

UNAMA



# Ciência da Computação

## MATURIDADE EM DESENVOLVIMENTO DE SOFTWARE

---

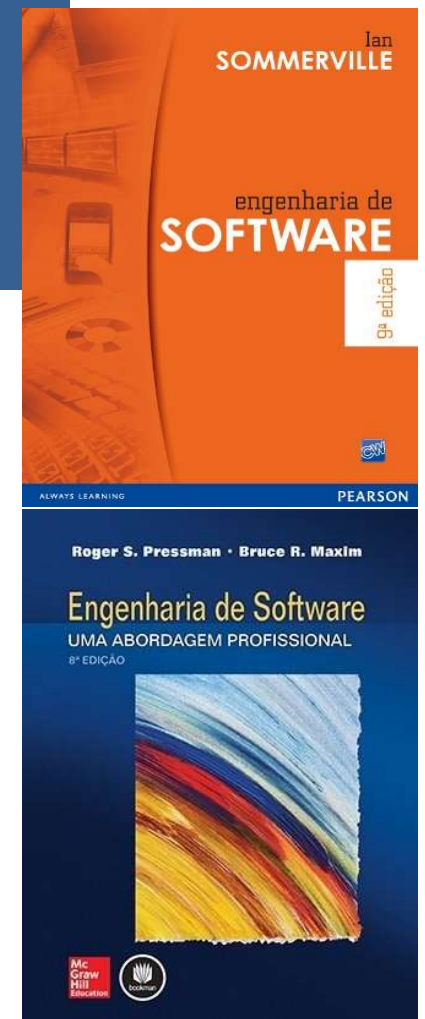
Prof. Dr. Jorge Cardoso

*[jcardoso217@gmail.com](mailto:jcardoso217@gmail.com)*

# Bibliografia Básica

SOMMERVILLE, IAN. Engenharia de Software - 8a edição - Versão em Português editada pela Addison Wesley.

PRESSMAN, ROGER S. Engenharia de Software. 6.ed. Rio de Janeiro: MacGraw-Hill, 2006.



# Agenda

- Ementa
- Conteúdo Programático
- Metodologia Avaliativa
- Introdução à Maturidade de desenvolvimento de software.

- Introdução ao modelo de maturidade de capacidade, Compreender os cinco níveis de maturidade de um software, Introdução à maturidade de capacidade integrada, Introdução à maturidade de capacidade de serviço e Compreensão da estrutura interna do modelo de maturidade de capacidade (CMM).

# Competências Específicas

- Entender a metodologia de gerenciamento de software.
- Fornece estrutura para organizar o software nas cinco etapas evolutivas de maturidade.

- **Unidade I:**
  - Introdução o Modelo de Maturidade de Capacidade (CMM)
  - Os Cinco Níveis de Maturidade

- **Unidade II:**
  - Modelo de Maturidade de Capacidade Integrado (CMMI)
  - Diferença entre CMMI E CMM?
  - Avaliação CMMI

- **Unidade III:**
  - Modelo de Maturidade de Capacidade de Serviço (CMM)
  - Melhoria do processo de software e determinação de capacidade - ISO 15504



- **Unidade IV:**
  - Estrutura interna do CMM
  - Limitações dos modelos CMM
  - Por que usar o CMM?
  - Quanto tempo leva para implementar o CMM?

# Introdução

# Histórico

- Há algum tempo a produção de software era feita sem nenhum **processo definido**. Não havia **controle** algum sobre possíveis manutenções ou possibilidade de **integração** com outros módulos. Os sistemas antigos tinham um ciclo de vida curto e elevados custos.

- À medida que a complexidade dos projetos aumentava, novas tecnologias iam surgindo, demandando alguns ajustes nos processos de produção e evitando que as empresas tivessem custos altíssimos devido ao desenvolvimento de sistemas realizado de forma completamente ad-hoc. Com isso, a engenharia de software foi em busca de um modelo que pudesse definir uma uniformidade para esses processos [KNEUPER 2009].

- Com essa evolução, conceitos importantes surgiram gradativamente, tais como divisão de software, arquitetura (top-down e bottom-up), diagramas e modelagens, conduzindo a engenharia de software ao estado atual.
- Com essa padronização, permitiu-se a oportunidade de mensurar alguns atributos de maneira mais precisa e segura, além de permitir mensurar o tamanho do software antes de sua construção. No entanto, mesmo com esse avanço, novos fatores surgiam aumentando a complexidade de produção para software.

