

*O Software:*  
um produto e um processo



Aula 2

# Sumário

## # O produto

- O software como produto
- Evolução
- Características
- Tipos
- Aplicações
- Atributos de um bom software

## # O processo

- Definição de processo, método e ferramenta
- Visão geral da Engenharia do Software
- O processo do software
- Modelos de processo
  - Linear sequencial
  - Prototipos
  - DRA
  - Modelos evolutivos

## #Produto e processo

# O produto

---

- # O software é simultaneamente um produto
  - ▣ software de gestão
  - ▣ pacotes de uso geral (folhas de cálculo, processadores de palavras)
- # e um veículo para o entregar
  - ▣ sistemas operativos,
  - ▣ software de comunicação
  - ▣ Ambientes de desenvolvimento
  - ▣ Ferramentas CASE

# Evolução do software

## # Antes

- Hardware
    - Caro
    - Poder Computacional Limitado
  - Programas simples
- ## # Preocupação:
- aproveitamento dos recursos de hardware i.e. Eficiência do produto

## # Depois

- Hardware
    - mais barato
    - e poderoso
  - Software mais complexo
- ## # Preocupação:
- Satisfação dos requisitos do utilizador (produto eficaz)
  - Baixar custos e tempo de desenvolvimento (processo eficiente)

# Características do software

- # O software é um produto lógico, não físico e portanto,
  - ▣ não é fabricado num sentido clássico, senão desenvolvido
  - ▣ O software não se estraga
  - ▣ Embora cada vez o software é uma assemblagem de componentes, ainda se faz muito software à medida

# Tipos de software

---

- # Genérico
- # Feito à medida
- # Problemas do desenvolvimento são os mesmos
- # A diferencia reside na especificação dos requisitos

# Aplicações do software

- # Software de sistemas
- # Software de tempo real
- # Software de gestão
- # Software de engenharia e científico
- # Software embebido
- # Software de PC's
- # Software baseado na WEB
- # Software de IA

