



# Revisão para a Prova 1

Eduardo Figueiredo

<http://www.dcc.ufmg.br/~figueiredo>

[dcc603@dcc.ufmg.br](mailto:dcc603@dcc.ufmg.br)

**03 Agosto 2020**

# [ Aula 1 - Apresentação ]

- Apresentação da disciplina
- Explicação sobre o trabalho prático

# [ Aula 2 - Introdução ]

---

- Engenharia de Software
- Estrutura da Engenharia de Software
- Atividades de Desenvolvimento de Software

# [ O que é software? ]

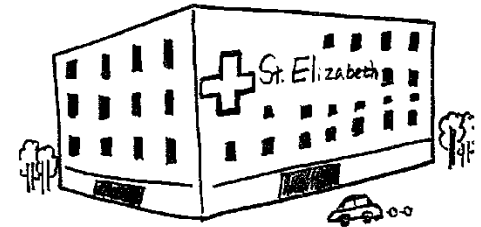
- Programa de computador + Documentação
- Classificação fundamental
  - Produtos genéricos (ex. MS Office)
  - Produtos encomendados (ex. Software de Controle da Loja do Zé)



# Desafios de Produzir Software

## Sistemas Críticos

- Equipamentos médicos
  - Extremamente críticos
  - Lidam com vidas
- Caixas eletrônicos
  - Prejuízos financeiros

