



UNIVERSIDADE  
FEDERAL DO CEARÁ  
CAMPUS DE RUSSAS

# Engenharia de Software

## Modelos de Processos Prescritivos (Tradicionais)

Profa. Dra. Anna Beatriz Marques



# Modelos de processo de software

- » Fornecem um guia específico para o trabalho de engenharia de software
- » Define o fluxo de atividades, ações e tarefas, o grau de iteração, os artefatos e a organização do trabalho a ser feito



# Modelos de processo de software

## Quem realiza?

- » Os engenheiros de software e seus gerentes adaptam um modelo de processo e o seguem

## Por que é importante?

- » O desenvolvimento sem controle torna-se caótico.
- » O processo propicia estabilidade, controle e organização.

## Quais são as etapas envolvidas?

- » Os passos necessários para realizar um trabalho de engenharia de software disciplinado.

Como podemos organizar as etapas do processo de software da melhor forma?



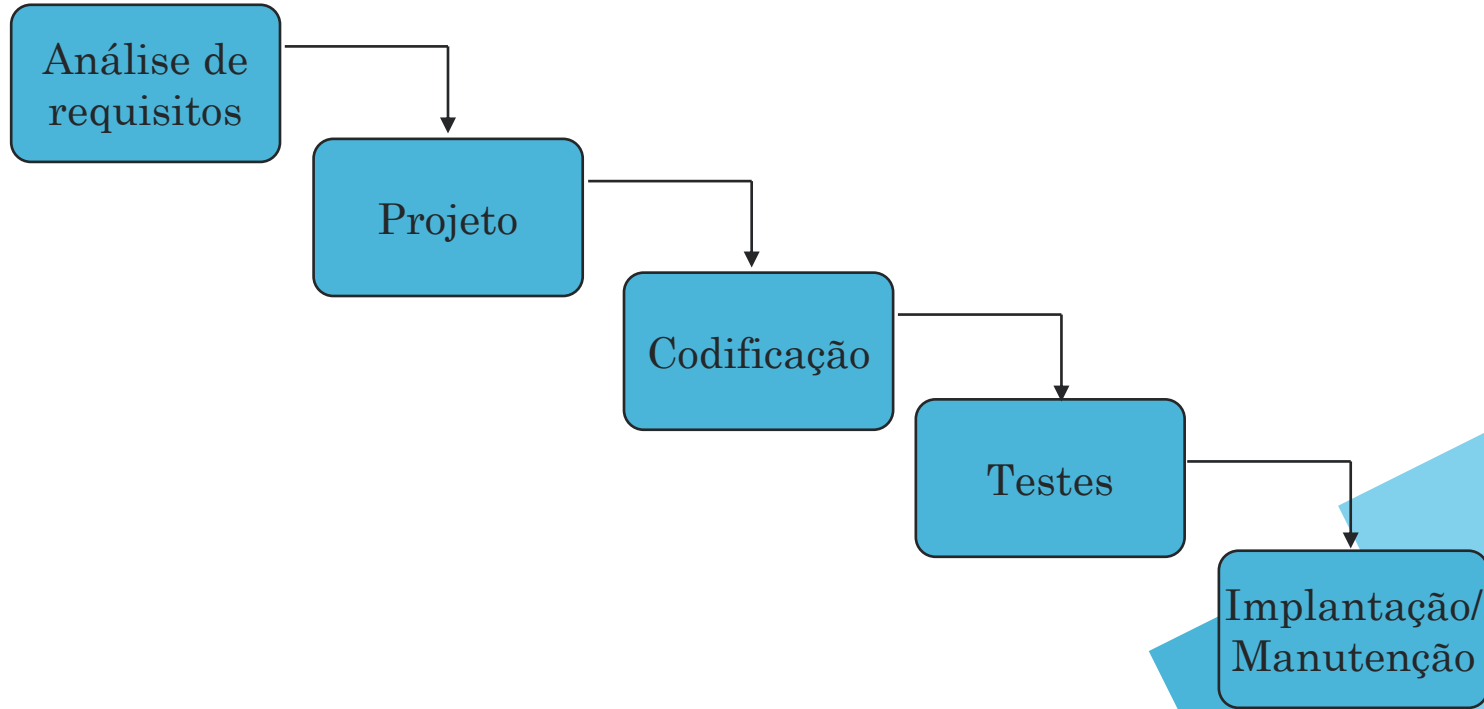


1.

