

# Revisão Processo de Software

Prof.<sup>a</sup> Érica Souza



Cornélio Procópio

# Engenharia de Software

---

- ▶ O que é Engenharia de Software?
- ▶ É a aplicação de uma abordagem **sistemática, disciplinada** e **quantificável** no desenvolvimento, operação e manutenção do software (IEEE Standard 610.12)

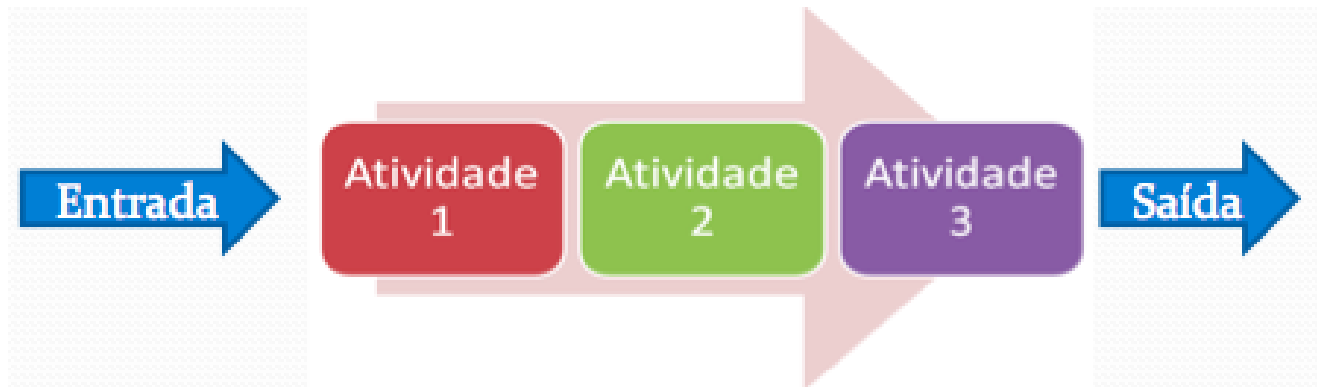


**PROCESSO**

# Processo

---

- ▶ Um processo é uma série de etapas que envolvem **atividades**, **restrições** e **recursos** para alcançar a saída desejada



- ▶ Um processo é um conjunto de procedimentos (**receita**)
- ▶ A Engenharia de Software abrange o uso de **processos**

# Modelo de Processo de Software

---

## ▶ **Processo de software**

- ▶ É um conjunto de **atividades ordenadas**, cuja meta é o desenvolvimento ou a evolução do software de qualidade



## ▶ **Modelo de Processo de Software**

- ▶ É uma **representação simplificada** de um processo de software, apresentada a partir de uma perspectiva específica
- ▶ Definem um conjunto **distinto** de atividades, ações, tarefas, marcos e produtos de trabalho



