

# Engenharia de Requisitos de Software

3º período

Professora: Michelle Hanne Soares de Andrade

[mhsandrade@sga.pucminas.br](mailto:mhsandrade@sga.pucminas.br)

[michellehanne.andrade@gmail.com](mailto:michellehanne.andrade@gmail.com)

# Apresentação

*São mais de 25 anos de experiência no segmento de Computação e Tecnologia da Informação, atuando na análise, coordenação e desenvolvimento de sistemas. Concomitante com a experiência prática, também atuo na educação com 14 anos de experiência em treinamentos de formação contínua, ensino técnico, graduação e pós-graduação. Mestre em Ciência da Computação pela UFMG e especialista em Inteligência e Constrainteligência pela Faculdade Pitágoras. Liderei e participei da equipe de implantação do Sistema Gestão em Saúde em Rede PBH e Sistema de Gestão de Convênios da Secretaria de Transporte e Obras Públicas de Minas Gerais. Atuei como micro empresário no desenvolvimento de sistemas nos segmentos de varejo, medicina estética e gestão comercial. Expertise em Ciência de Dados, Machine Leraning, Engenharia de Software e Transformação Digital.*

**Linkedin:** <https://www.linkedin.com/in/michellehanne/>

**Blog:** <http://ormuztech.com.br/blog/>

**Experiência Profissional  
25 anos**

**Experiência Docente  
14 anos (graduação e pós)**

**Assuntos de interesse**  
Inteligência Artificial, Transformação Digital, Indústria  
4.0, Business Agility, Inferência Estatística...



**Contatos:**  
[mhsandrade@sga.pucminas.br](mailto:mhsandrade@sga.pucminas.br)  
[michellehanne.andrade@gmail.com](mailto:michellehanne.andrade@gmail.com)

(31) 98543-0198 (WhatsApp)

# Grupo de Oportunidades de TI



<https://chat.whatsapp.com/BabWyV4Ha5yCMcGTVXLNcm>



**Comunidade - Oportunidades TI**

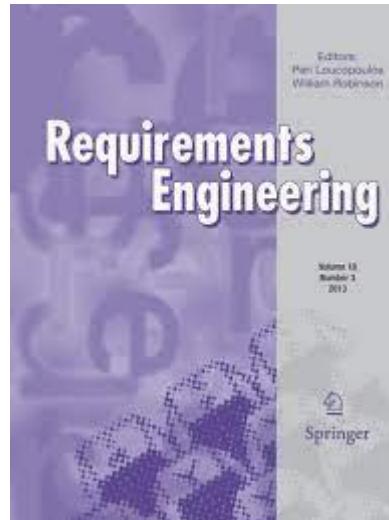
# Bibliografias



VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira. **Engenharia de requisitos: software orientado ao negócio.** Rio de Janeiro (RJ) : Brasport, c2016. E-book



ORGANIZADOR EDUARDO SANTOS KERR. **Gerenciamento de Requisitos.** Pearson 212.



REQUIREMENTS ENGINEERING. Basel, Suíça: Springer,. Trimestral (de 3 em 3 meses).

Disponível em  
<https://link.springer.com/journal/766>



WAZLAWICK, Raul Sidnei. Análise e design orientados a objetos para sistemas de informação: modelagem com UML, OCL e IFML 3ª Edição LTC Raul Sidnei Wazlawick



# Ementa

- Conceitos de Engenharia de software.
- Conceitos e tipos de Processos de Software.
- Engenharia de requisitos.
- Métodos e técnicas para análise e especificação de softwares.
- Introdução às estimativas de software .

# Objetivos

- Desenvolver habilidade e **competência para especificação de requisitos, através de técnicas de interação para levantamento, identificação e documentação.**
- Capacitar para modelagem de sistemas através do uso de métodos e **técnicas orientadas por objetos (UML).**
- Compreender os **objetivos da especificação de requisitos e modelagem** dentro das fases que compõem o processo de desenvolvimento de software.
- Capacitar para uso de **técnicas de levantamento e registro de histórias de usuário** em métodos ágeis.
- Diferenciar as abordagens do processo de software em **cascata, prescritivo (Processo Unificado) e processo ágil (SCRUM).**
- Compreender as técnicas para **Estimativa de Software.**

# Unidades de Ensino

- Unidade 0 - Nivelamento (2h)
  - Nivelamento sobre a compreensão de processos de negócio e conceitos iniciais de requisitos de software .
- Unidade I - Processo de Software e Introdução a Requisitos (16h)
  - Tipos de Processo de Software: cascata, processo unificado e abordagem ágil (SCRUM). Contextualização do Processo de Requisitos no âmbito do Processo de Software.
  - Requisito. Requisitos funcionais, não funcionais, de cliente e do produto.
  - Processo de Engenharia de Requisitos.