# MODELO CASCATA OU CICLO DE VIDA CLÁSSICO




# Ciclo Clássico ou Cascata






# Ciclo Clássico ou Cascata

- ◎ Método sistemático e sequencial, no qual o resultado de uma fase se constitui na entrada da outra
  - ◎ Cada fase é estruturada como um conjunto de atividades
  - ◎ Meta: tentar a linearidade, para manter o processo previsível e fácil de controlar
  - ◎ O processo deve ser sujeito à disciplina, planejamento e gerenciamento
- 



# Ciclo Clássico ou Cascata

- ◎ A definição dos requisitos deve ser “congelada” logo no início (dificuldade para cliente declarar todas as exigências)
  - ◎ Projetos reais raramente seguem o fluxo sequencial
  - ◎ Exige paciência do cliente (demora em produzir resultados)
  - ◎ Desestimula a prototipação e não promove o reaproveitamento de software
  - ◎ Conduz a “estados de bloqueio” – nos quais alguns membros da equipe precisam esperar outros
- 



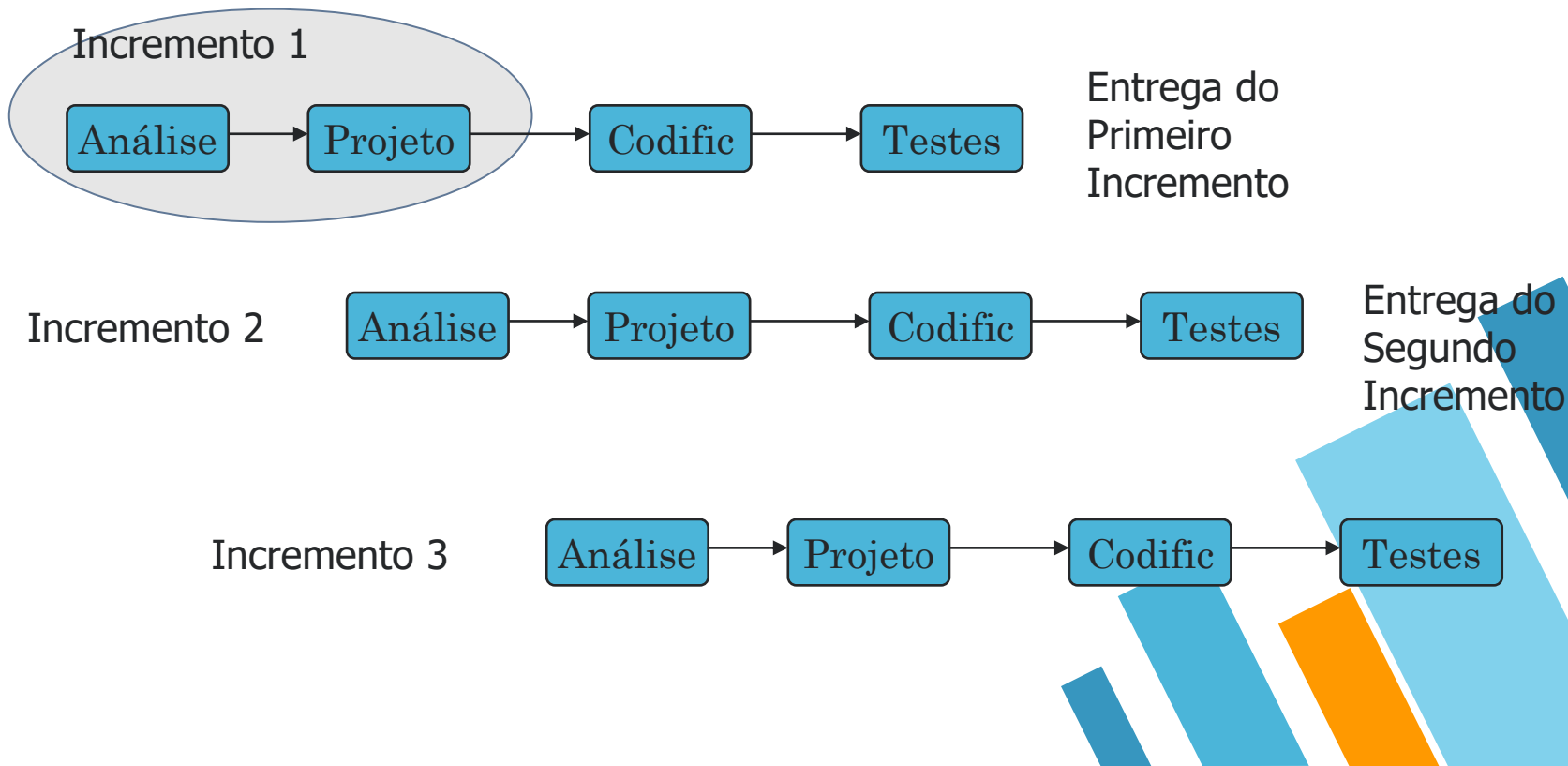


# 2.

## MODELO INCREMENTAL




# Modelo Incremental





# Modelo Incremental

- © Combina elementos do modelo sequencial linear (aplicado repetidamente) com a filosofia interativa da prototipagem
  - © Um modelo incremental aplica várias sequências lineares de um forma racional
  - © Cada sequência linear produz um incremento factível do software
  - © Diferentemente da prototipagem, o modelo incremental objetiva a elaboração de um produto operacional a cada incremento
- 

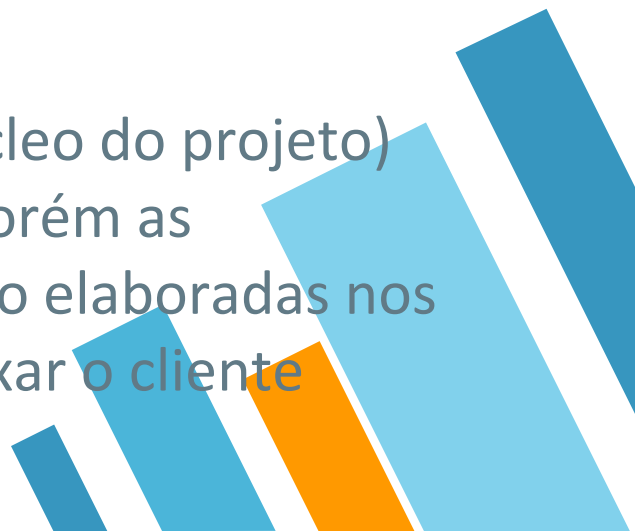


# Modelo Incremental

## Vantagens:

- ◎ Útil quando não há mão-de-obra disponível para uma implementação completa, dentro do prazo comercial de entrega estabelecido para o projeto

