

---

# Engenharia de Software

Processos de Desenvolvimento

**Luís Morgado**

Instituto Superior de Engenharia de Lisboa  
Área Departamental de Engenharia de Electrónica e Telecomunicações e de Computadores

---

## Processo de desenvolvimento de software

- Processo de organização do trabalho de produção de software
- Define e organiza das actividades a realizar
- Utiliza métodos de desenvolvimento e ferramentas de suporte

- **Processo**



Suporte

- **Métodos**

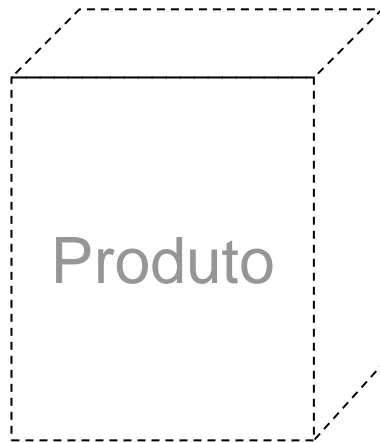


Suporte

- **Ferramentas**

# Processo de Desenvolvimento

Objectivo



Concretização



## Processo

# Processo de Desenvolvimento

- O que fazer?

- Artefactos

- Como fazer?

- Actividades

- Quem faz?

- Participantes

- Processo

- Define os **artefactos** (documentação, modelos, programas, ...) que devem ser realizados
  - Define as **actividades** envolvidas e a respectiva sequência de realização
  - Guia e enquadra as tarefas dos **participantes** como um todo coerente
  - Providencia critérios para **monitorização** e **avaliação** dos artefactos e actividades realizadas

## Monitorização



Foco no  
objectivo

# Processo de Desenvolvimento

## Um processo de desenvolvimento é realmente necessário?

Sem um **processo sistemático**, independentemente de qual seja, o desenvolvimento de sistemas, em particular de software, **ocorre sem rumo**, sendo **impossível garantir qualidade**, ou o atingir do que se pretende

- Complexidade predomina
- Confusão, anarquia, desorganização
- Esforço elevado
- Risco elevado
- Qualidade baixa

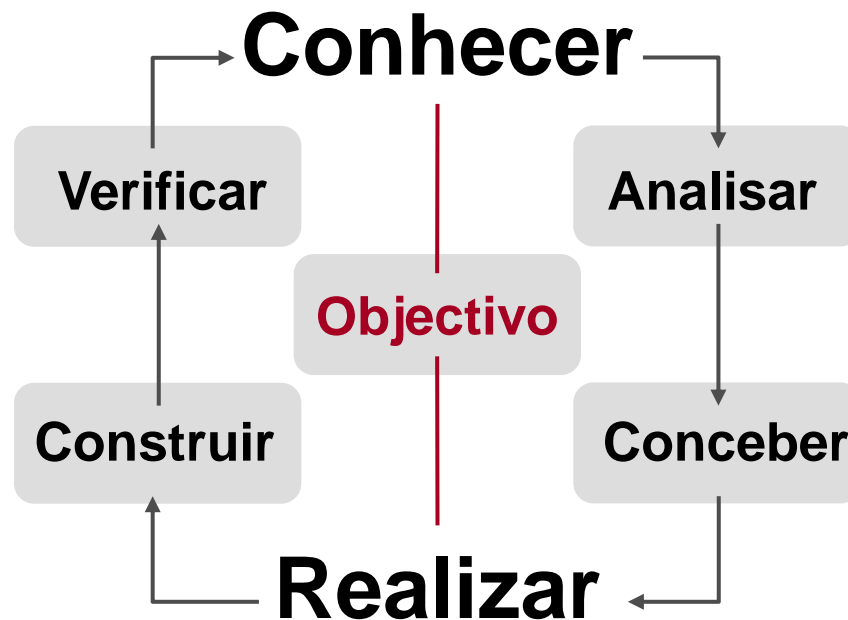
# Processo de Desenvolvimento

## Tentativa – Erro

- Parte de uma ideia vaga do que é pretendido
- Ausência de método sistemático de trabalho
- Reagir, corrigir, remediar são a regra
- Caos, ineficiência
- Deficiente gestão de recursos
- Incumprimento de prazos
- Custos não controlados
- Baixa qualidade
- ...