# Unidades de Ensino

- Unidade II - Elicitação de Requisitos (20 h).
  - Elicitação de requisitos, Técnicas de elicitação de requisitos. Elicitação de requisitos em métodos ágeis .
- Unidade III - Detalhamento e Modelagem de Requisitos (22h).
  - Casos de uso. Elementos de um caso de uso. Regras de negócio. Documentação de protótipos de interface.
- Unidade IV - Verificação, validação e gerênciade requisitos (12h).
  - Técnicas de verificação e validação de requisitos. Técnicas de inspeção. Mecanismos de controle de mudanças. Rastreabilidade de requisitos. Ferramentas de registro e controle de requisitos .
- Unidade V - Introdução às Estimativas de Software (8 h)



# Sumário

- Conceitos iniciais de Engenharia de Software

# Engenharia de Software

Engenharia de *software* é uma disciplina cujo o foco está em todos os aspectos da produção de *software* – desde a especificação do sistema até a sua manutenção.



*Imagem: commons.wikimedia.org*

# Engenharia

**Engenharia:** arte de aplicar conhecimentos científicos, conhecimentos empíricos e disciplinas específicas à criação de processos

**Objetivo:** atender as necessidades humanas (software de entretenimento, por exemplo)

# O que é Software

“Um Software é desenvolvido ou passa por um processo de Engenharia; ele não é fabricado no sentido clássico” (Pressmann, 2016).

# Processos de Software

“Um processo de software é um conjunto de atividades relacionadas que levam à produção de um produto de software.”  
(Sommerville, 2010)

# Engenharia de Software

**Desafio do engenheiro de software:** escolher e montar as estruturas que a computação e a arte permitem realizar



# Processos de Software

Processos de desenvolvimento:  
maneiras pelas quais criamos os  
jogos, respeitando certos padrões  
ao longo do tempo.

# Processos de Software

**Existem muitos processos de software diferentes, mas todos devem incluir quatro atividades fundamentais para a engenharia de software (SOMMERVILLE, 2011):**

- 1. Especificação de software:** A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.
- 2. Projeto e implementação de software:** O software deve ser produzido para atender às especificações.
- 3. Validação de software:** O software deve ser validado para garantir que atenda às demandas do cliente.
- 4. Evolução de software:** O software deve evoluir para atender às necessidades de mudança dos clientes.

# Processos de Software

Processos de  
desenvolvimento:

- Baseados em ações sistemáticas, metódicas, planejadas (não no improviso)
- Não só para o desenvolvimento, mas também na operação e manutenção do jogo (servidores online, upgrades, etc...)