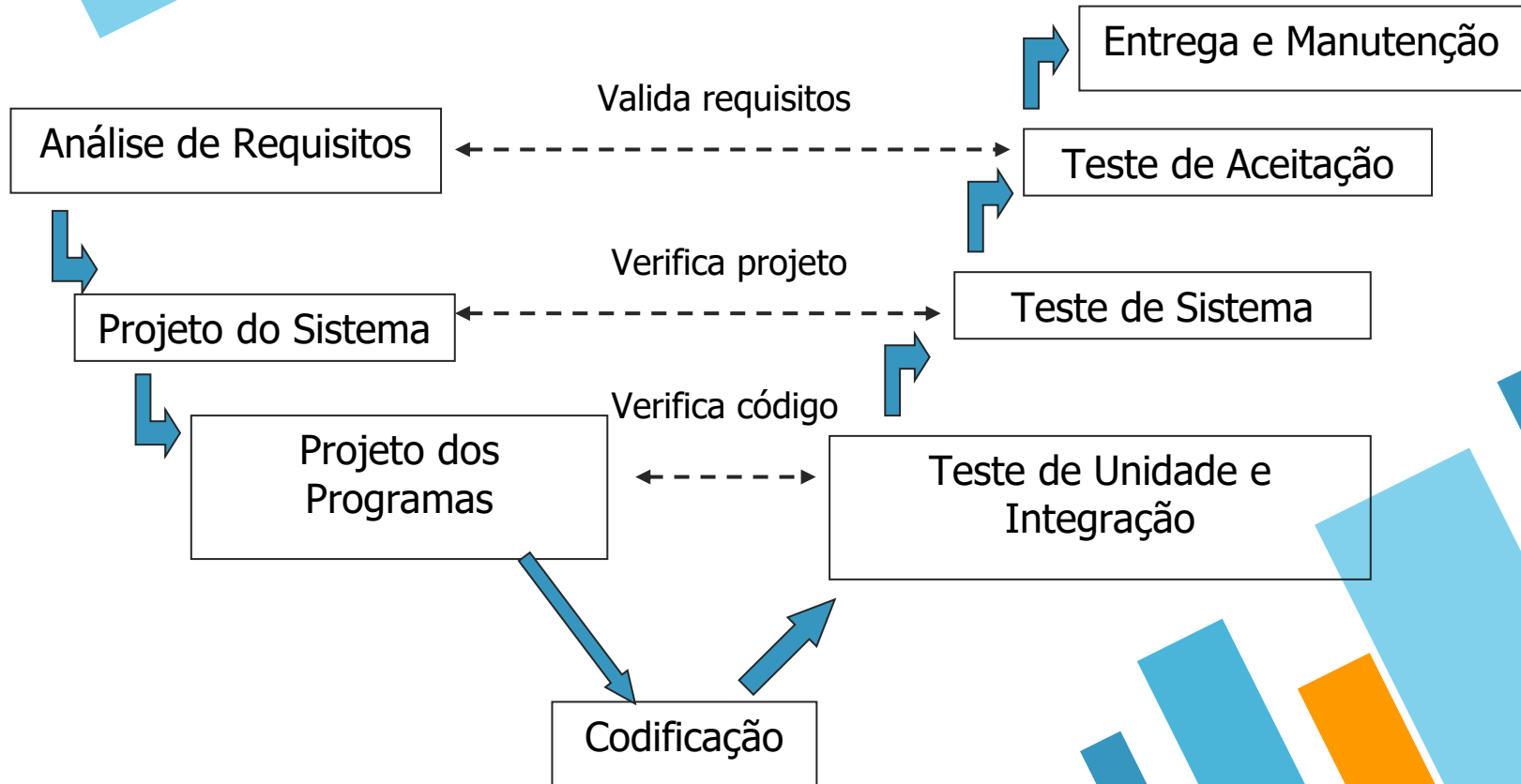
## Problemas:

- ◎ O primeiro incremento (chamado núcleo do projeto) deve atender os requisitos básicos, porém as características suplementares só serão elaboradas nos outros incrementos – o que pode deixar o cliente ansioso ou decepcionado
- 

# 3.


## MODELO V

# Ciclo de Vida Modelo V





# Ciclo de Vida Modelo V

- ◎ Variação do Modelo Cascata
  - ◎ Relaciona ações de garantia da qualidade às ações de análise e projeto
  - ◎ Fornece uma maneira de visualizar como as ações de Verificação & Validação são aplicadas ao longo do desenvolvimento de software
  - ◎ As conexões implicam em retrabalho se problemas são encontrados
- 



4.


