
Engenharia de Software

Arquitectura de Software

Luís Morgado

Instituto Superior de Engenharia de Lisboa
Departamento de Engenharia de Electrónica e Telecomunicações e de Computadores

Processo de Desenvolvimento

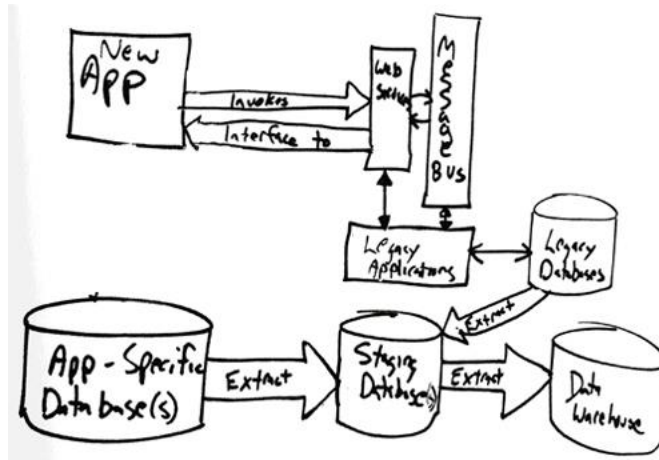
- **Análise**
- **Concepção**
 - Transição análise de requisitos – projecto de arquitectura de software
 - Analisar, compreender e verificar a forma de realizar a solução proposta
 - Elaborar a arquitectura da solução
 - Conceber e especificar a arquitectura da solução
- **Construção**
- **Verificação**

Arquitectura de Software

Architecture is defined by the recommended practice as the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.

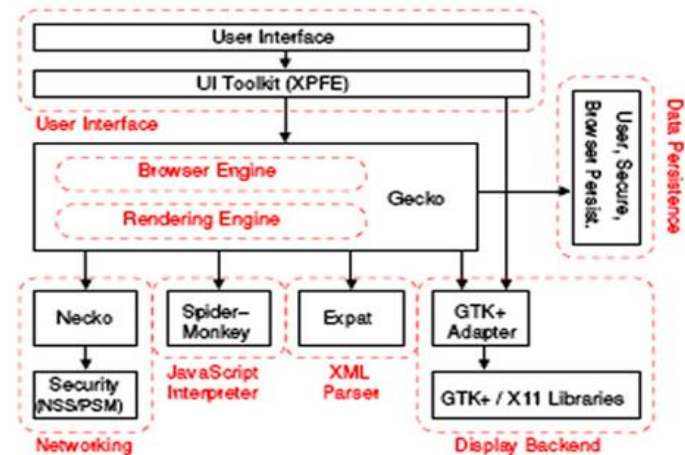
[ANSI/IEEE Std 1471-2000, *Recommended Practice for Architectural Description of Software-Intensive Systems*]

Enterprise Architecture



McGovern, J., Ambler, S., Stevens, M., Linn, J., Sharan, V., and Jo, E. *A Practical Guide to Enterprise Architecture*. Upper Saddle River, New Jersey: Prentice Hall, 2004

Mozilla Web Browser



Grosskurth, A. and Godfrey, M. "A Reference Architecture for Web Browsers." *IEEE Conference on Software Maintenance*, 2005

- Especificação de um sistema de software
 - **Partes**
 - Componentes
 - Encapsulam um conjunto coerente de funcionalidades
 - Elementos estruturais
 - Interfaces
 - Comportamento
 - **Relações** entre partes
 - Ligações, conectores
 - Realizam a interligação entre componentes
 - **Princípios**
 - Guiam a concepção e organização

Níveis de Modelação

- **Três níveis principais de modelação**

- **Independente do Modelo Computacional (CIM)**

- Também designado ***Modelo de Negócio*** ou de ***Domínio***, descreve o contexto de utilização do sistema e o seu comportamento e características esperadas