# Processo de Desenvolvimento

## Processo sistemático



**Foco no objectivo  
a atingir**

**Comunicação**

**Planeamento**

**Disponibilização**

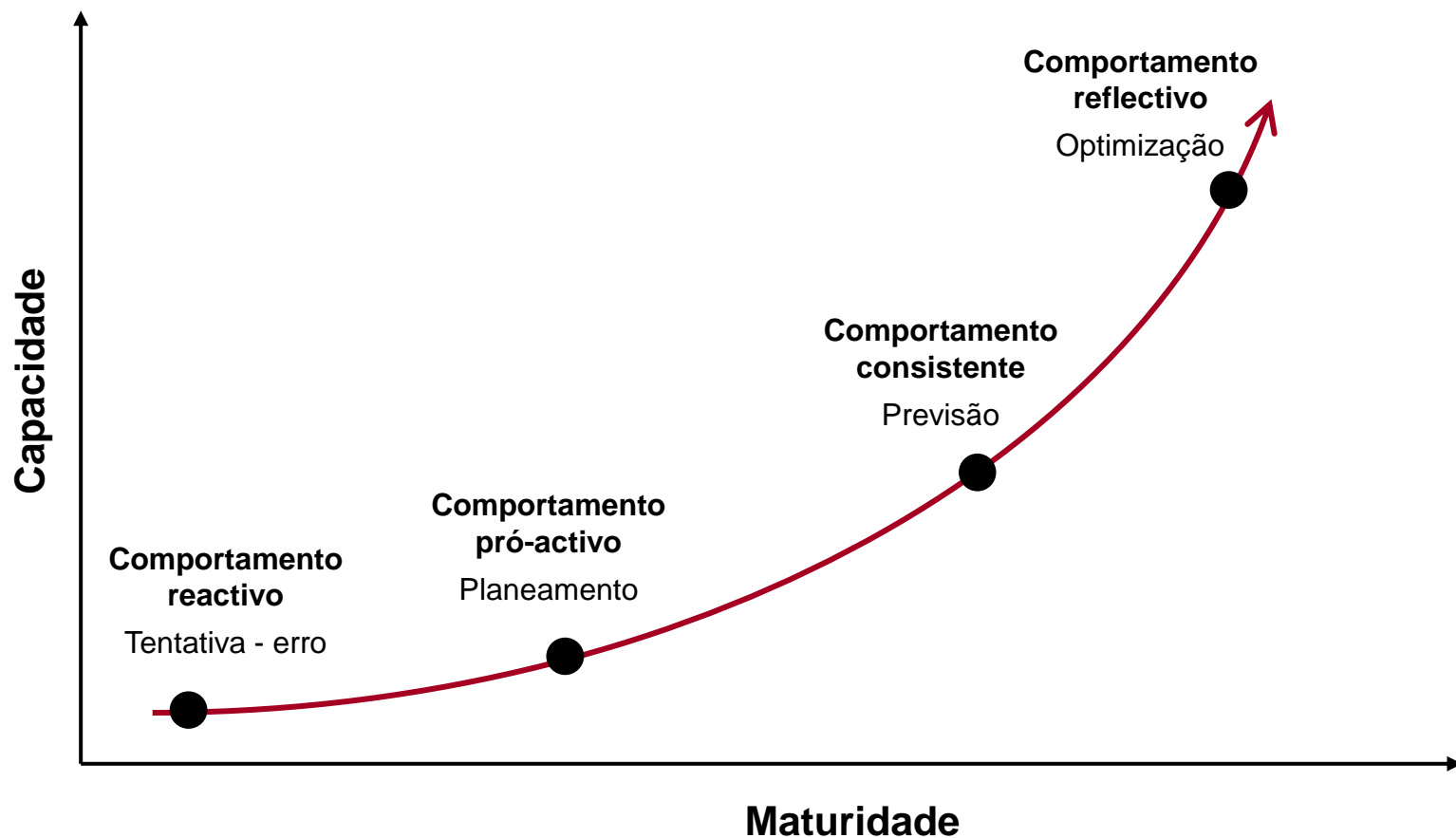
**Suporte**

**Operação**

**Manutenção**

# Processo de Desenvolvimento

## Maturidade e Capacidade (aptidão, valor)

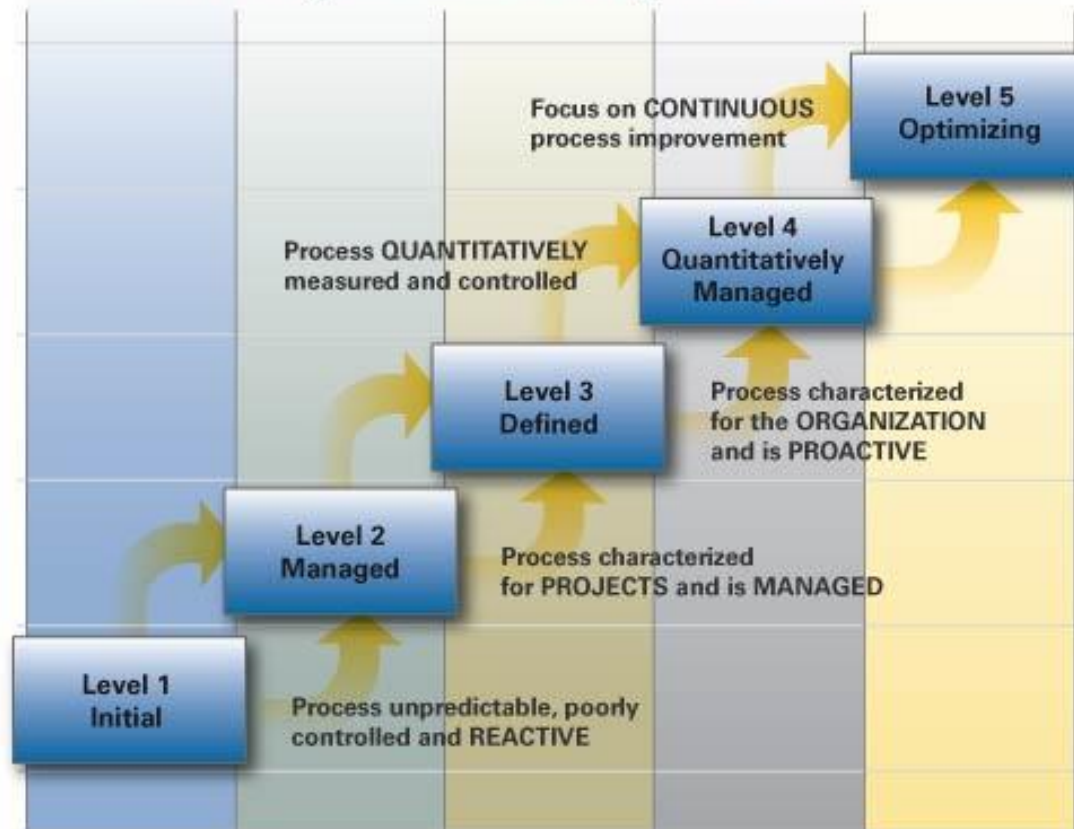




# Capability Maturity Model



## CMMI Staged Maturity Levels



[Software Engineering Institute]

# Capability Maturity Model

Maturity level	Description
Initial	<p>Processes are often ad hoc or chaotic with performance often unpredictable. Success is often due to the heroics of people rather than having high-quality processes in place. The defined process is often abandoned in times of crisis, and there are no audits to enforce the process</p> <p>It is difficult to repeat previous success since success is due to heroic efforts of its people rather than processes. These organizations often over-commit, as they often lack an appropriate estimation process on which to base project commitments</p> <p>Firefighting is a way of life in these organizations. High-quality software might be produced, but at a cost including long hours, high level of rework, over budget and schedule and unhappy staff and customers. Projects do not perform consistently as their success is dependent on the people involved</p> <p>They may have few processes defined and poor change control, poor estimation and project planning and weak enforcement of standards</p>

[O'Regan, 2017]

# Capability Maturity Model

Maturity level	Description
Managed	<p>A level two organization has good project management practices in place, and planning and managing new projects is based on experience with similar previous projects</p> <p>The process is planned, performed and controlled. A level two organization is disciplined in following processes, and the process is enforced with independent audits</p> <p>The status of the work products produced by the process is visible to management at major milestones, and changes to work products are controlled. The work products are placed under appropriate configuration management control</p> <p>The requirements for a project are managed and changes to the requirements are controlled. Project management practices are in place to manage the project, and a set of measures are defined for the budget, schedule and effort variance. Subcontractors are managed</p> <p>Independent audits are conducted to enforce the process. The processes in a level two organization are defined at the project level</p>

[O'Regan, 2017]