# Engenharia de Software



Como o cliente explicou



Como o líder de projeto entendeu



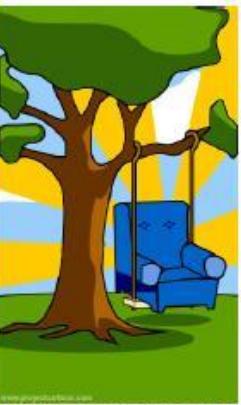
Como o analista planejou



Como o programador codificou



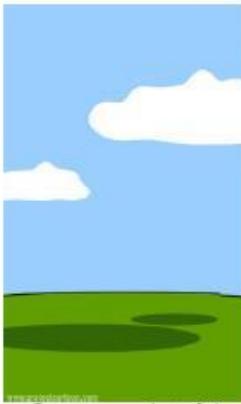
O que os beta testers receberam



Como o consultor de negócios descreveu



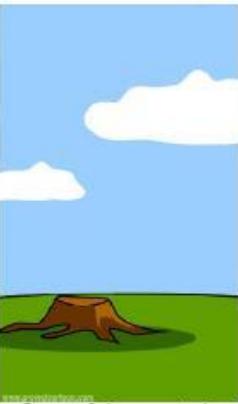
Valor que o cliente pagou



Como o projeto foi documentado



O que a assistencia técnica instalou



Como foi suportado



Quando foi entregue



O que o cliente realmente necessitava

# Processos de Software

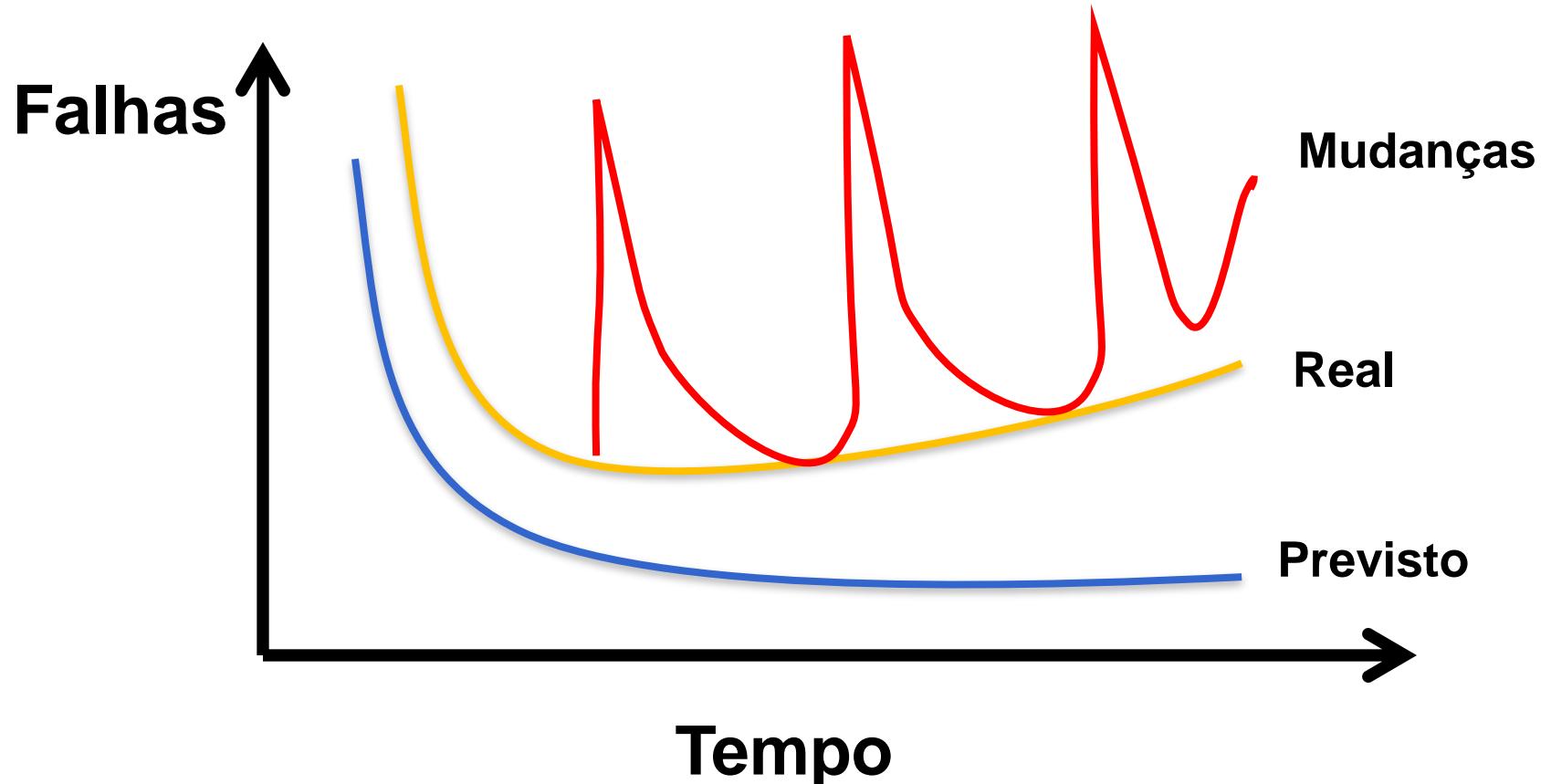
**“Software não se desgasta, mas sim deteriora” (Pressmann, 2016).**

Jogos não se desgastam

- **Eles se deterioraram com o tempo**
- **Causas:**
  - Mudanças na plataforma (Sistema operacional)
  - Problemas com drivers obsoletos
  - Surgimento de novos requisitos e demandas (Gráficos)
  - Patchs excessivos (Bugs, remendos)