- **Arquitectura conceptual**

- **Independente da Plataforma de Execução (PIM)**

- Descreve o sistema com tanto detalhe quanto possível de forma independente da plataforma de execução



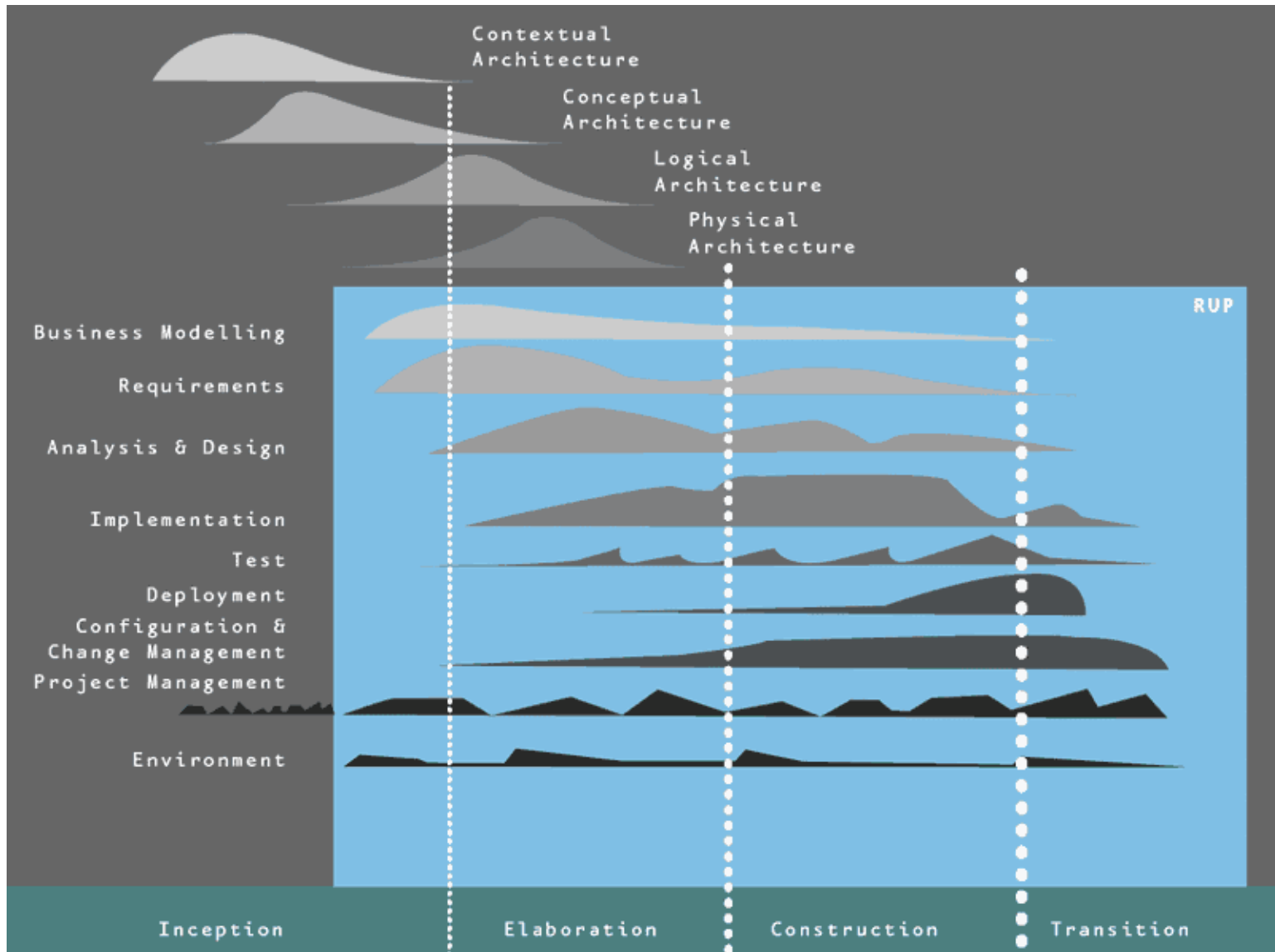
- **Arquitectura lógica**

- **Específico da Plataforma de Execução (PSM)**

- Descreve a concretização do sistema para uma **plataforma de execução específica**