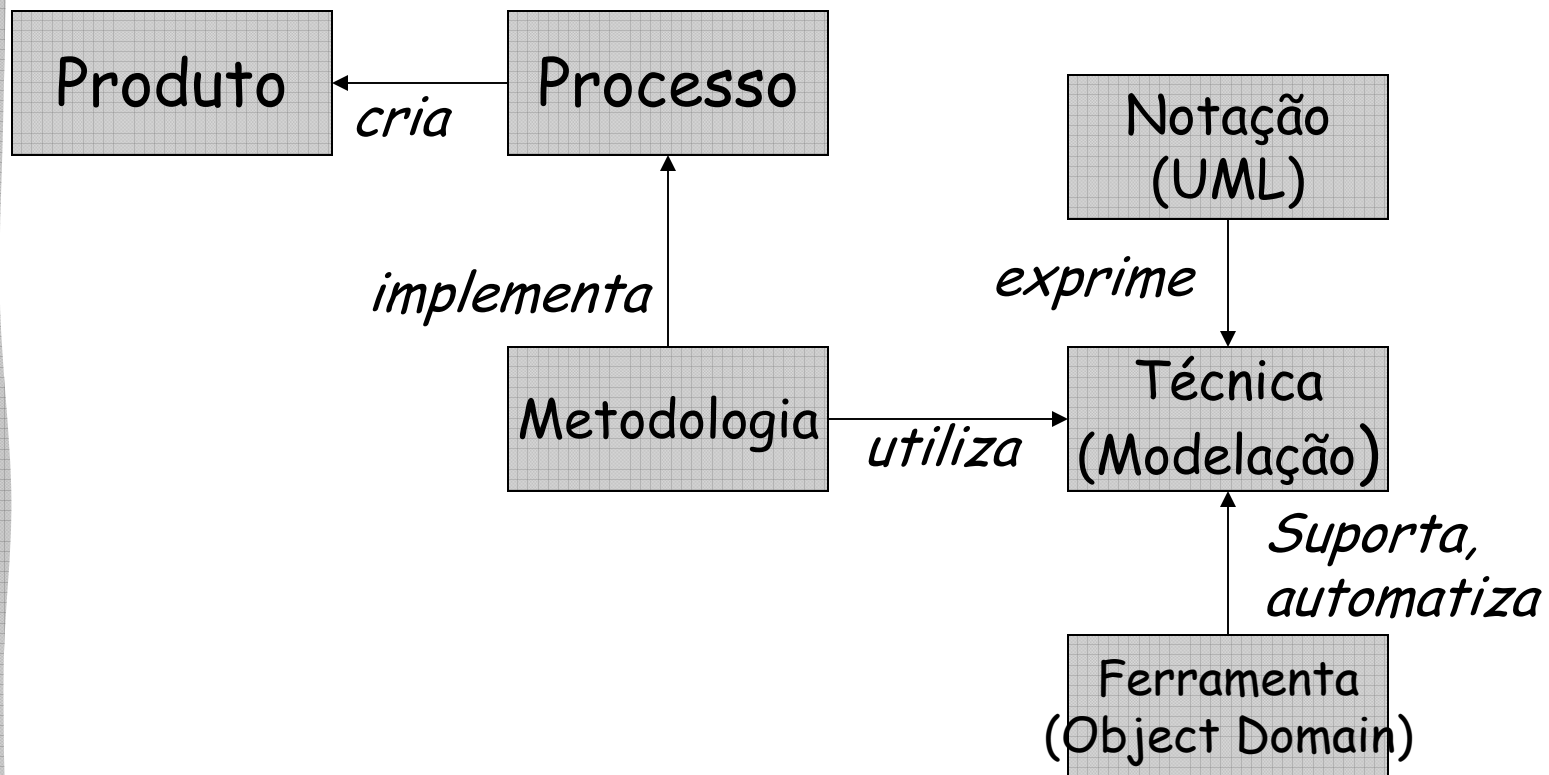
# Atributos de um bom software

| <i>Atributo</i>        | <i>Descrição</i>  |
|------------------------|---|
| <i>Maintainability</i> | O software deve ser capaz de evoluir perante as mudanças nos requisitos |
| <i>Dependability</i>   | Características como Confiabilidade e seguridade                        |
| <i>Eficiência</i>      | Aproveitamento dos recursos de hardware                                 |
| <i>Usabilidade</i>     | Apropriadas interface do utilizador e documentação                      |