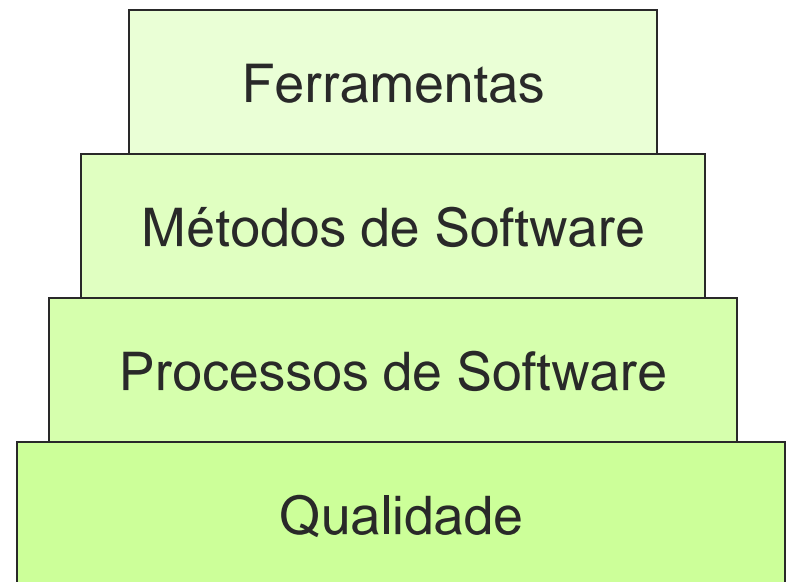


# Engenharia de Software

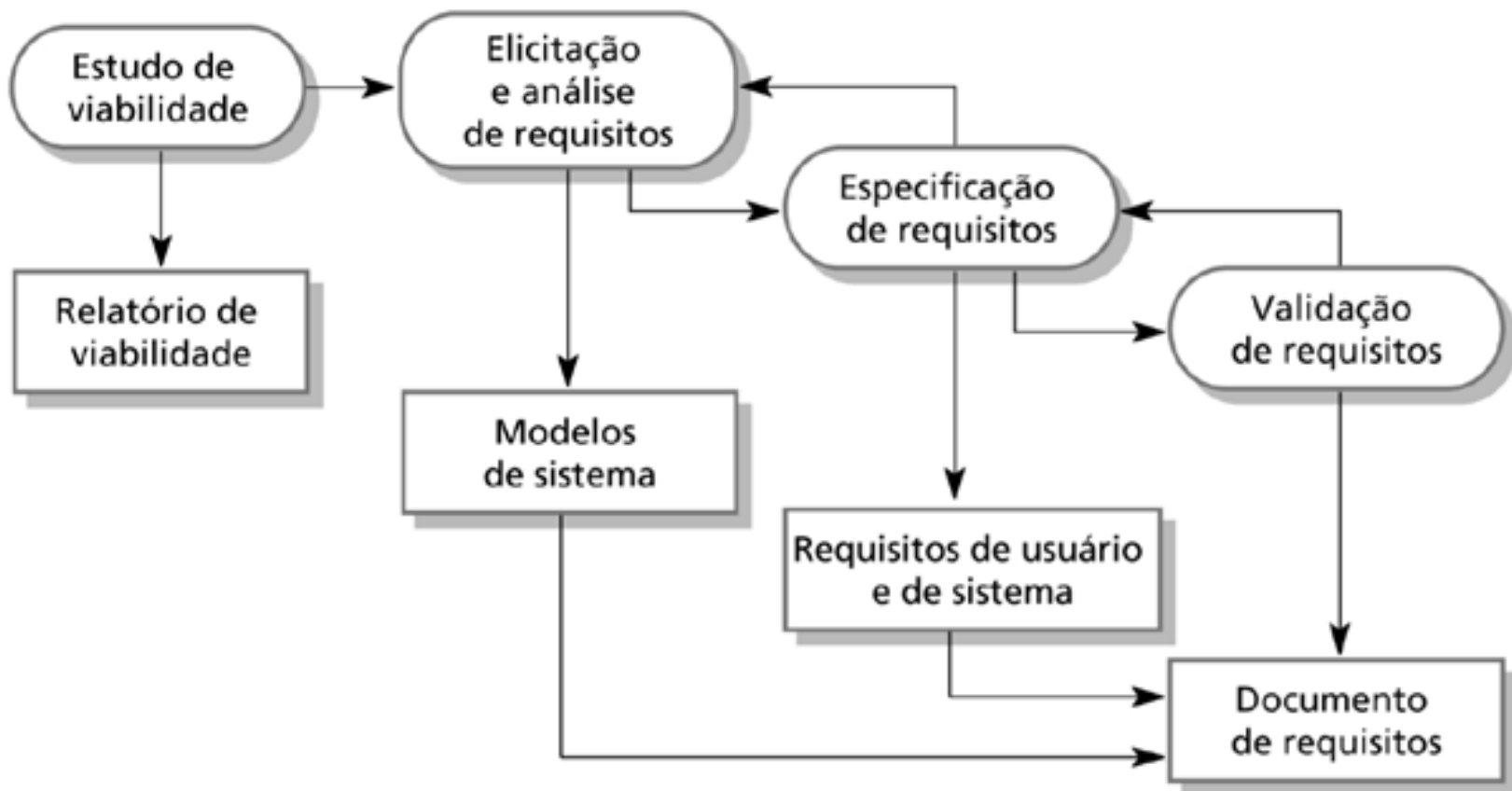
- Pode ser organizada em camadas com foco em qualidade

