

# ENGENHARIA DE SOFTWARE I

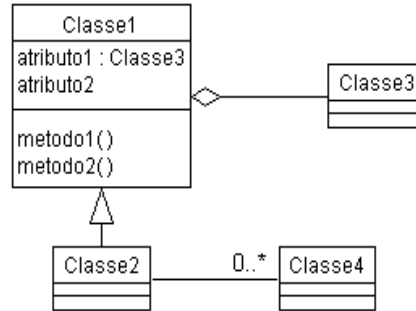
# Leitura complementar

- Livro: Engenharia de Software: uma abordagem profissional - 7ª edição (2011)  
Autor: Pressman
- 2.3.1. O modelo cascata
- 2.3.2. Modelos de processo incremental

# Fases de um projeto de software



**Elicitação/  
Especificação  
de requisitos**



**Projeto**



**Codificação**



**Testes**



**Implantação**

**Manutenção**



**GERÊNCIA DE PROJETOS**



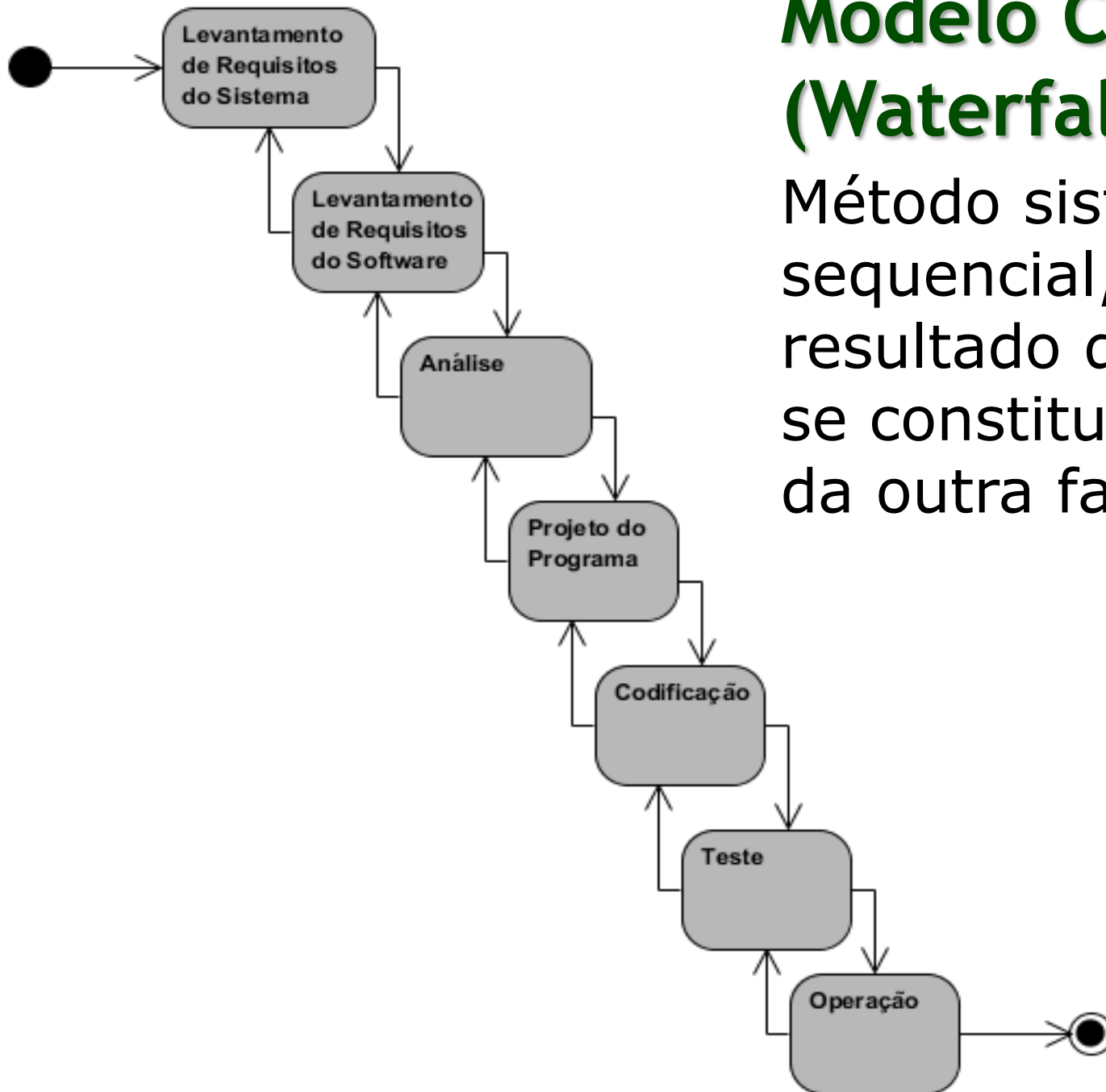
# Modelos de processo de software

# Principais modelos

- Modelo Cascata
- Modelo Incremental
- Modelo Espiral
- existem outros, mas estes são os mais utilizados

# Modelo Cascata (Waterfall)

Método sistemático e sequencial, em que o resultado de uma fase se constitui na entrada da outra fase.