# Revisão

---

- ▶ **Modelos Tradicionais**

- ▶ Modelos Sequenciais

- ▶ Modelos Incrementais

- ▶ Modelo Evolutivos de Processo de Software

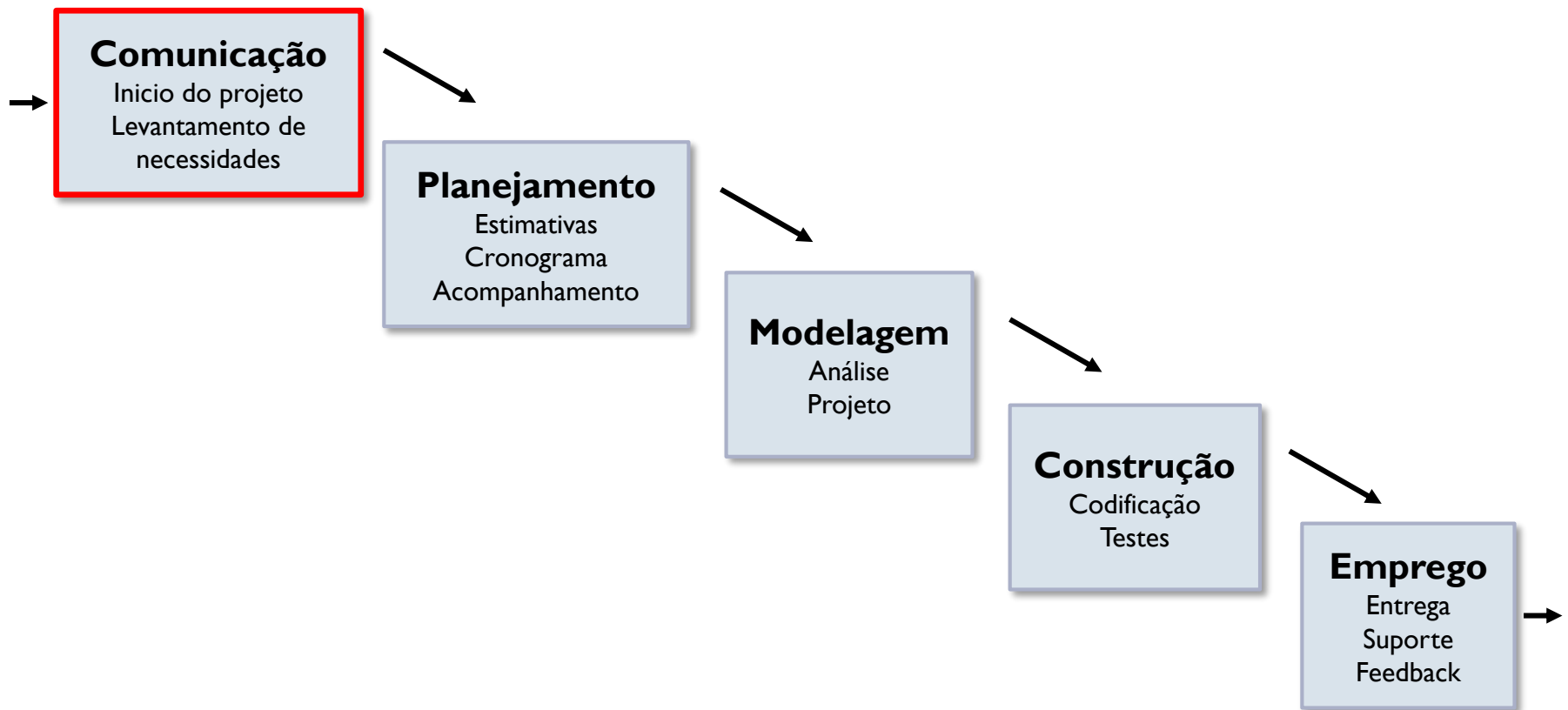
- ▶ **Métodos Ágeis**



# Modelos de Processos Sequências

---

- **Modelo Cascata:** Também chamado de modelo de **ciclo de vida clássico** – 1970. Organiza as atividades de forma sequencial.