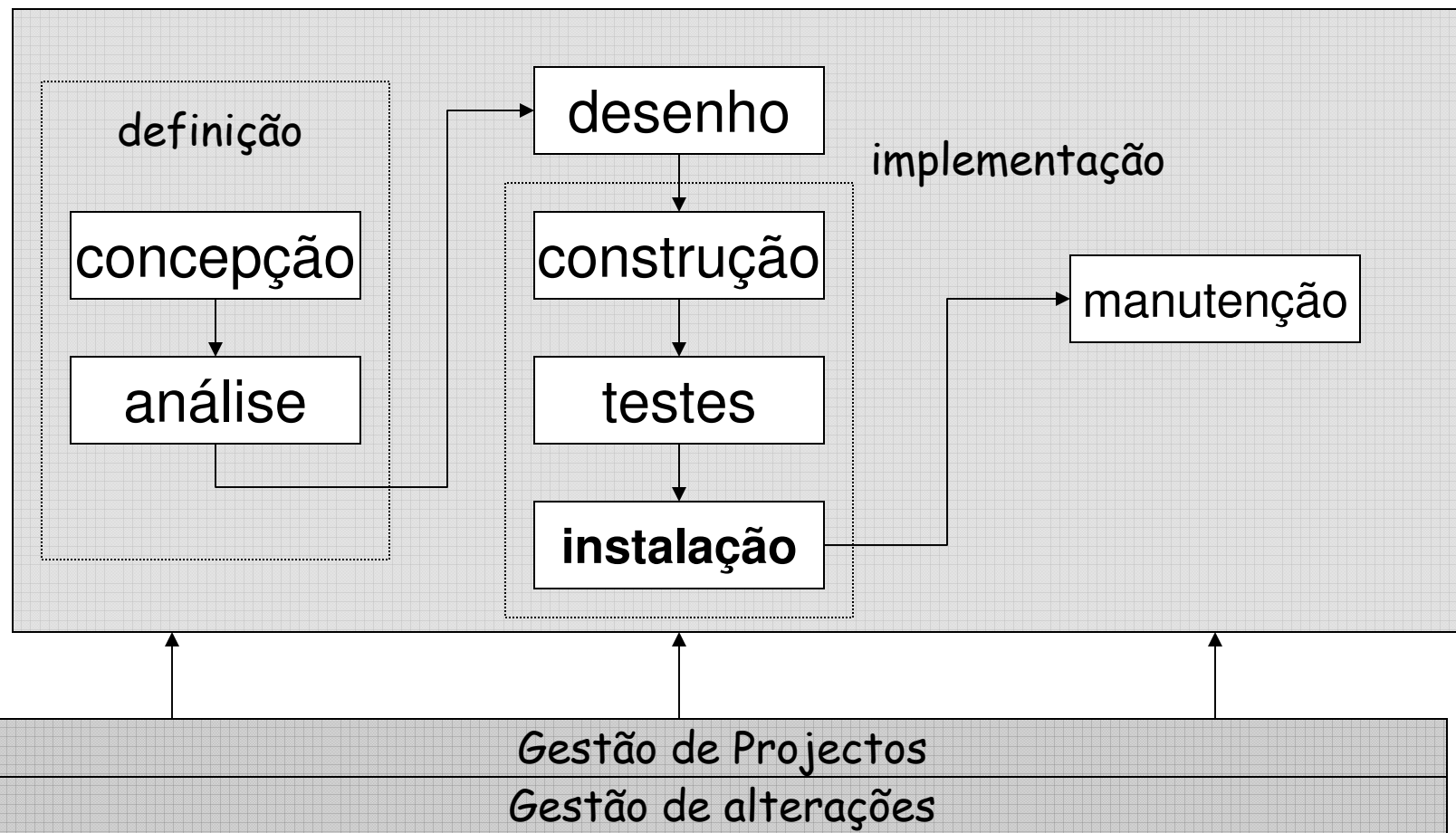
# Estrutura de conceitos (I)

- # A criação de um *produto* de software implica um *processo*
- # Processo: Conjunto de actividades e resultados associados que produzem um produto de software
- # Metodologia: descreve um processo incluindo a utilização de um conjunto de ferramentas, técnicas e notações específicas (Unified Software Development Process)
- # Os resultados associados são especificados em termos de *modelos*
- # Modelo: é uma interpretação da realidade exprimido através de uma linguagem textual ou gráfica.
- # Técnicas, Notações e Ferramentas: A modelação é uma técnica, o UML é uma notação e o Object Domain, uma ferramenta