- **Arquitectura detalhada**

Níveis de Arquitectura



[Macaulay, 2004]

Níveis de Arquitectura

- Subsistemas

- Agregados de mecanismos
- Funcionalidade global

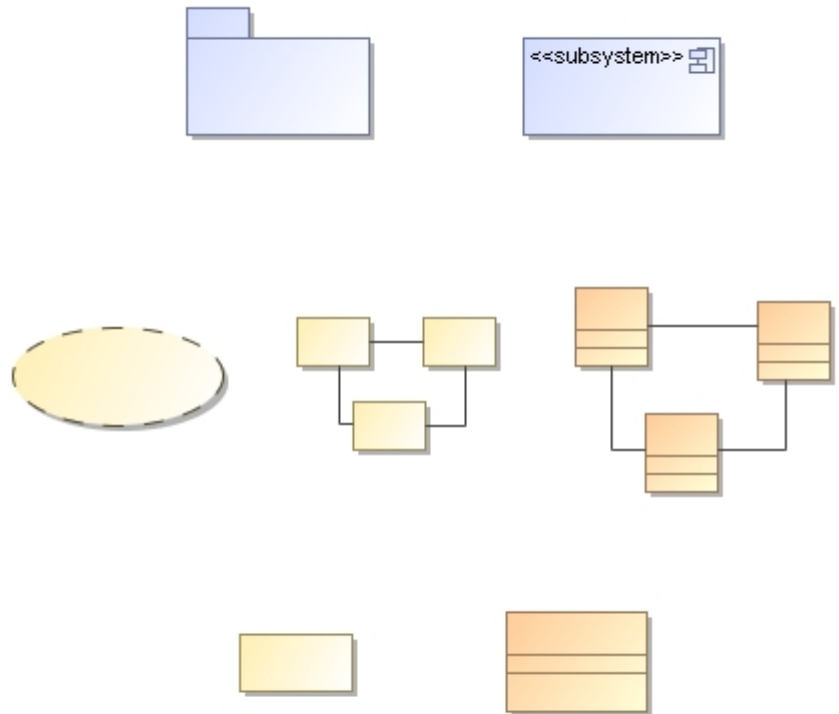
- Mecanismos

- Agregados de elementos
- Funcionalidade local

- Elementos

- Partes base
- Funcionalidade micro

UML



3 Níveis de Projecto

- Projecto de Subsistemas

Âmbito: Subsistemas, Processos, ...

O que é especificado:

- Organização do sistema
- Estratégias de concorrência e comunicação entre processos
- ...

Sistema

- Projecto de Mecanismos

Âmbito: Grupos de classes

O que é especificado:

- Instâncias de Padrões de Arquitectura ("*Design Patterns*")
- Utilização de classes Contentoras
- Estratégias de gestão de erros (nível intermédio)
- ...

Inter-objecto

- Projecto Detalhado

Âmbito: Classes, Interfaces, ...