# O Modelo de Processo Incremental

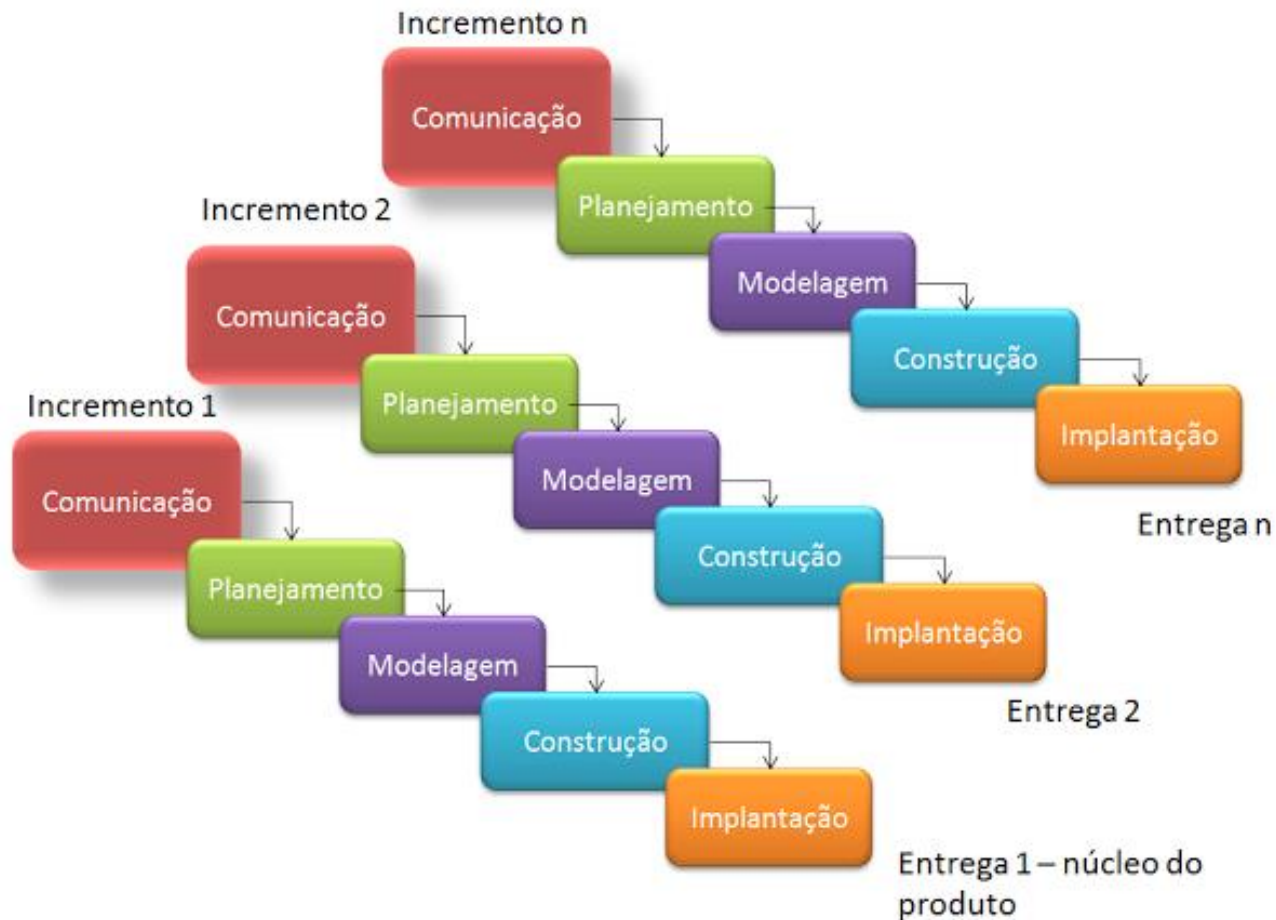
---

- ▶ A **cada ciclo**, uma versão operacional do sistema é produzida e entregue para uso ou avaliação detalhada do cliente
- ▶ O primeiro incremento tipicamente contém funcionalidades centrais, tratando dos requisitos básicos. Outras características são tratadas em ciclos subsequentes
- ▶ O modelo incremental combina elementos dos fluxos de processos lineares