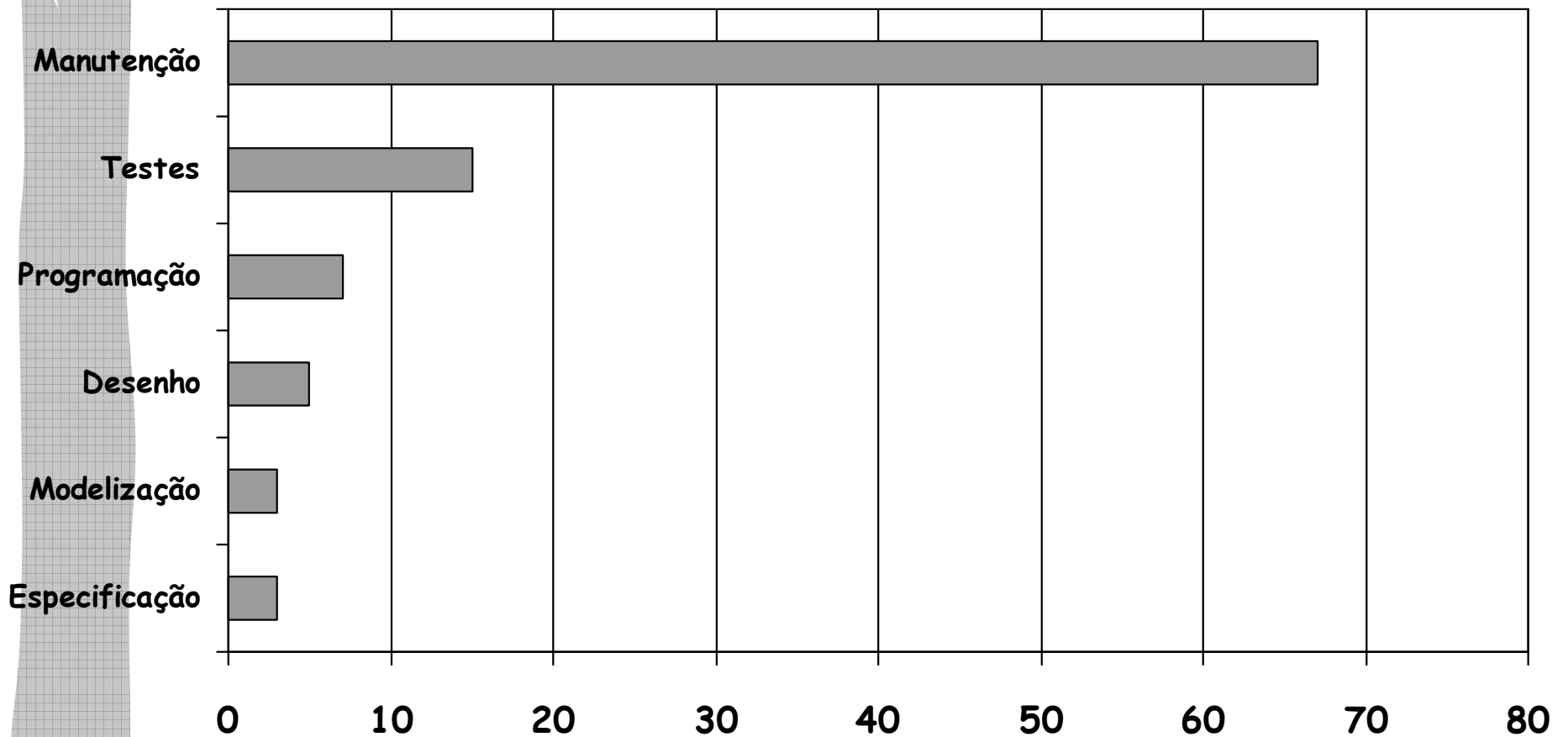
# Estrutura de conceitos (II)



# Fases do processo...



# Custos relativos do processo..



# Uma visão diferente... do problema à solução

| <i>Actividade</i>                     | <i>Determina...</i>     |
|---------------------------------------|-------------------------|
| <i>Concepção</i>                      | O enunciado do Problema |
| <i>Análise</i>                        | O domínio do problema   |
| <i>Desenho</i>                        | O domínio da Solução    |
| <i>Construção, Testes, Instalação</i> | O domínio da realização |