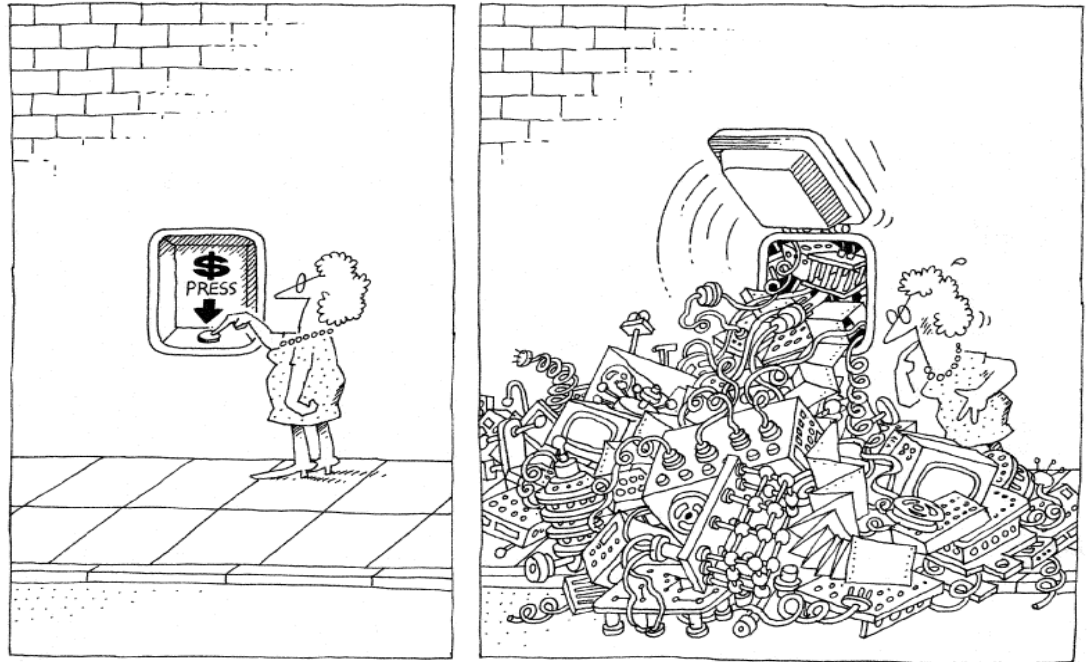
O que é especificado:

- Detalhes de implementação de atributos e operações
- Definição de algoritmos
- ...

Intra-objecto

Arquitetura de Software

- Métricas
- Princípios
- Padrões



[Booch, 2004]

Complexidade

- Redução
- Controlo

- **Métricas**
- Princípios
- Padrões

Métricas de Arquitectura

- **Coesão**

- Nível coerência funcional de um subsistema/módulo (até que ponto esse módulo realiza uma única função)

- **Acoplamento**

- Grau de interdependência entre subsistemas

- **Simplicidade**

- Nível de facilidade de compreensão/comunicação da arquitectura

- **Adaptabilidade**

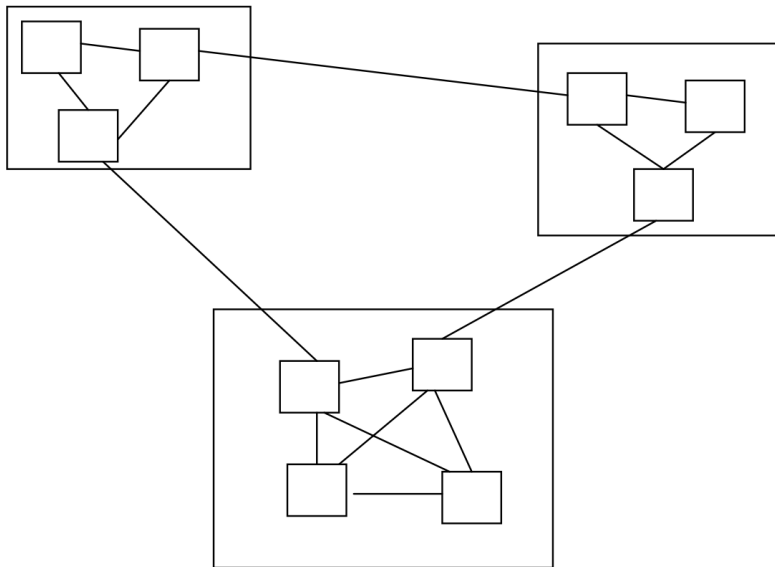
- Nível de facilidade de alteração da arquitectura para incorporação de novos requisitos ou de alterações nos requisitos previamente definidos