# O Modelo Incremental

---



Cada sequência linear gera “incrementais” do software

---

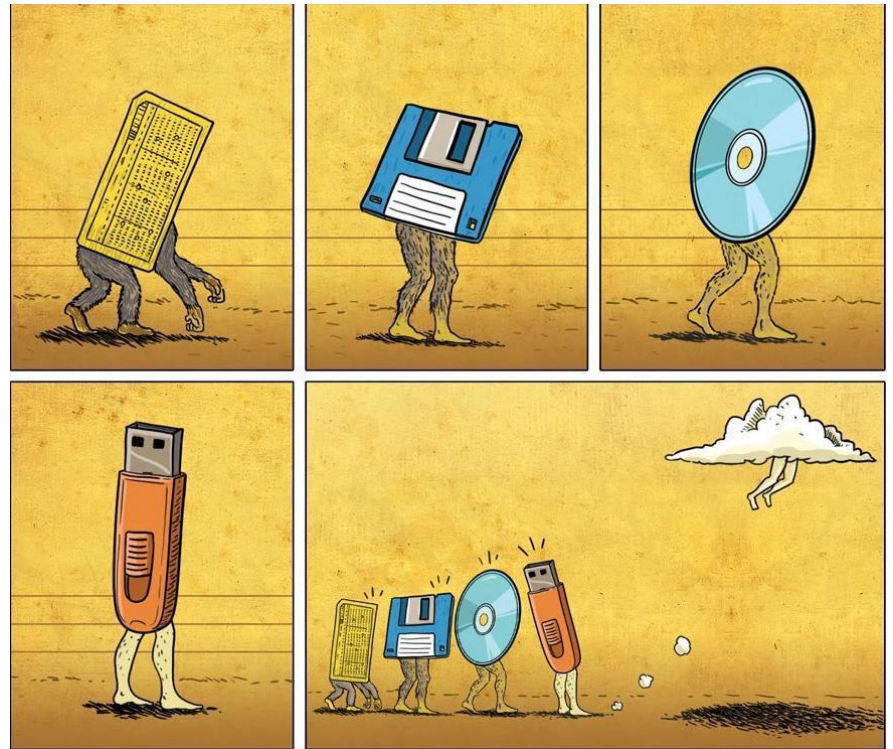




# Modelos Evolutivos

---

- **Software Evolui! → mudanças frequentes**



# Modelos Evolutivos

---

O desenvolvimento evolucionário baseia-se na ideia de desenvolvimento de uma implementação inicial, expondo o resultado aos comentários dos usuários e refinando esse resultado por meio de várias versões até que seja desenvolvido um sistema adequado



# Modelos Evolutivos

---

- ▶ Importante ter como opção modelos de ciclo de vida que lidem com incertezas e acomodem melhor as **contínuas mudanças**
- ▶ Os modelos de processo evolucionário produzem uma versão cada vez mais completa do software a cada **iteração**
- ▶ São Iterativos



# Incremental x Iterativo

---

## Incremental



## Iterative



# Modelo Espiral

---

- ▶ É um dos modelos evolutivos mais difundidos
- ▶ Acopla a natureza **iterativa** da prototipação
- ▶ O processo é representado como um espiral em uma série de regiões (áreas)
- ▶ Apresenta uma abordagem cíclica voltada para ampliar, de forma incremental, o grau de definição e a implementação de um sistema, enquanto diminui o grau de risco do mesmo