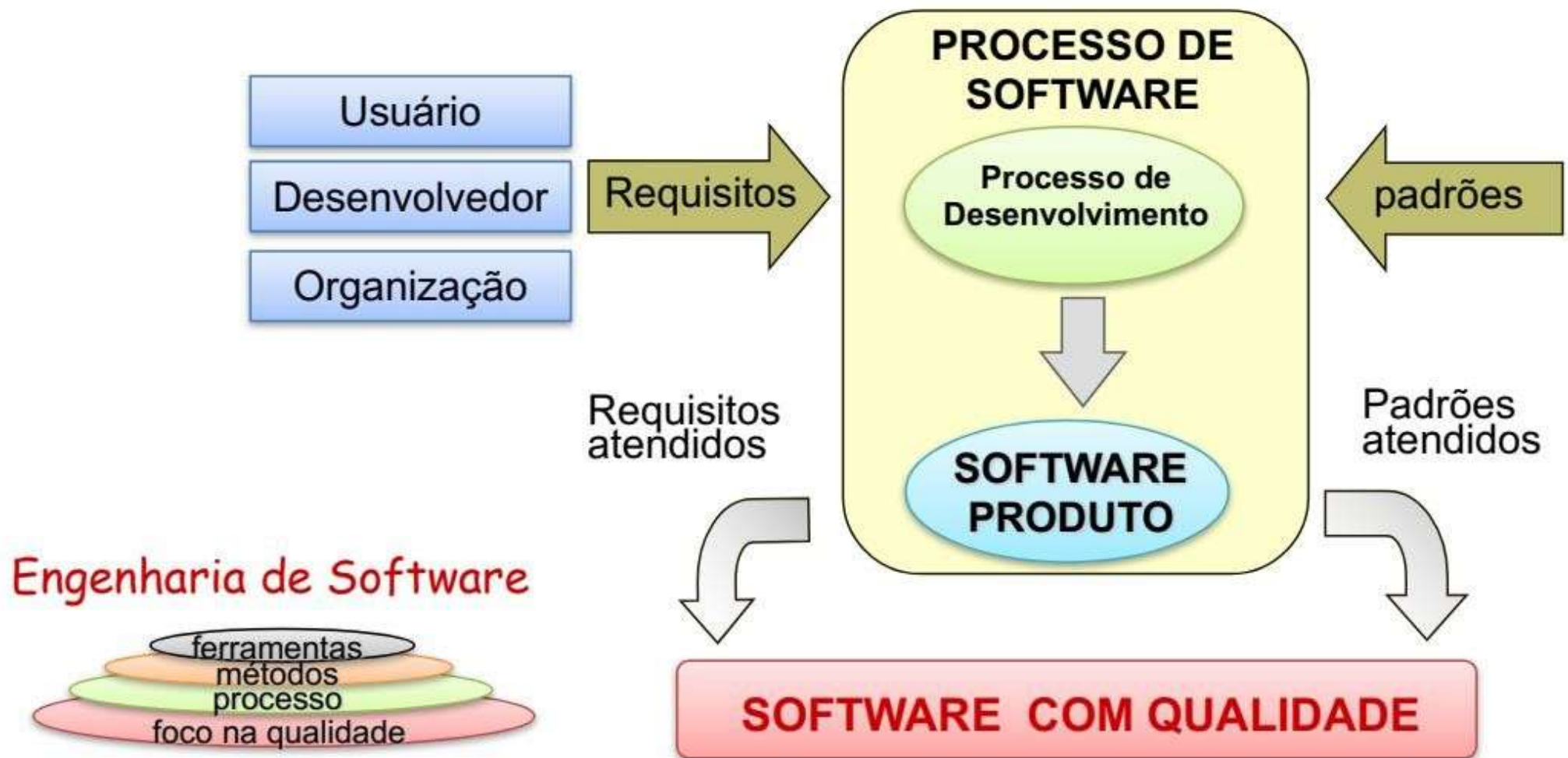
# Introdução

- O problema da qualidade de software
  - Insatisfação dos usuários
  - Atrasos de entrega
  - Correções dispendiosas
- A qualidade dos produtos depende da qualidade dos processos

# Um pouco de história...

- Na década de 80, o Instituto de Engenharia de Software (SEI) foi criado
- Objetivos
  - Fornecer software de qualidade para o Departamento de Defesa dos EUA
  - Aumentar a capacitação da indústria de software

# Base da Qualidade





# Qualidade do Processo de Software

- Antes de tudo.. O que é processo de software?