# Modelos de processo

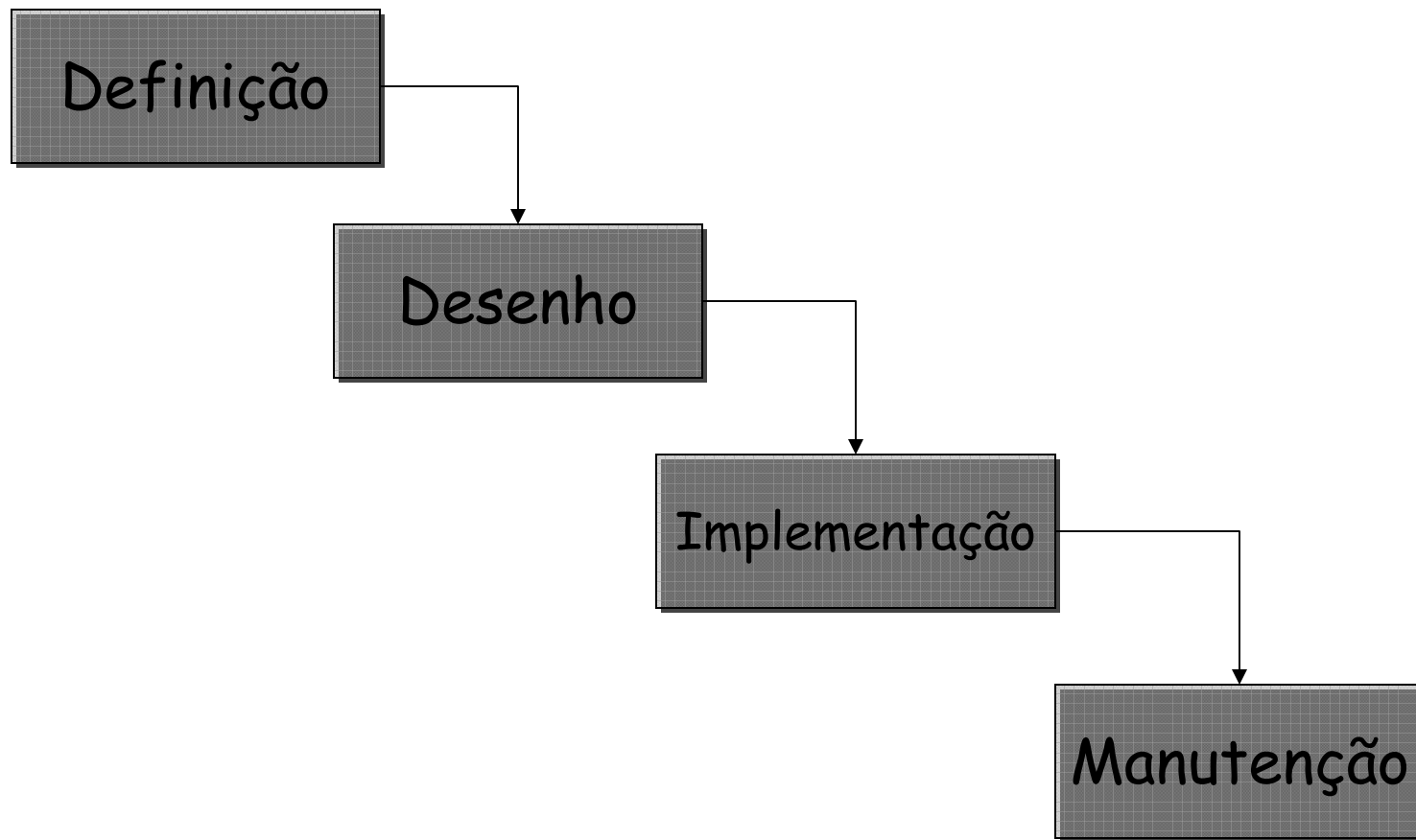
## # Definição:

- Estratégia de desenvolvimento, especialmente a ordem de execução das fases do processo