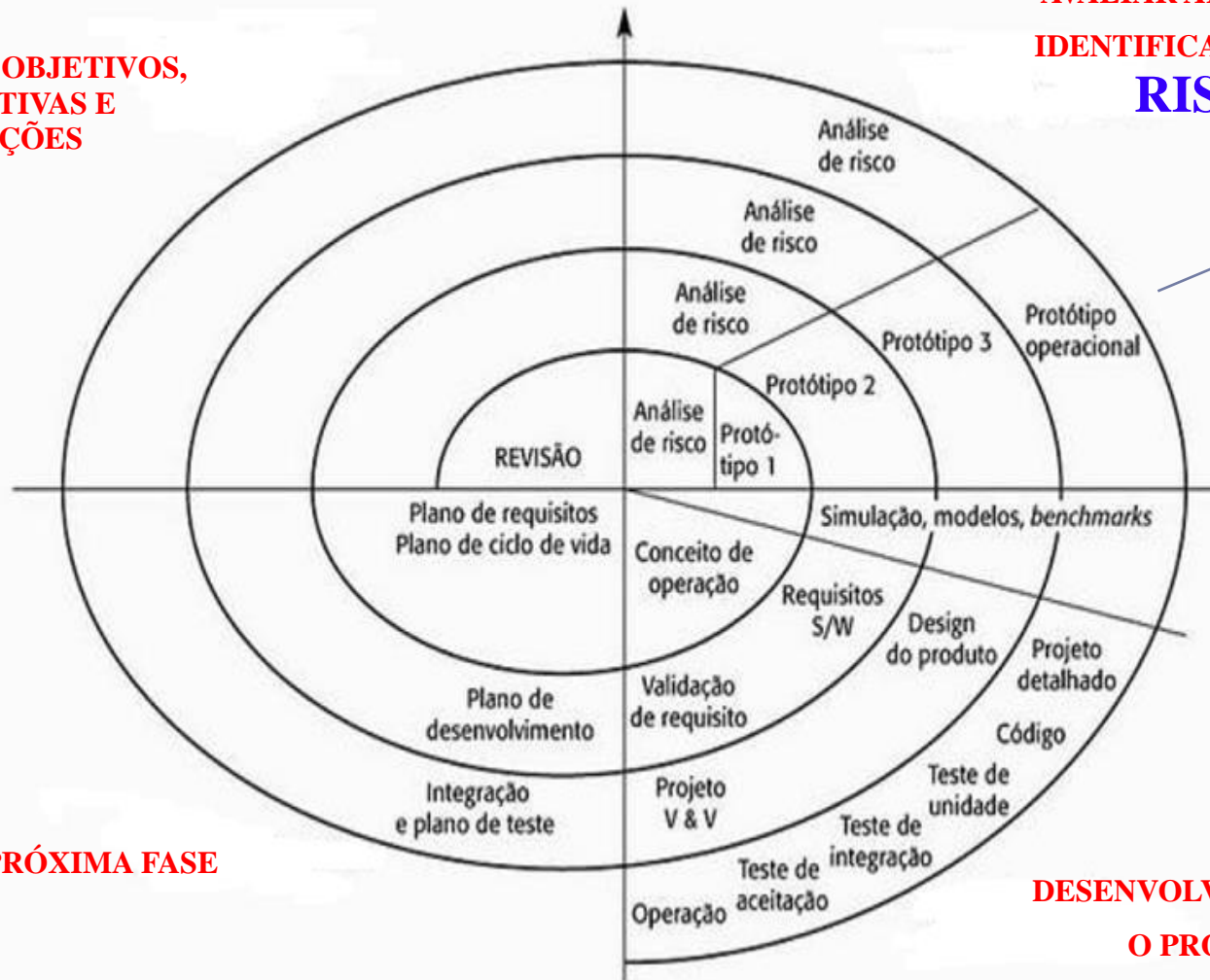
# Modelo Espiral

## ► Representação (4 regiões) [Sommerville]

**DETERMINAR OBJETIVOS,  
ALTERNATIVAS E  
RESTRICÇÕES**

**AVALIAR ALTERNATIVAS  
IDENTIFICAR, RESOLVER  
RISCOS**

Mecanismo para  
redução de riscos



**PLANEJAR PRÓXIMA FASE**

**DESENVOLVER, VERIFICAR  
O PRODUTO NO  
PRÓXIMO NÍVEL**

# Desenvolvimento ágil

---

- ▶ Um dos principais desafios no desenvolvimento de software é lidar com **mudanças**
- ▶ Uma alternativa para esses casos seria a **agilidade – respostas rápidas**
- ▶ A engenharia de software ágil combina **filosofia com um conjunto de princípios** de desenvolvimento





# O que é agilidade?

---



" Yes, you are a developer and yes, you're agile but that doesn't necessarily make you an agile developer. "



# Desenvolvimento ágil

---

Uma equipe ágil é aquela que consegue **responder rapidamente a mudanças**, dá valor às características e **habilidades** de cada membro e reconhece que a **colaboração** é a chave para o sucesso do projeto (PRESSMAN, 2011)