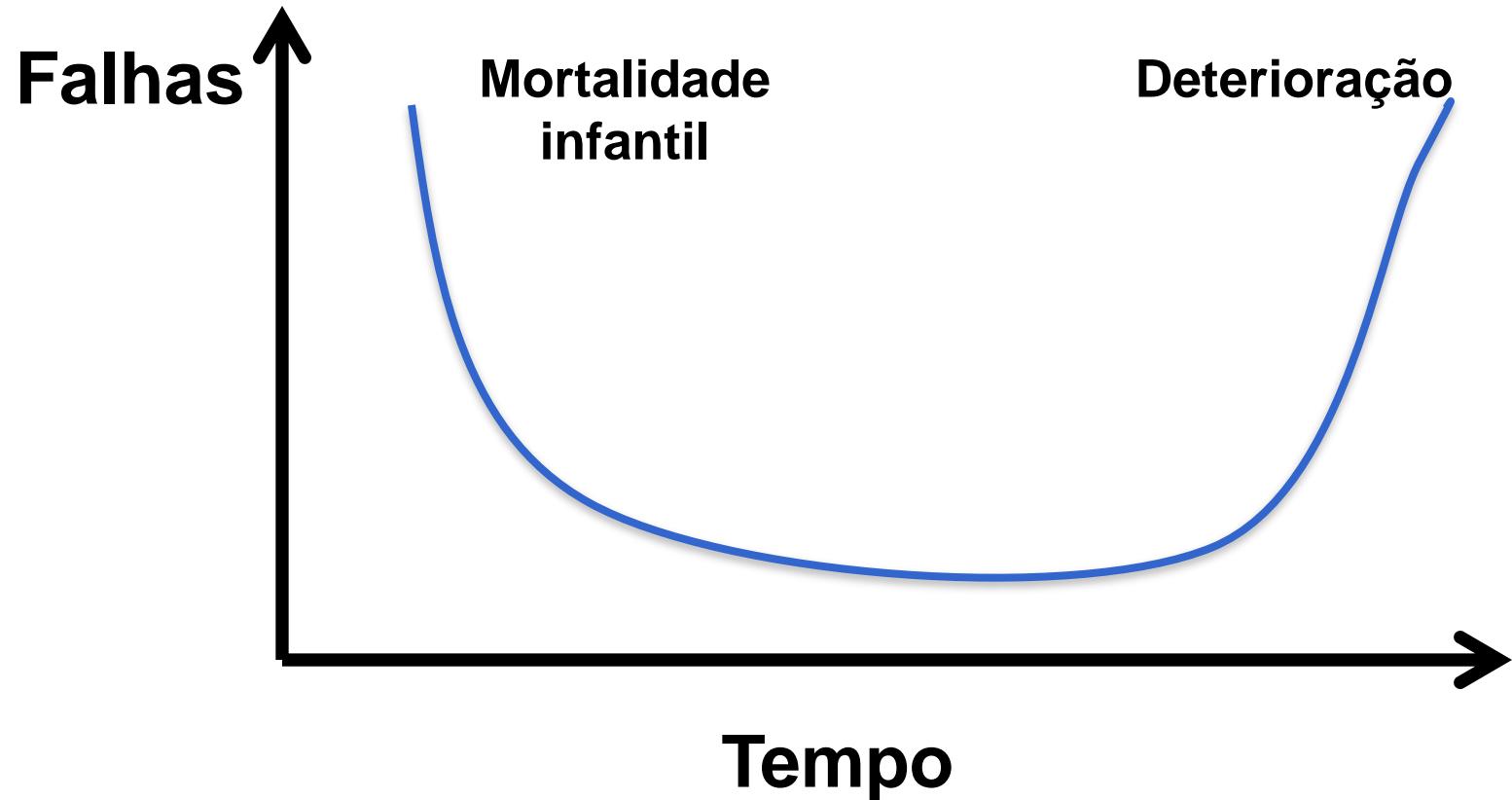
# Processos de Software

- Curva de falhas



# Processos de Software

- Curva de falhas



# Tipos de Software

- **Software de Sistema** – Ex: Compilador, drivers, Utilitários, etc.
- **Software de Aplicação** – Ex: ERP, Logística, Acadêmico, etc.
- **Software Científico/Engenharia** – Ex: Astronomia (processamento numérico pesado)
- **Software Embutidos/Embarcados**– Ex: Sistema eletrônico de carros, telefones, modems, robôs, eletrodomésticos, brinquedos, sistemas de segurança, etc
- **Aplicações WEB** - Ex: E-commerce, portais, sites e Apps, etc.
- **Software de IA** – Ex: Sistemas especialistas, robótica, Reconhecimento e Padrões, Redes Neurais Artificiais, Teoria de Jogos, etc.

# Linhas de produtos de software

Uso de técnicas de engenharia que permitem criar um grupo de *softwares* similares a partir de um conjunto de características comuns a todos esses sistemas.

“A reutilização de software se baseia no uso de conceitos, produtos ou soluções previamente elaboradas ou adquiridas para criação de um novo software, visando melhorar significativamente a qualidade e a produtividade”.

# Referências

PFLEEGER, S. L. **Engenharia de software: teoria e prática.** Belo Horizonte: Prentice Hall, 2004.

PRESSMAN, Roger S. MAXIM, Bruce R. **Engenharia de Software - Uma Abordagem Profissional.** 8.ed. Porto Alegre: Amgh Editora, 2016. 968p. ISBN 9788580555332.

SOMMERVILLE, Ian. **Engenharia de software.** 8.ed. São Paulo: A. Wesley publishing company, 2010.