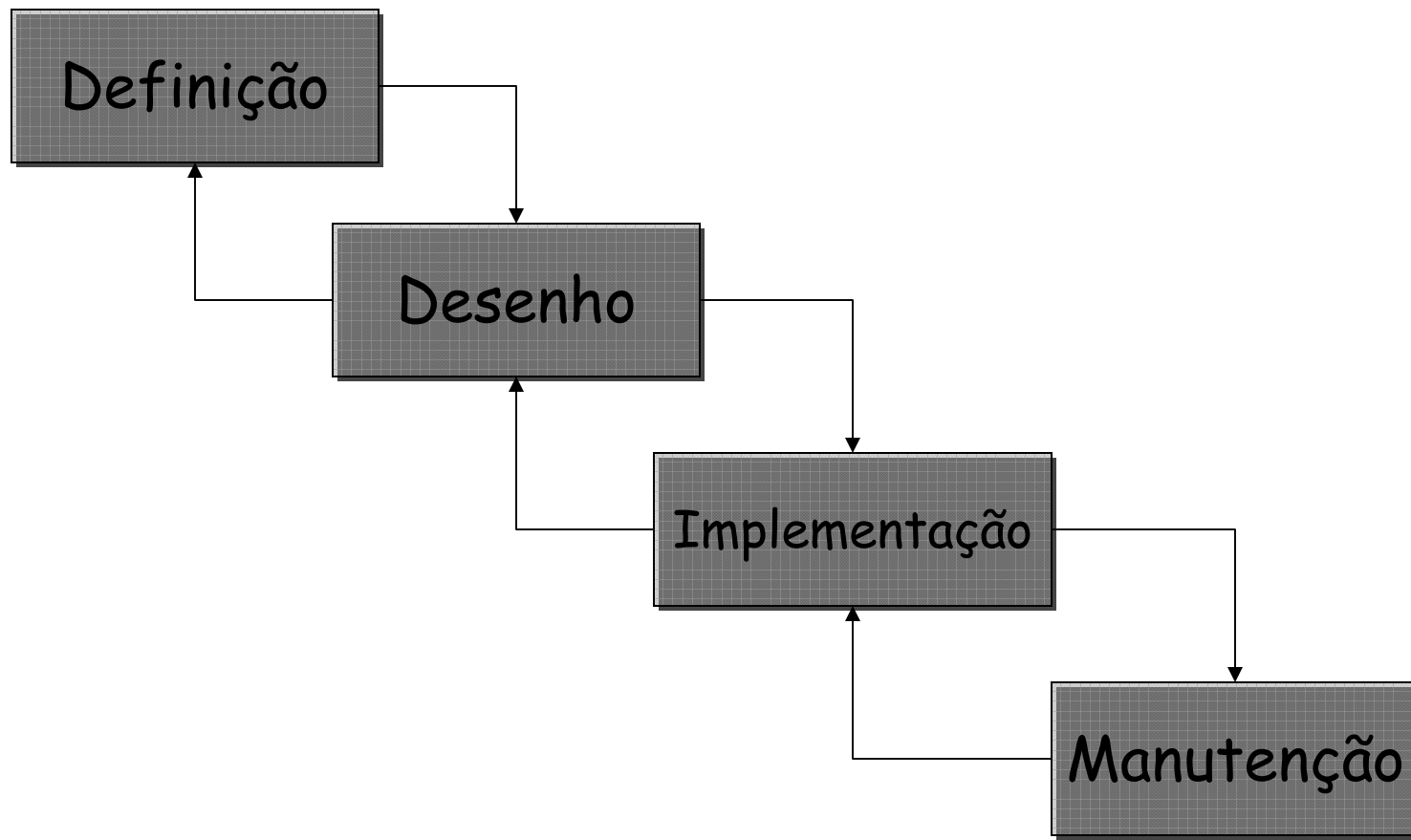
## # Tipos:

- Modelos em cascata
  - Sequencial
  - Revisto
- Construção de Protótipos
- Desenvolvimento Rápido de Aplicações
- Modelos evolutivos
  - Incrementais
  - Iterativos
  - Iterativos e incrementais
  - Espiral