# MODELOS EVOLUCIONÁRIOS



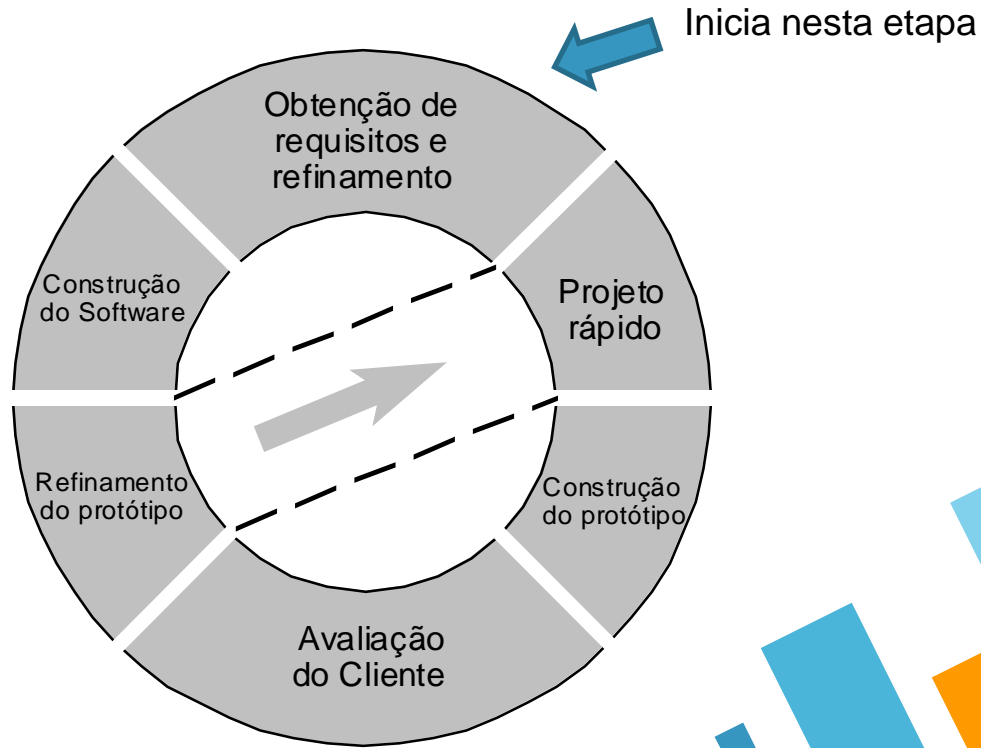




# Modelos Evolucionários


- ◎ São modelos iterativos (composto por ciclos que se repetem)
  - ◎ São caracterizados de forma a permitir aos engenheiros de software desenvolver versões cada vez mais completas do software
  - ◎ Tipos de Modelos Evolucionários:
    - Modelo de Prototipagem
    - Modelo Espiral
- 