- A engenharia de software inclui

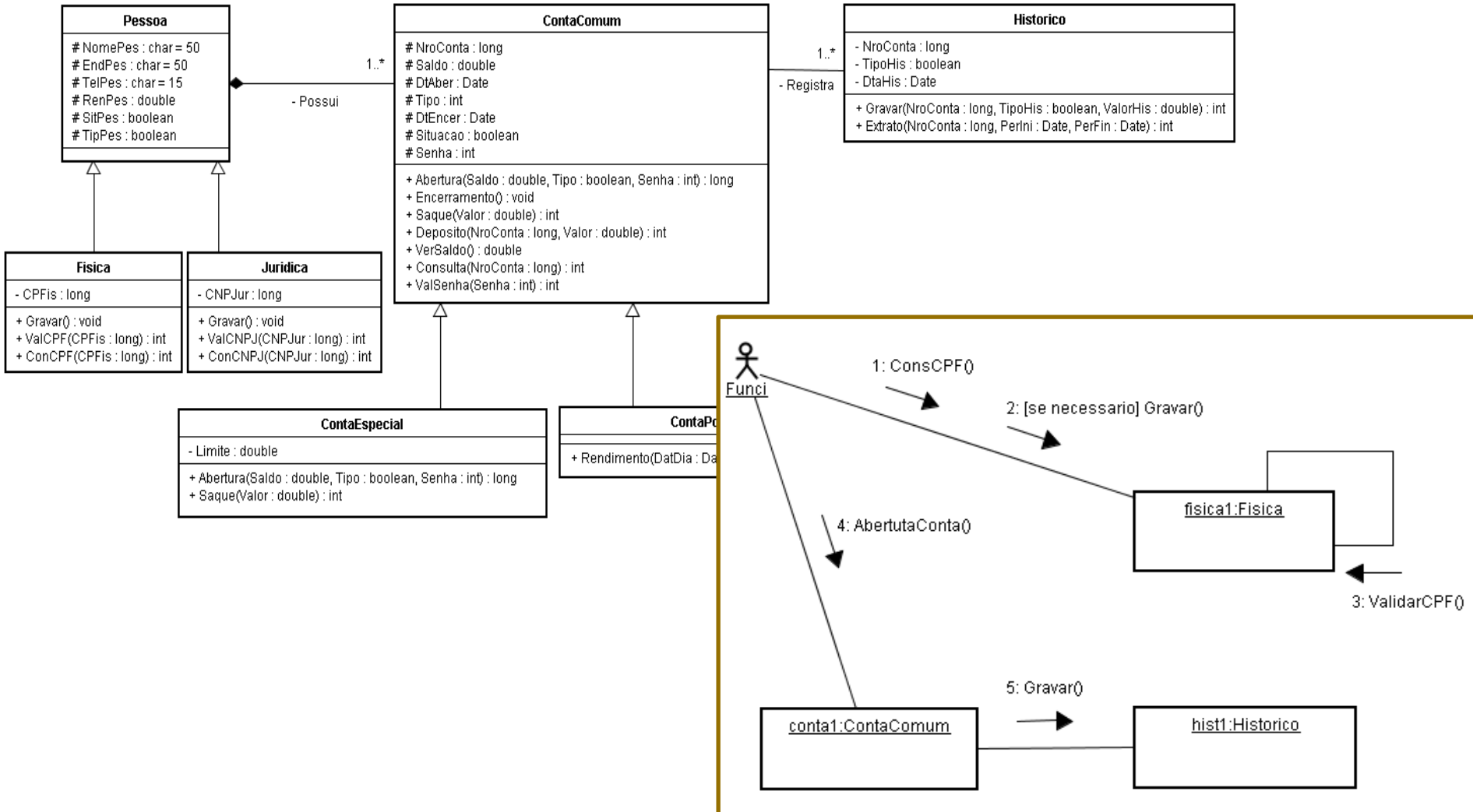
- Processo
- Métodos
- Ferramentas



# Engenharia de Requisitos



# Modelagem de Software



# [ Implementação ]

- A implementação segue as definições da fase anterior
  - Transcreve as decisões de projeto arquitetural e detalhado para uma linguagem de programação
- Abordado em outras disciplinas
  - PDS 1 e 2, POO, Programação de Computadores, Programação Modular, etc.



# Verificação e Validação



- Tem por objetivo garantir que o sistema satisfaça os requisitos
- Verificação consiste na realização de alguns **testes** para encontrar erros
- A inexistência de erros não representa a *adequação operacional* do sistema
  - Deve ser feita a **validação** com o cliente

# [ Manutenção e Evolução ]

- O custo de manutenção é geralmente muito maior que o custo de desenvolvimento
- Cada vez menos sistemas são desenvolvidos “do zero”
  - Sistemas são desenvolvidos/adaptados a partir de outros sistemas
- Faz sentido considerar desenvolvimento e manutenção como atividades contínuas

# [ Aula 2 - Bibliografia ]

- Ian Sommerville. Engenharia de Software, 9ª Edição. Pearson Education, 2011.
  - Capítulo 1
- Roger Pressman. Engenharia de Software: Uma Abordagem Profissional, 7ª Edição. McGraw-Hill, 2011.
  - Capítulo 1

# Aula 3 - Processos de Software

- Conjunto de atividades que leva ao desenvolvimento do produto software
- Um processo define
  - Quem faz, o que faz e quando fazer
  - Nem sempre diz como fazer
- Não existe um processo ideal
  - Organizações desenvolvem seus próprios processos

# Modelo Cascata

- Atividades sequenciais
- Uma fase deve ser terminada para a outra começar
  - Raramente ocorre na prática