# Processo de Software

- Consiste em uma série de **atividades, práticas, eventos, ferramentas e métodos** que garantem, técnica e administrativamente que o software pode ser desenvolvido com qualidade e de maneira organizada, disciplinada e previsível.

# Modelos de Processo de Software

- Uma das maiores dificuldades encontradas pelas empresas de software é o gerenciamento de seus processos de software.



**Modelos de Processo de Software**

# Modelos de Processo de Software

- Um modelo de processo procura descrever formalmente e de maneira organizada todas as atividades que devem ser seguidas para a obtenção segura de um produto de software.
- É importante escolher um modelo apropriado às metas da organização e saber o grau em que esse modelo será implementado.
- Modelos CMMI e ISO/IEC 12207 são relacionados a processo de software.

# Um Processo de Software Imaturo

- **Ad hoc:** processo improvisado por profissionais e gerências.
- Não é rigorosamente seguido e o cumprimento não é controlado.
- Altamente dependente dos profissionais atuais.
- Baixa visão do progresso e da qualidade.

# Um Processo de Software Imaturo

- A qualidade do produto pode ficar comprometida para que prazos sejam cumpridos.
- Arriscado do ponto de vista do uso de nova tecnologia.
- Custos de manutenção excessivos.
- Qualidade difícil de se prever.

# Um Processo de Software Maduro

- Coerente com as linhas de ação, o trabalho é efetivamente concluído.
- Definido, documentado e melhorando constantemente:
  - compreendido
  - utilizado
  - vivo e ativo

# Um Processo de Software Maduro

- Com o apoio visível da alta administração e outras gerências.
- Bem controlado - fidelidade ao processo é objeto de auditoria e de controle.
- São utilizadas medições do produto e do processo.
- Uso disciplinado da tecnologia.



# Maturidade de Processo de Software

- A qualidade do processo de software pode ser analisada por meio do **nível de maturidade do processo.**
- A maturidade dos processos de software de uma organização influencia na sua capacidade de atingir metas de custo, qualidade e cronograma.

# Valor da Melhoria de Processo

- A melhoria do processo de software oferece um retorno no investimento que pode ser medido - quando é medido.
- Benefícios adicionais são intangíveis e não podem ser quantificados facilmente.

- **Qualidade do Produto**

- ISO 9126 e ISO 25010 – Qualidade de produto de software
- ISO/IEC 9126 (NBR 13596)
- ISO/IEC 12119
- ISO/IEC 14598-5

- **Qualidade do Processo**

- **CMM e CMMI**
- ISO 12207 - Qualidade do processo de software
- ISO 15504 (SPICE)
- MPS.BR

# Capability Maturity Model (CMM)

- Modelo de Maturidade de Processo de Software que auxilia as empresas a melhorar seus processos de software.
- 1986 - Software Engineering Institute (SEI) por solicitação do DoD que necessitava de um método de avaliar a capacidade de seus fornecedores de software.
- 1991 - 1a versão do CMM
- 2002 - 1a versão do CMMI (Capability Maturity Model Integration)
- 2010 - CMMI Versão 1.3
- <http://www.sei.cmu.edu/cmml/>  
<http://cmminstitute.com/>

e

# Capability Maturity Model (CMMI)

- O primeiro ponto importante a se observar no modelo CMMI:
- É um modelo que se refere ao processo pelo qual o produto é construído.
- CMMI provê modelos para três áreas de interesse:
  - Desenvolvimento de produtos e serviços (CMMI for Development)
  - Aquisição (CMMI for Acquisition)
  - Prestação de serviços (CMMI for Services)

# Capability Maturity Model (CMMI)

- Supõe-se que uma organização que possui um processo maduro tem maiores probabilidades de produzir bons produtos consistentemente do que uma outra cujo processo é imaturo e caótico.

# Capability Maturity Model (CMMI)

- O CMMI é um modelo de referência para a qualidade de processo de produção de software.
- Por meio de um processo formal de avaliação, uma organização é classificada em um “nível de maturidade” que varia de um a cinco.

# CMMI: nível de maturidade

- O nível de maturidade indica em que medida os processos daquela organização são maduros.
- Quanto maior o nível de maturidade, melhores e mais maduros são os processos.



# Dúvidas??