# Capability Maturity Model

Maturity level	Description
Defined	<p>A maturity level three organization has standard processes defined that support the whole organization</p> <p>These standard processes ensure consistency in the way that projects are conducted across the organization. There are guidelines defined that allow the organization process to be tailored and applied to each project</p> <p>There are standards in place for design and development and procedures defined for effective risk management and decision analysis</p> <p>Level 3 processes are generally defined more rigorously than level 2 processes, and the definition includes the purpose of the process, inputs, entry criteria, activities, roles, measures, verification steps, exit criteria and output. There is also an organization-wide training program and improvement data is collected</p>

[O'Regan, 2017]

# Capability Maturity Model

Maturity level	Description
Quantitatively managed	<p>A level 4 organization sets quantitative goals for the performance of key processes, and these processes are controlled using statistical techniques. Processes are stable and perform within narrowly defined limits. Software process and product quality goals are set and managed.</p> <p>A level 4 organization has predictable process performance, with variation in process performance identified and the causes of variation corrected.</p>

[O'Regan, 2017]

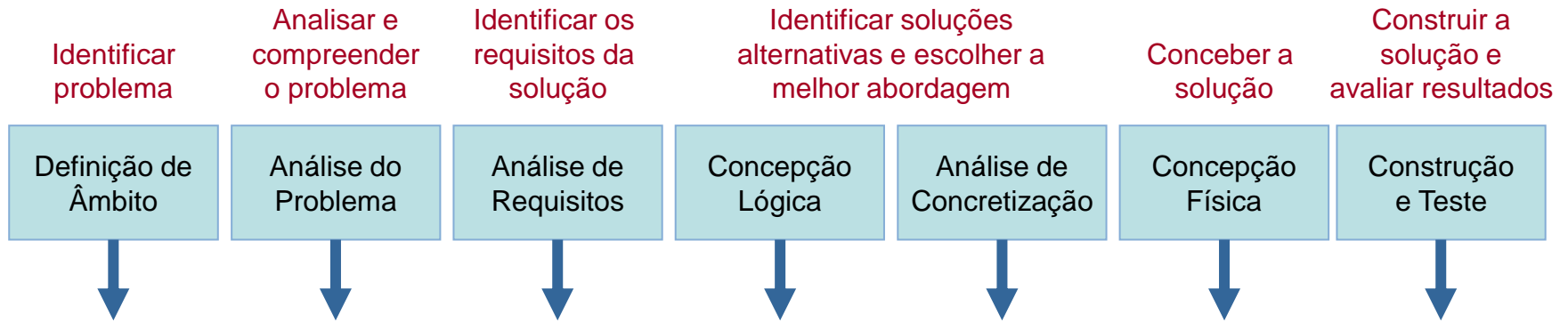
# Capability Maturity Model

Maturity level	Description
Optimizing	<p>A level 5 organization has a continuous process improvement culture in place, and processes are improved based on a quantitative understanding of variation</p> <p>Defect prevention activities are an integral part of the development lifecycle. New technologies are evaluated and introduced (where appropriate) into the organization. Processes may be improved incrementally or through innovative process and technology improvements</p>

[O'Regan, 2017]