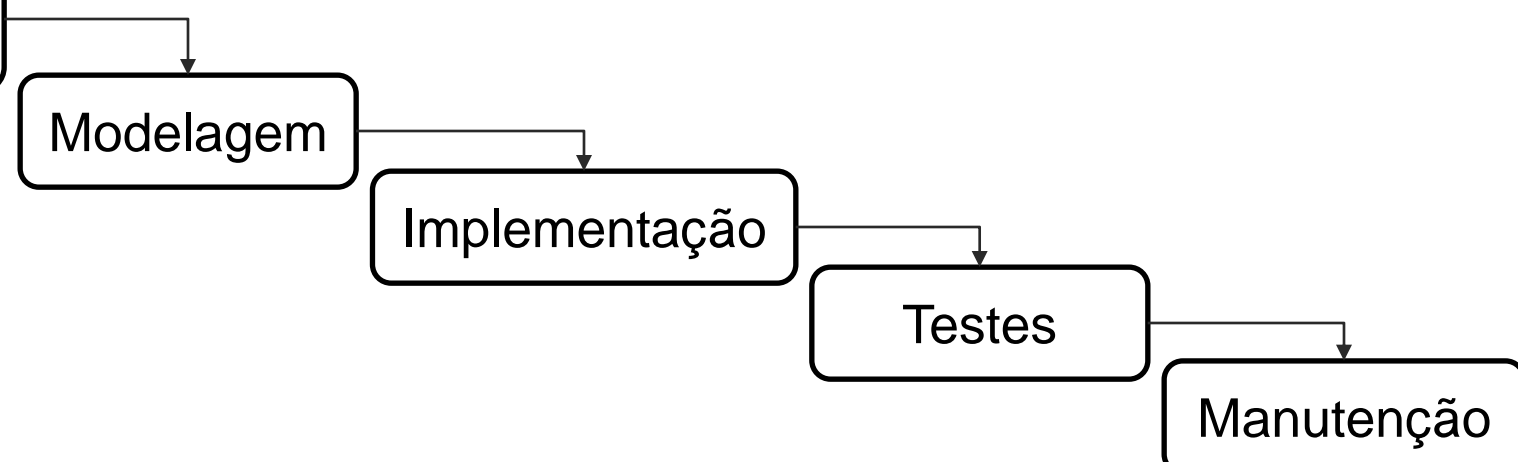
Definição de  
Requisitos

Modelagem

Implementação

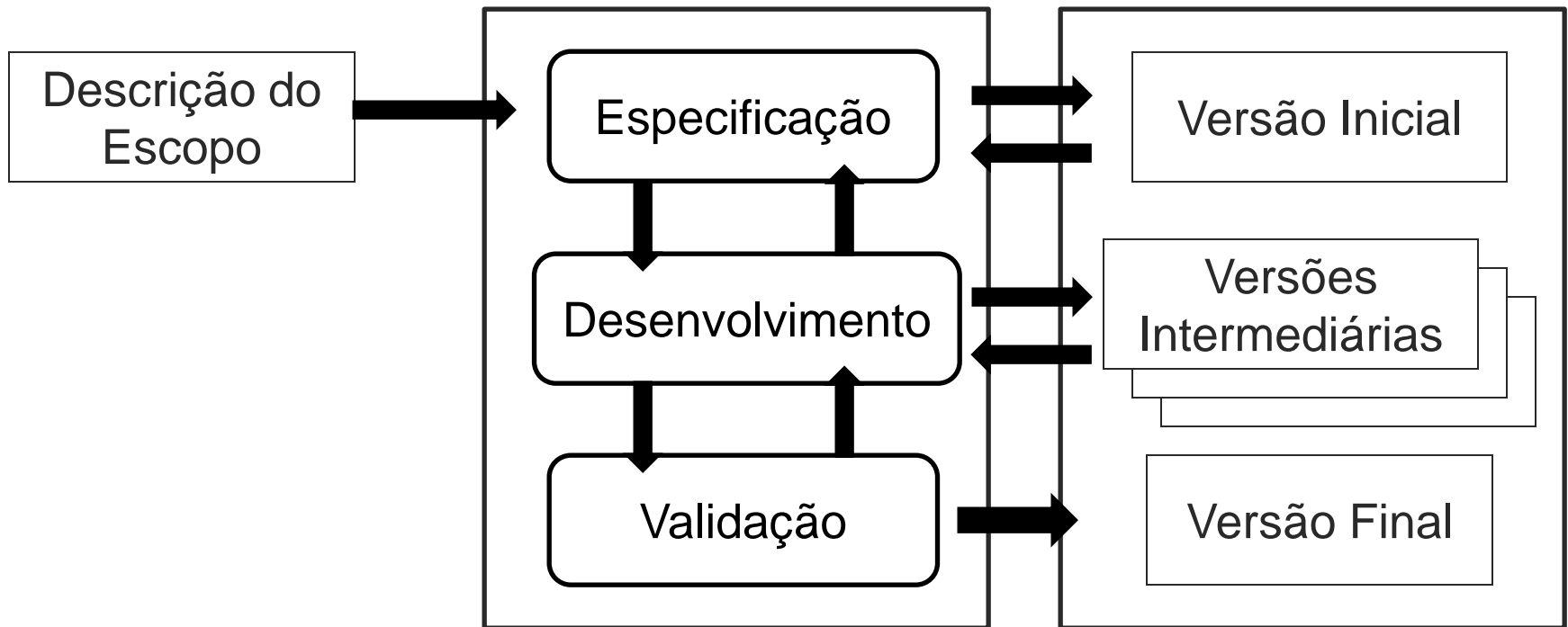
Testes

Manutenção



# Desenvolvimento Incremental

- Atividades são intercaladas
- Objetivo: dar feedback rápido ao cliente



# Baseado em Reutilização

- Baseia-se na existência de um número significativo de componentes reusáveis
- O processo se concentra na integração dos componentes reusáveis

Especificação de Requisitos