Coesão e Acoplamento

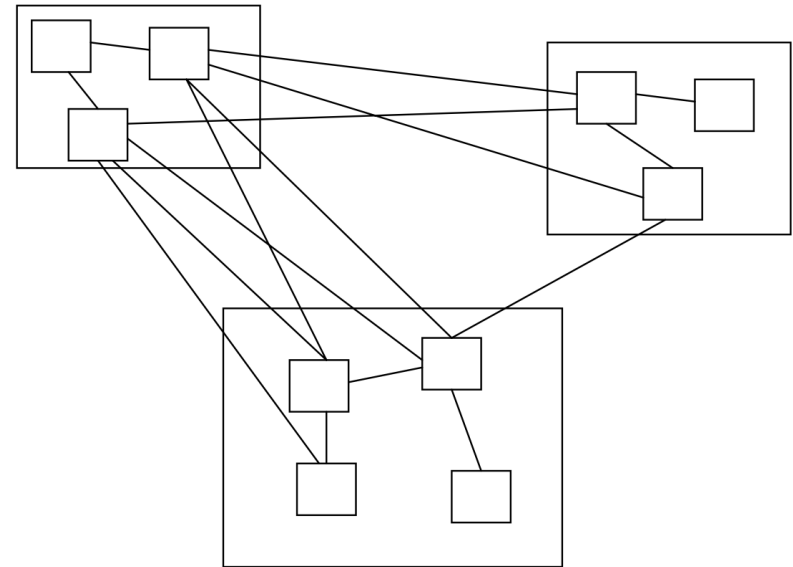
- **Coesão:**
 - Nível coerência funcional de um subsistema/módulo (até que ponto esse módulo realiza uma única função)
 - Característica **intra-modular**
- **Acoplamento:**
 - Grau de interdependência entre subsistemas
 - Característica **inter-modular**

