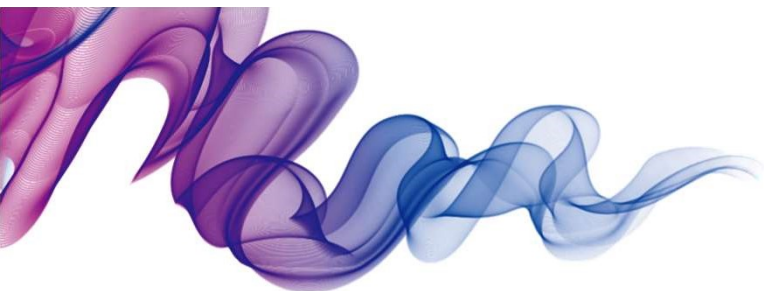
## E qual a diferença entre programação e Engenharia de Software?

Programação	Eng. Software
<ul style="list-style-type: none"><li>• Sem planejamento;</li><li>• Sistemas pequenos;</li><li>• Poucas pessoas envolvidas;</li><li>• Não se preocupa com padronização e nem mesmo documentação;</li><li>• Pouco impacto em seu entorno;</li></ul>	<ul style="list-style-type: none"><li>• Planejamento meticuloso;</li><li>• Sistemas grandes;</li><li>• Desenvolvido por um time;</li><li>• Leva muito em conta a padronização e tem que ser muito documentado;</li><li>• Muito impacto em seu entorno;</li><li>• Deve estar apoiado em um processo que:<ul style="list-style-type: none"><li>• Especifique os documentos a serem produzidos;</li><li>• Indique o que fazer, quando fazer e se possível, como fazer;</li><li>• Especifique o ciclo de vida de um sistema;</li></ul></li></ul>



## Fundamentos da Engenharia de Software (aplicável a todos os tipos de SW)

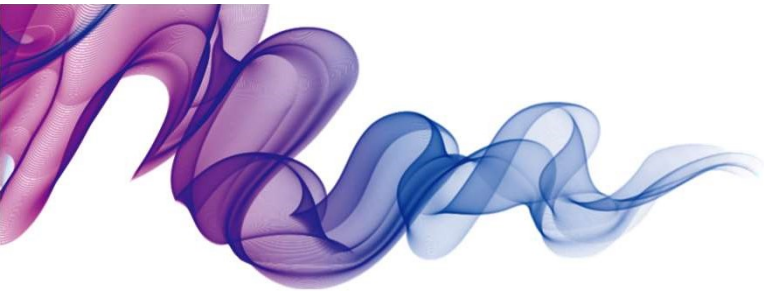
- Devem ser desenvolvidos em um processo gerenciado e compreendido;
- Confiança e desempenho são importantes para todos os tipos de sistema;
- Entender e gerenciar as especificações e requisitos de software;
- Deve-se fazer o melhor uso possível dos recursos existentes.



## Atributos essenciais para um bom software

<b>Manutenibilidade</b>	Deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes. É um atributo crítico, pois a mudança em um software é inevitável em um ambiente de negócio em mudança.
<b>Confiança e proteção</b>	Um software não deve causar prejuízos físicos ou econômicos no caso de falha; usuários maliciosos ou não autorizados não devem ser capazes de acessar e/ou prejudicar o sistema.
<b>Eficiência</b>	Não deve desperdiçar os recursos do sistema (memória, ciclo de processamento). Inclui capacidade de resposta, tempo de processamento, uso de memória/disco, etc.
<b>Aceitabilidade</b>	Deve ser aceitável para o tipo de usuário para o qual foi projetado. Deve ser compreensível, usável e compatível aos demais sistemas usados por ele.





Revendo a definição, engenharia de software ...

**“Abrange um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos – que possibilita ao gerente o controle do processo de desenvolvimento de software e oferece ao profissional uma base para a construção de software de alta qualidade produtivamente”**

**Pressman, 1995.**