Análise de Componentes

Desenvolvimento e Integração

Validação do Sistema

Alteração nos Requisitos

Projeto do Sistema com Reuso

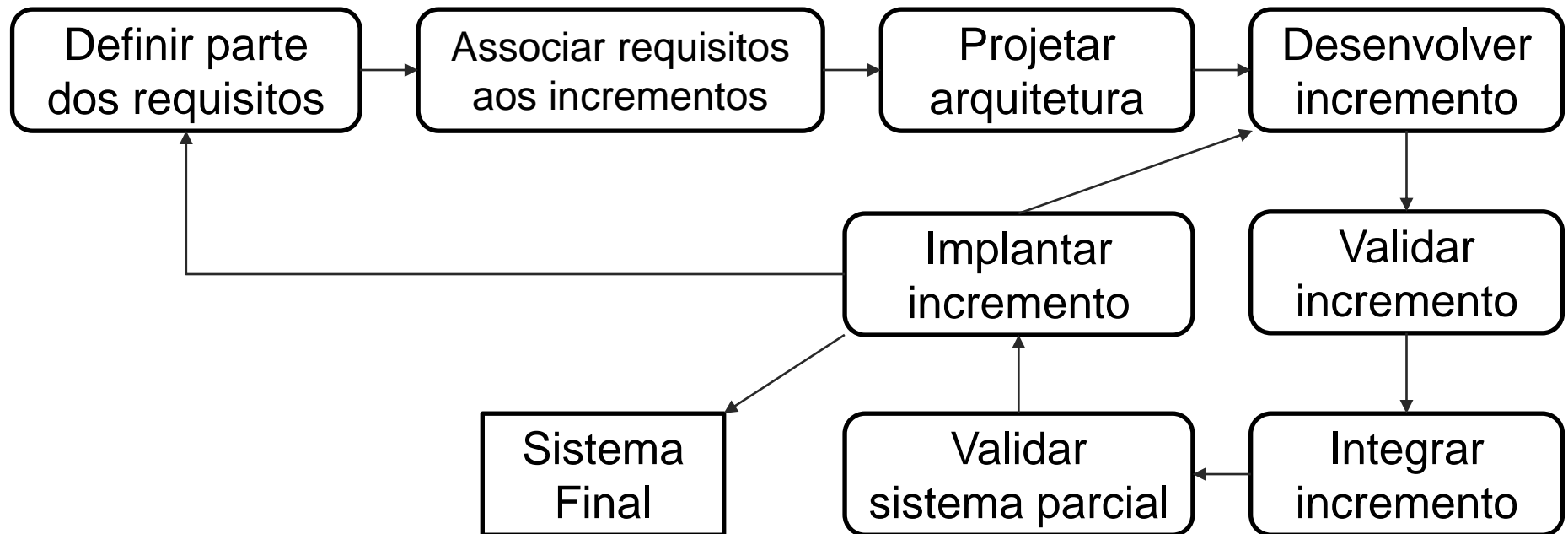


# [ Prototipação ]

- É geralmente usada junto com outro modelo de processo
- Planeja e modela rapidamente um protótipo
  - Mais comum na definição de interfaces com os usuários (telas)
- Começa com os requisitos menos compreendidos
  - Objetivo: entender os requisitos