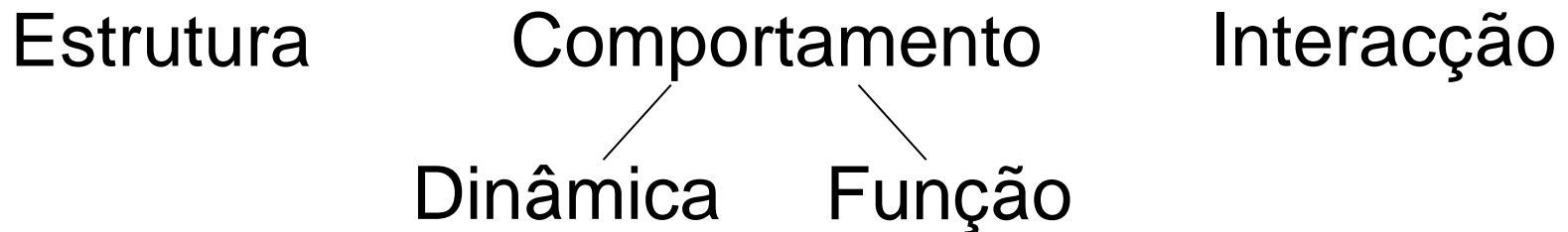


# Desenvolvimento de Software



## Artefactos

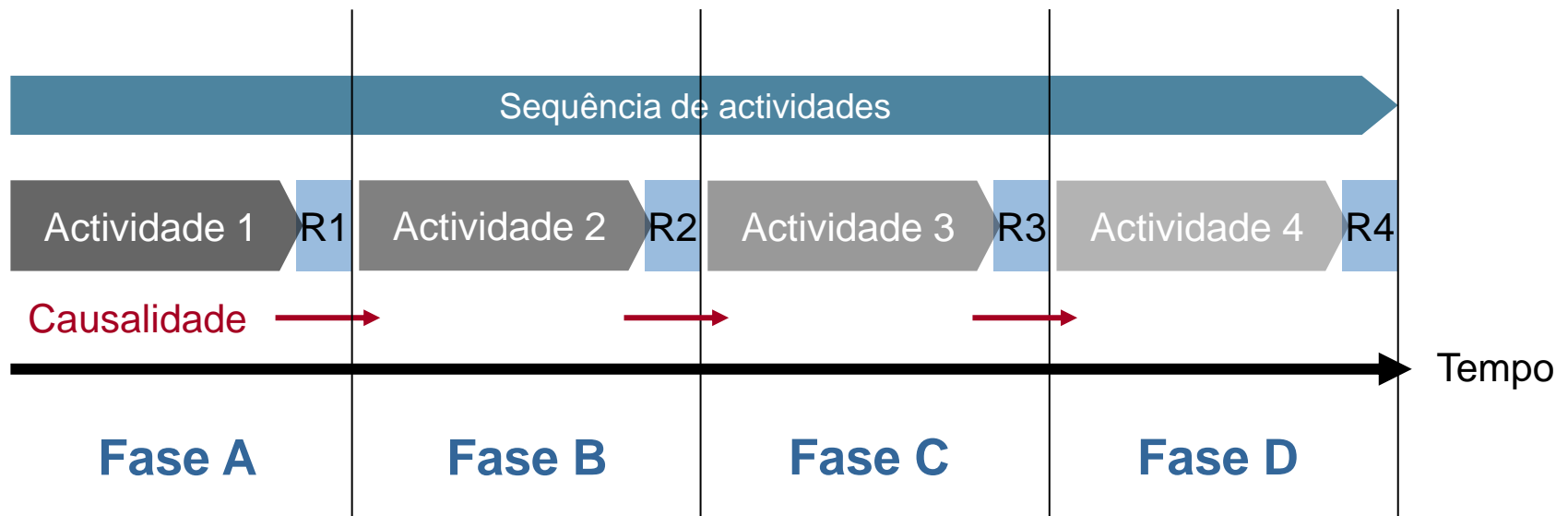
### Perspectivas de descrição



# Processo de Desenvolvimento

## Modelo linear

- Define um encadeamento sequencial na realização das actividades
- Organizado em fases
  - Delimitadas por marcos de desenvolvimento (*milestones*)



$R_i$  – Resultado de actividade / marco de desenvolvimento (*milestone*)



# Processo de Desenvolvimento

- Fases de actividade

1. **Início**

- Produz uma **descrição do problema** que define o âmbito, objectivos, prazos e orçamento para realização de uma solução para o problema

2. **Análise**

- Produz uma **definição dos requisitos**, expectativas e prioridades dos utilizadores para uma solução do problema

3. **Concepção**

- Produz a **arquitectura** e as especificações para uma solução que satisfaça os requisitos