# Modelo de Prototipagem



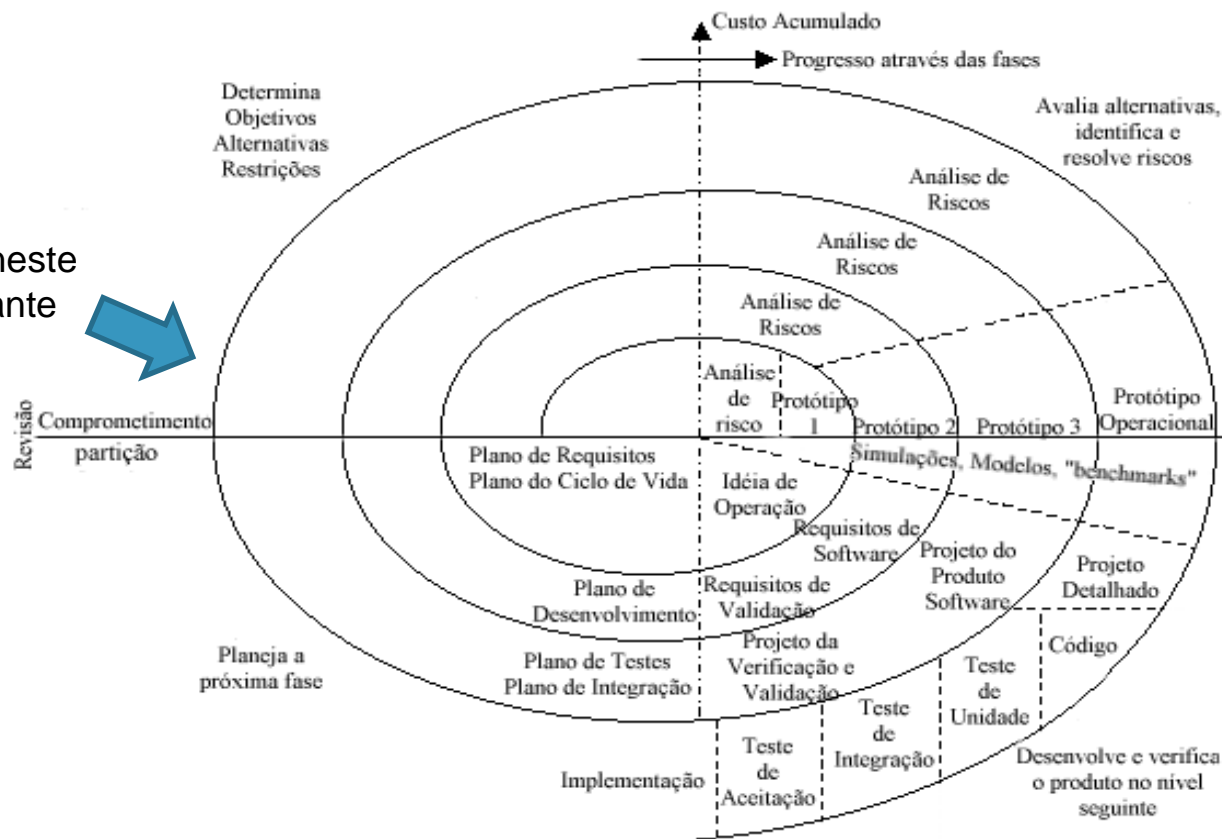


# Modelo de Prototipagem

- ◎ Baseado no desenvolvimento e implementação de um produto inicial, que é submetido aos comentários e críticas do usuário
  - ◎ O produto inicial elaborado deve ser descartado
  - ◎ Vantagens: É um mecanismo eficaz de identificação de requisitos
  - ◎ Problemas: O protótipo muitas vezes é feito sem controles rígidos de qualidade. Cliente tem a falsa noção de que o software já está pronto
- 


# Ciclo de Vida em Espiral

Inicia neste quadrante





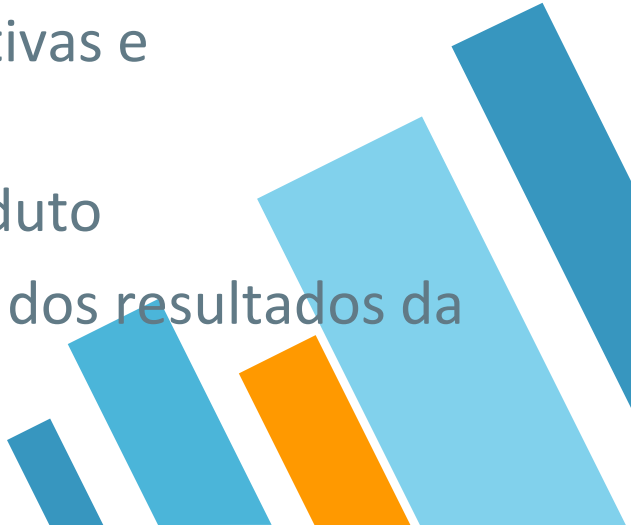
# Ciclo de Vida em Espiral

- © É um modelo evolucionário, que combina a natureza iterativa da prototipagem com os aspectos controlados e sistemáticos do modelo cascata
  - © O software também é produzido em versões incrementais
  - © A versão incremental, durante as primeiras iterações podem ser modelos em papel ou protótipos
- 



# Ciclo de Vida em Espiral

4 atividades que se repetem até o software estar concluído:

- ◎ Planejamento: determinação dos objetivos, alternativas e restrições
  - ◎ Análise dos Riscos: análise de alternativas e identificação/resolução dos riscos
  - ◎ Engenharia: desenvolvimento do produto
  - ◎ Avaliação feita pelo cliente: avaliação dos resultados da engenharia
- 




# Ciclo de Vida em Espiral

Vantagem:

- ◎ Abordagem “evolucionária” - permite entender e reagir aos riscos de cada etapa

Desvantagens:

- ◎ Exige considerável experiência na avaliação dos riscos
  - ◎ Difícil explicar aos clientes/usuários a abordagem evolutiva
- 

# Referências

- **Sommerville, I. (2011). Engenharia de Software, 9 edição. Pearson Prentice Hall.**
- **Pressman, R., & Maxim, B. (2016). Engenharia de Software – Uma abordagem profissional-8ª Edição. McGraw Hill Brasil.**





**OBRIGADA!**