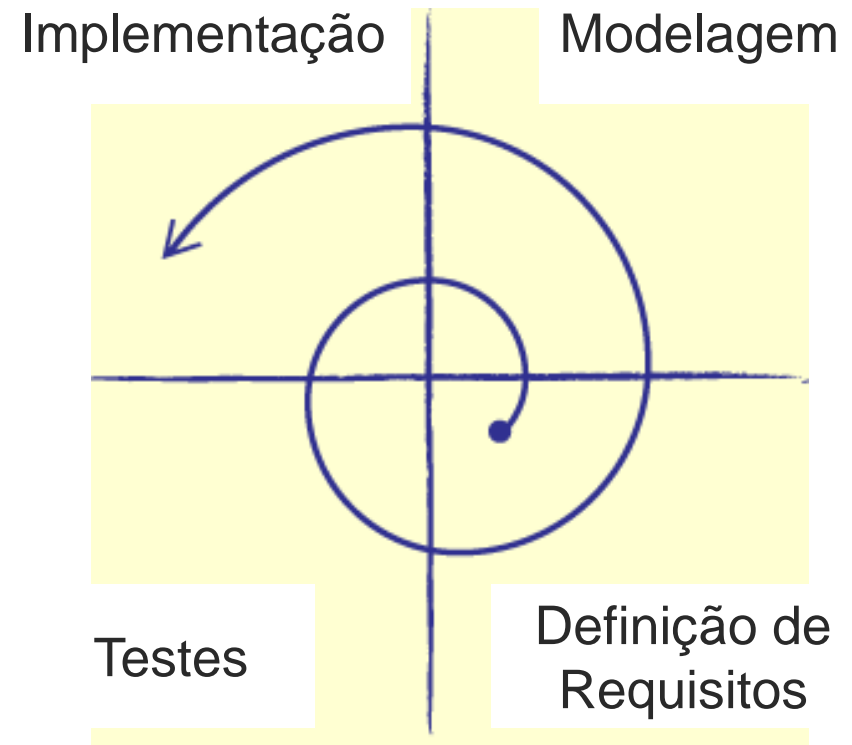
# [ Entrega Incremental ]

- Combina elementos do modelo cascata aplicados de maneira iterativa



# Modelo Espiral

- Combina elementos dos modelos incrementais e de Prototipagem
  - É sequência adotada do Modelo Cascata
- Software é desenvolvido em versões
  - Prototipagem nas primeiras versões
  - Incremental nas últimas versões



# [ Aula 3 - Bibliografia ]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
  - Capítulo 2

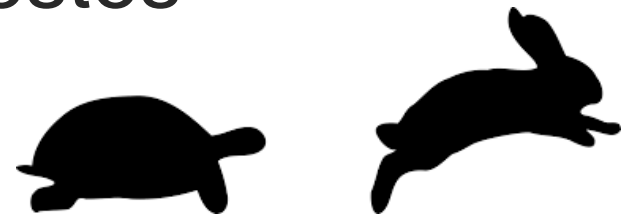
# [ Aula 4 - Exercício ]

---

- Exercício sobre Processos de Software usando o Trello

# Aula 5 - Métodos Ágeis

- Processos tradicionais são baseados em especificação detalhada dos requisitos, projeto e testes



- Métodos ágeis têm por objetivo criar software útil rapidamente
  - Não se preocupam com a documentação completa em todas as fases