# Modelo cascata (Waterfall)

- Modelo mais antigo e o mais amplamente usado na engenharia de software, modelado em função do ciclo da engenharia convencional.
- Modelo dirigido a planos. Fases de especificação e desenvolvimento separadas e distintas.

# Modelo Cascata

## Problemas do modelo Cascata:

- dificuldade de acomodação de mudanças depois que o processo já foi iniciado
  - Por isso esse modelo só é **apropriado quando os requisitos são bem entendidos** e as mudanças durante o processo de projeto serão limitadas
  - Poucos sistemas de negócio possuem requisitos estáveis.



# Modelo Cascata

## Problemas do modelo Cascata:

- Requisitos devem ser estabelecidos de maneira completa correta e clara no início de um projeto
- Difícil avaliar o progresso verdadeiro do projeto durante as primeiras fases
- Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento
- Ao final do projeto, é necessário um grande esforço de integração e testes
- Gera grande quantidade de documentação

# Modelo Cascata

- Este modelo tem sido muito criticado pelos defensores dos métodos ágeis

# Modelo incremental e iterativo

Delivery 1



Delivery 2



Delivery 3



Incremental plan



Iterative plan

# Modelo incremental e iterativo

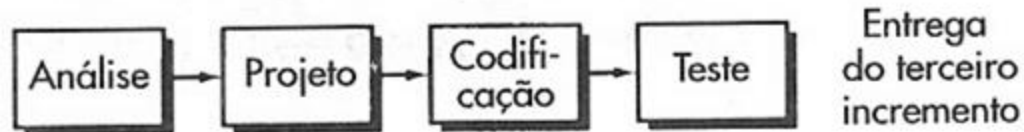
Incremento 1



Incremento 2



Incremento 3



Incremento 4



# Modelo Incremental

- Ao invés de entregar o sistema em uma única entrega, o desenvolvimento e a entrega são distribuídos em incrementos, nos quais cada incremento entrega parte da funcionalidade necessária.
- O processo se repete até que um produto completo seja produzido.
- Os requisitos do usuário são priorizados e os requisitos de mais alta prioridade são incluídos nos primeiros incrementos
- Assim que o desenvolvimento de um incremento é iniciado os requisitos são congelados, mas os requisitos dos incrementos posteriores podem continuar a evoluir
- Necessidade de entrega de um produto funcional em pouco tempo
- A cada incremento é produzida uma versão operacional do software.
- Abordagem normalmente usada em métodos ágeis

# **Benefícios do desenvolvimento incremental**

- **O custo para acomodar mudanças nos requisitos do cliente é reduzido:**
  - A quantidade de análise e documentação que precisa ser feita é bem menor do que a necessária no modelo cascata.
- **É mais fácil obter feedback do cliente sobre o trabalho de desenvolvimento que tem sido feito:**
  - Os clientes podem comentar demonstrações do software e ver o quanto foi implementado.
- **Possibilidade de mais rapidez na entrega e implantação de software útil para o cliente:**
  - Os clientes podem usar e obter ganhos do software mais cedo do que é possível no processo cascata.
- **Menor risco de falha geral do projeto**

# Problemas do desenvolvimento incremental

- **A maioria dos sistemas requer um conjunto de funções básicas que são usadas por diferentes partes do sistema:**
  - Como os requisitos não são definidos em detalhes até que um incremento seja implementado, pode ser difícil identificar funções comuns que são necessárias a todos os incrementos.
- **A estrutura do sistema tende a degradar conforme novos incrementos são adicionados:**
  - A menos que tempo e dinheiro sejam gastos na reconstrução para melhorar o software, as mudanças regulares tendem a corromper a estrutura do sistema. A incorporação posterior de mudanças no software se torna progressivamente mais difícil e cara.
- **O gerenciamento de custo, cronograma e configuração é mais complexo**
- **Se os requisitos são instáveis ou não descritos de forma completa quanto se esperava, alguns incrementos podem precisar ser retirados de uso ou causar retrabalhados**