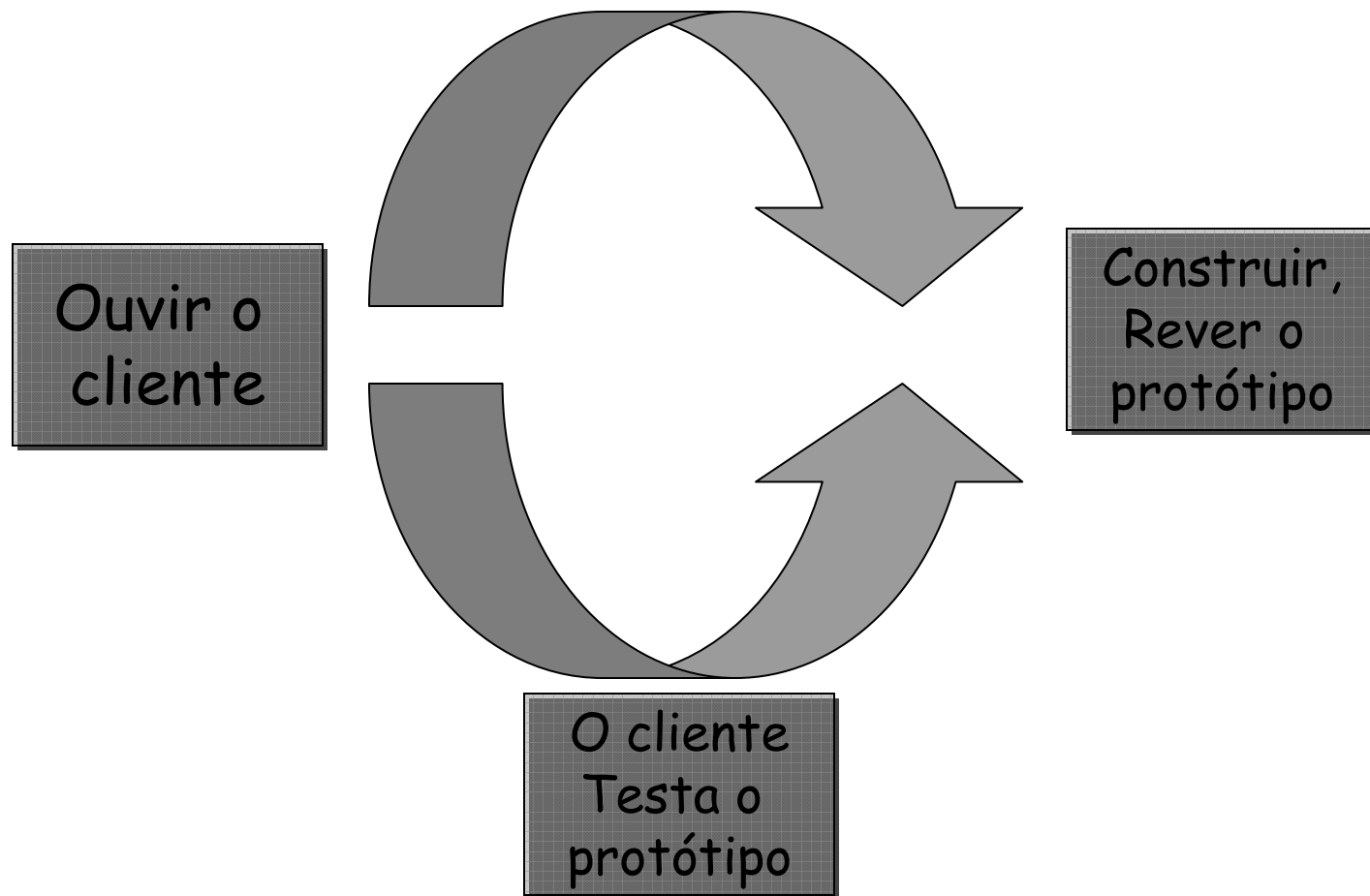
# Modelo Sequencial



# Modelo Revisto



# Construção de protótipos



# Desenvolvimento rápido de aplicações



# Modelo iterativo

## # Ideia:

- Desenvolvimento através de iterações sucessivas do ciclo de vida clássico
- Refinação progressiva do sistema
- O âmbito é definido no início e o não é alterado

## # Vantagens:

- Identificação atempada dos maiores riscos e dúvidas
- Encoraja a participação activa dos utilizadores
- Identificação atempada de inconsistência entre a análise, o desenho e a implementação
- Distribuição da participação dos envolvidos ao longo do projecto
- Aprendizagem e melhora continua do processo
- Provas inequívocas do avanço do projecto

# Modelo incremental

## # Ideia:

- ▣ Desenvolvimento através de incrementos sucessivas do âmbito do sistema
- ▣ O sistema é alargado progressivamente

## # Vantagens:

- ▣ Esta abordagem é útil para
  - ▣ Problemas complexos
  - ▣ Recursos humanos insuficientes
  - ▣ Datas de entrega inflexíveis