4. **Construção**

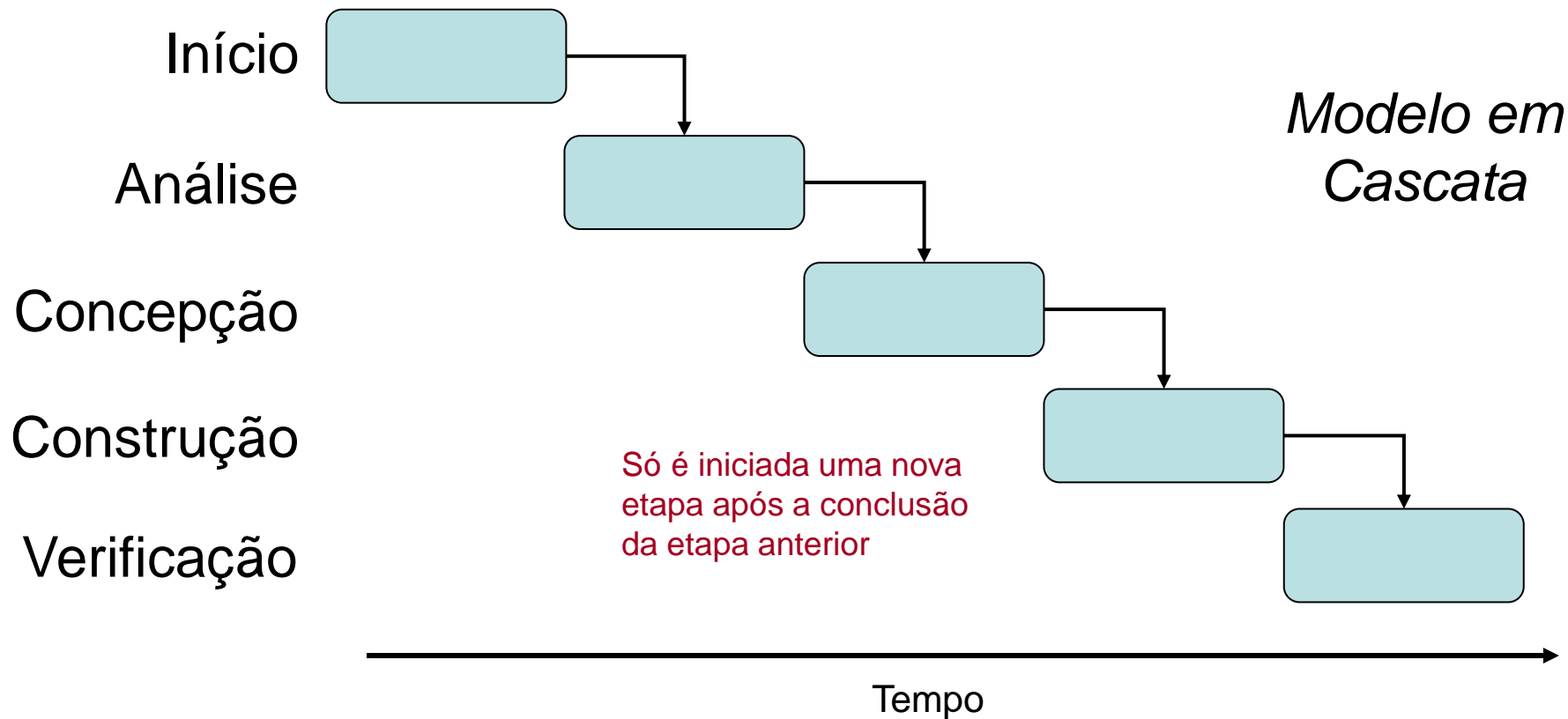
- Produz o **sistema** que concretiza a solução de acordo com a arquitectura e as especificações definidas

5. **Verificação**

- Produz **informação de validação** referente à concretização dos requisitos definidos

# Processo de Desenvolvimento

## Desenvolvimento linear



**Foco: Tarefas a realizar**

# Modelo em Cascata

- Características
  - O mais antigo e mais utilizado
  - Bem documentado e padronizado
  - Ênfase no planeamento antes da construção
  - Escalonamento linear das tarefas facilita a gestão
  - Conteúdo dos documentos e critérios de revisão bem definidos
  - Separação clara entre actividades
    - Análise / Conceção / Construção / Verificação
  - Favorece uma abordagem do geral para o específico (“*top-down*”)

# Modelo em Cascata

- Problemas

- Os projectos reais raramente tem um encadeamento sequencial
- Não há concorrência na concretização das diferentes etapas
- Não tem em conta a necessidade de contínua adaptação à mudança
- Produto tardio

# Bibliografia

[Pressman, 2003]

R. Pressman, *Software Engineering: a Practitioner's Approach*, McGraw-Hill, 2003.

[Weaver, 1948]

W. Weaver, *Science and Complexity*, American Scientist, 36: 536, 1948.

[Hitchins, 1992]

D. Hitchins, *Putting Systems to Work*, John Wiley, 1992.

[Boehm, 2000]

B. Boehm, *Software Cost Estimation with COCOMO II*, 2000.

[DAUP, 2001]

Systems Engineering Fundamentals. Defense Acquisition University Press, 2001

[O'Regan, 2017]

G. O'Regan, *Concise Guide to Software Engineering: From Fundamentals to Application Methods*, Springer, 2017