# [ Manifesto Ágil ]

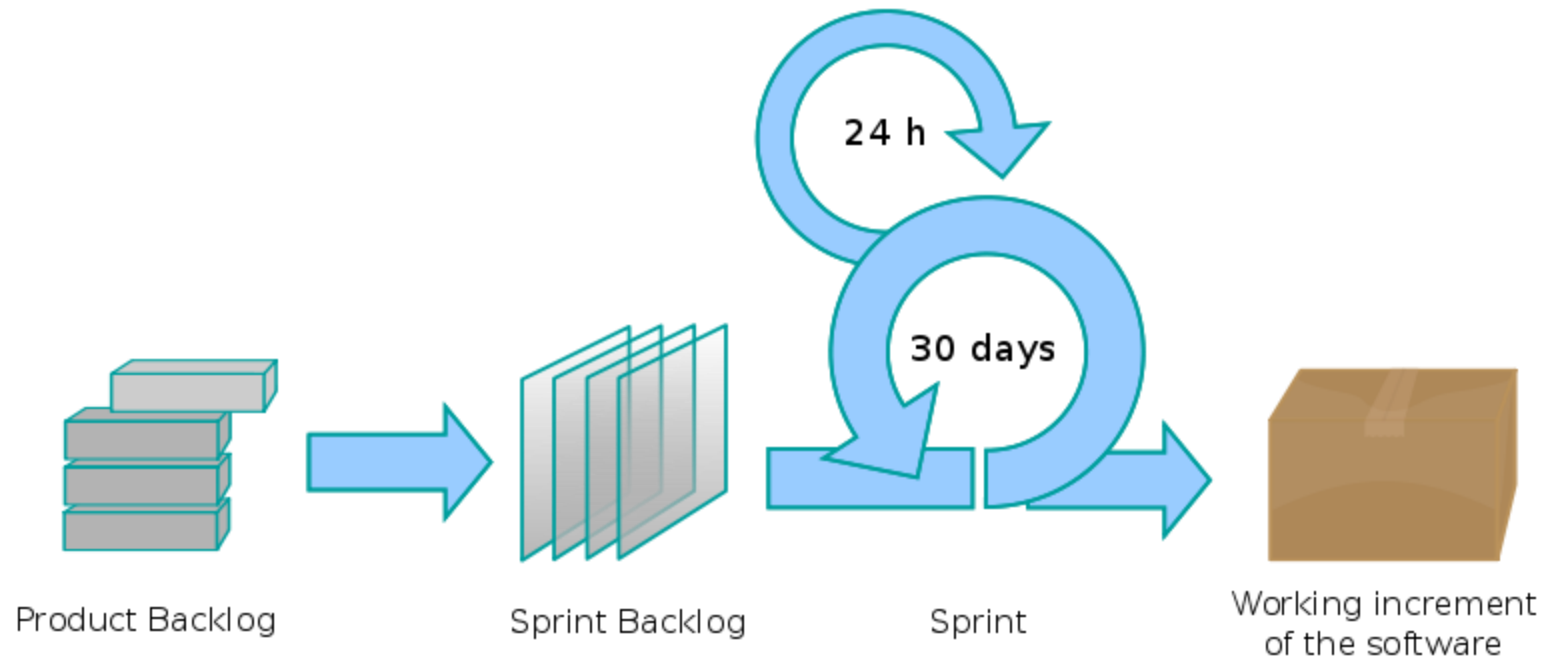
- Passamos a valorizar:
  - **Indivíduos e interações** mais que processos e ferramentas
  - **Software em funcionamento** mais que documentação abrangente
  - **Colaboração com o cliente** mais que negociação de contratos
  - **Responder a mudanças** mais que seguir um plano

# [ Programação Extrema (XP) ]

- Proposta a partir de boas práticas de desenvolvimento incremental
- Propõe o envolvimento do cliente ao extremo
  - O cliente (ou seu representante) deve estar disponível durante todo o desenvolvimento
- Programadores trabalham em pares

# Scrum

- Planejamento (*pre-game phase*)
- Desenvolvimento (*game phase*)
- Encerramento (*post-game phase*)



# [ Aula 5 - Bibliografia ]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
  - Capítulo 3
- A. Koscianski e M. Soares. **Qualidade de Software**, 2ª Edição. Novatec, 2006.
  - Seção 10.3 Metodologias Ágeis

# [ Aula 7 - Requisitos (5/8) ]

- Classificação de Requisitos
- Engenharia de Requisitos
- Técnicas de Elicitação de Requisitos

# [ Engenharia de Requisitos ]

- Requisitos
  - Descrições dos serviços fornecidos pelo sistema e das restrições sobre estes serviços
- Engenharia de Requisitos
  - Processo de descobrir, analisar, documentar e verificar requisitos

# [ Elicitação de Requisitos ]

- Técnicas para levantamento de requisitos
  - Descuberta de Requisitos (Pontos de vista)
  - Entrevistas
  - Cenários
  - Casos de Uso
  - Etnografia

# [ Aula 7 - Bibliografia ]

- Ian Sommerville. **Engenharia de Software**, 9ª Edição. Pearson Education, 2011.
  - Capítulo 4