# Modelo iterativo e incremental

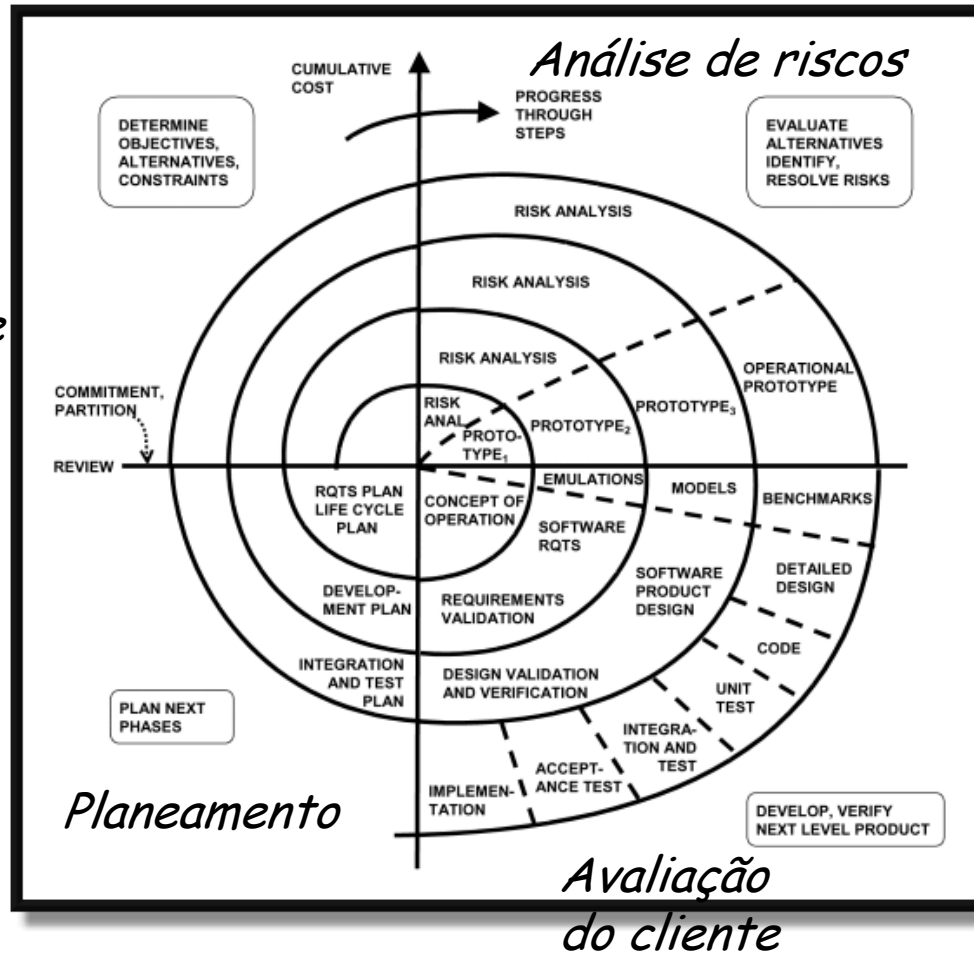
## # Ideia:

- ▣ Combina os 2 métodos anteriores
- ▣ Envolve uma refinação e alargamento sucessivo do sistema da sua qualidade, detalhe e âmbito

## # Vantagens:

- ▣ A maturação e completação progressiva dos produtos
- ▣ Cada iteração-incremento produz um conjunto de produtos utilizáveis

# Espiral de Boehm



*Comunicação com o cliente*

*Engenharia*

*Construção e adaptação*

# Modelo espiral de Boehm

## # Vantagens:

- ▣ Enfoque realista
- ▣ Reduz os riscos
  - ▣ Construção de protótipos
  - ▣ Consideração directa
- ▣ Mantém o enfoque sistemático do ciclo clássico

## # Desvantagens

- ▣ Custa crer que é controlável
- ▣ Requer boa capacidade de estimação de riscos
- ▣ Pouco utilizado

# Outros modelos

- # Win-Win
  - ▣ Negociação com o cliente
- # Desenvolvimento concorrente
  - ▣ Actividades concorrentes
  - ▣ Diagramas de estado
- # Métodos formais
  - ▣ Especificação matemática
  - ▣ Exacta e rigorosa
  - ▣ Detecta e corrige requisitos incompletos, ambíguos e inconsistentes
- # Baseado em componentes
  - ▣ Unified Development Process
  - ▣ Derivado da orientação a objectos
  - ▣ Utiliza UML
- # Ferramentas de 4ª Geração
  - ▣ Suporte automatizado à especificação de requisitos

# Produto vs. Processo

---

- # Se o processo é fraco, o produto vai sofrer
- # A dependência excessiva no processo também é má
- # A comunidade alterna a sua preocupação pelo produto e processo cada 5 ou 10 anos aprox.
- # Esta dualidade vai se manter por uns tempos