Enfatiza a entrega rápida do software e diminui a importância dos artefatos intermediários (nem sempre um bom negócio) (PRESSMAN, 2011)

Assume o cliente como parte da equipe

Incremental e iterativo



# Desenvolvimento ágil

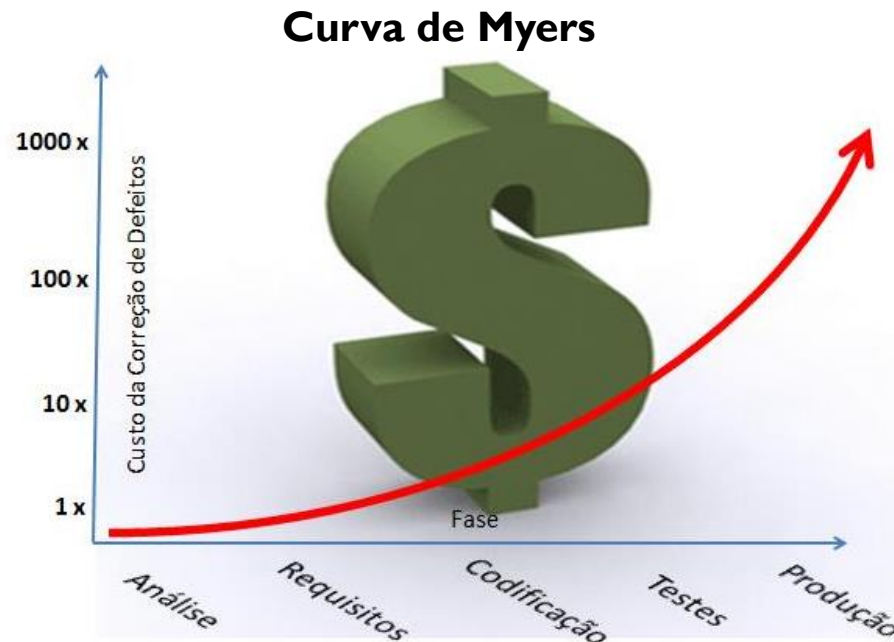
---

- ▶ Os princípios de agilidade podem ser aplicados a quaisquer processos de software
- ▶ Dentre os métodos ágeis, destacam-se:
  - ▶ **eXtreme Programming (XP)**
  - ▶ **Scrum**



# Cenário atual de desenvolvimento

- ▶ 85% dos erros são causados por defeitos inseridos durante a análise de **requisitos** e projeto do sistema
- ▶ Os erros mais caros são aqueles cometidos no processo de desenvolvimento de requisitos e descobertos pelo usuário



# Cenário atual de desenvolvimento

---

## ▶ Problemas comuns

- ▶ Problemas de escopo
  - ▶ O **limite do sistema** é mal definido
- ▶ Problemas de entendimento (clientes e usuários)
  - ▶ Não estão completamente certos do que é **necessário**
  - ▶ Dificuldade de **comunicar** as necessidades
- ▶ Problemas de volatilidade
  - ▶ Os requisitos **mudam**

# Importância dos Requisitos

---

- ▶ Estabelece uma **base de concordância** entre cliente e o fornecedor sobre o que o software fará
- ▶ Fornece uma referência para a **validação** do produto final
- ▶ Uma especificação de requisitos de alta qualidade é **pré-requisito** para um software de alta qualidade
- ▶ Reduz o custo de desenvolvimento

# Sugestão de Leitura

---

- ▶ Projeto: ***N*aming the *P*ain in *R*equirements *E*ngineering (NaPiRE)**
  - ▶ <http://www.mendezfe.org/projects/napiRE/>
- ▶ Artigos publicados no WER - *Workshop on Requirements Engineering*
  - ▶ <http://wer.inf.puc-rio.br/WERpapers/>