Coesão e Acoplamento

Exemplo



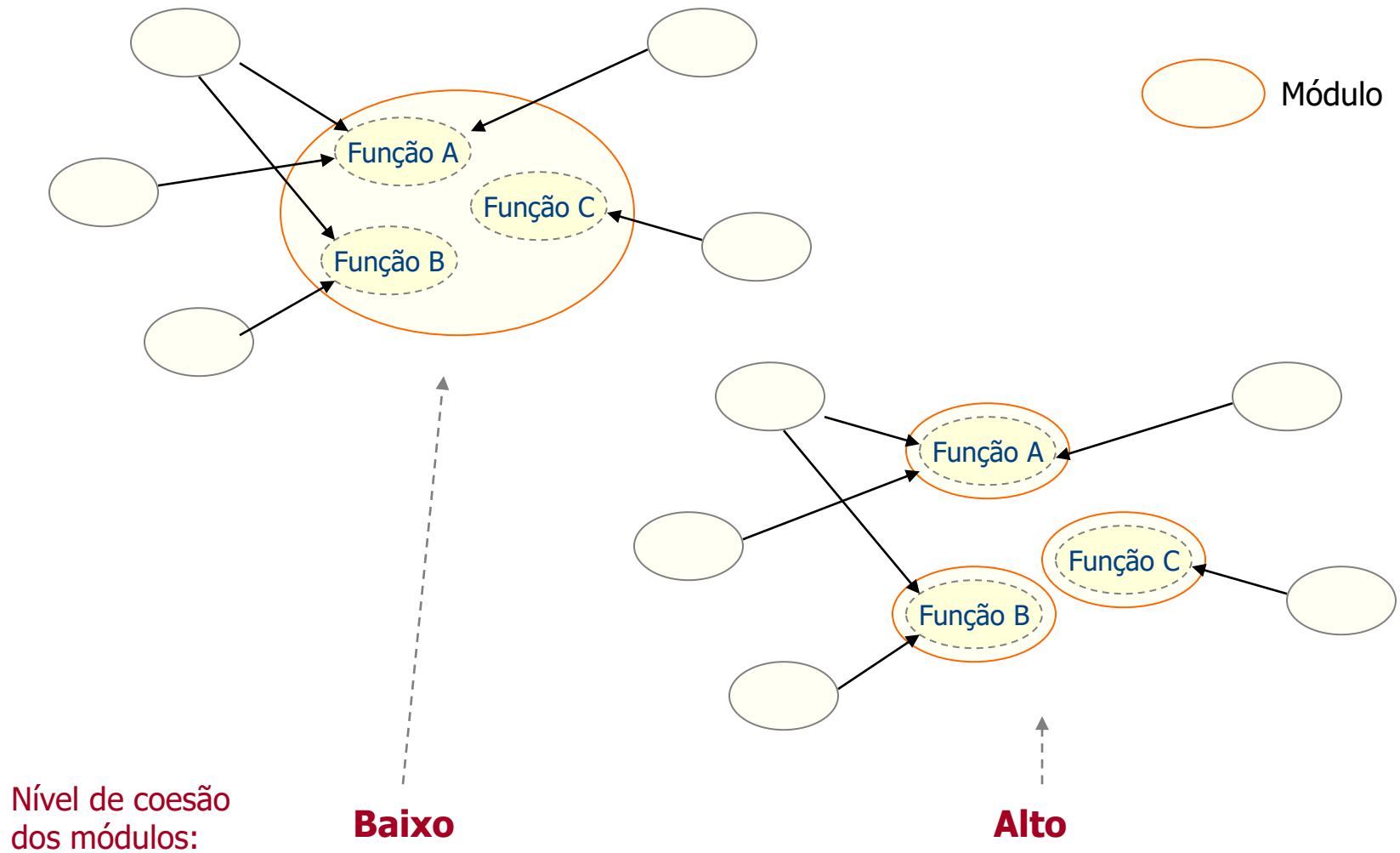
Alta coesão
Baixo acoplamento



Baixa coesão
Alto acoplamento

[Laplante, 2007]

Coesão (intra-modular)



Nível de coesão:

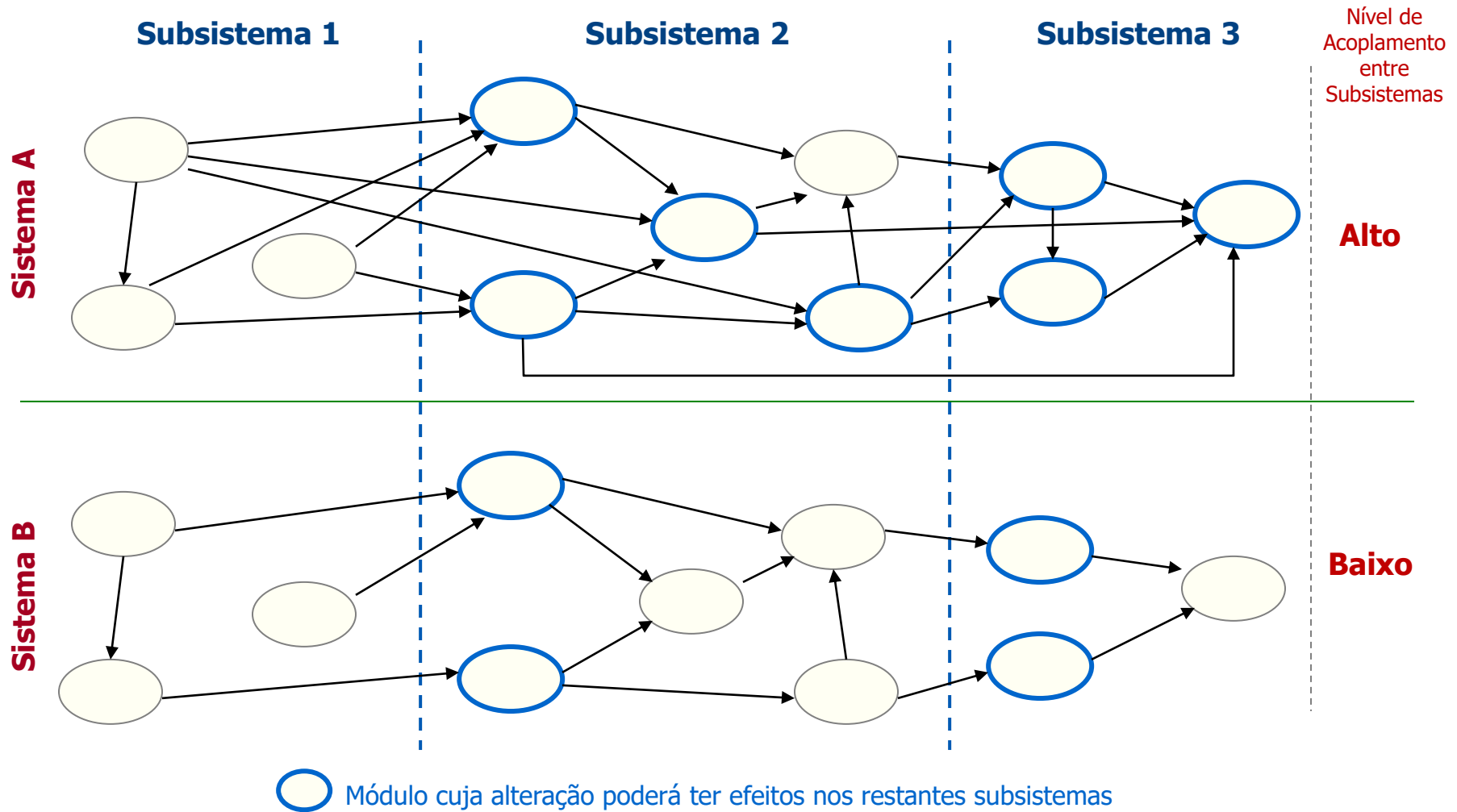
- Um módulo com um nível de coesão baixo é mais **complexo**, logo mais difícil de conceber e de testar
- Um nível de coesão baixo leva a que, em caso de necessidade de **alteração** de um subsistema, o **número de módulos afectados seja elevado**
- Se o nível de coesão for elevado, o número de módulos afectados será minimizado

- **Coesão lógica**
 - Partes do mesmo tipo são agrupadas num módulo
- **Coesão temporal**
 - Partes que são executadas em momentos próximos no tempo são agrupadas num módulo
- **Coesão funcional**
 - Partes que contribuem para uma função específica bem definida são agrupadas num módulo

Acoplamento

- **Acoplamento estrutural**
 - Uma parte depende estruturalmente de outra parte, podendo directamente utilizar, aceder ou alterar outra parte
- **Acoplamento estrutural comum**
 - Múltiplas partes podem aceder ou alterar uma parte comum
- **Acoplamento funcional**
 - Uma parte depende de um contracto funcional (interface), independentemente da parte que o implementa
- **Acoplamento denotacional (semântico)**
 - Uma parte depende do significado associado a uma característica ou funcionalidade, independentemente da parte que o implementa e do contracto funcional com que é definido

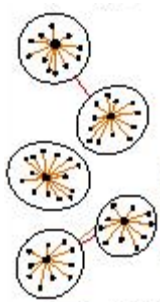
Acoplamento (inter-modular)



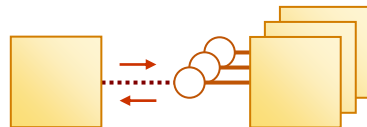
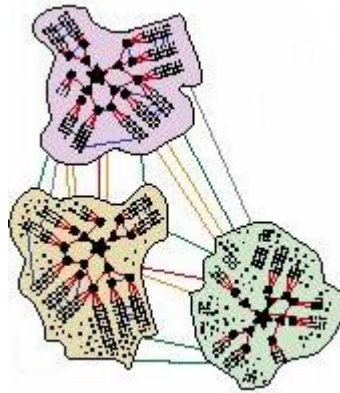
A redução do nível de acoplamento permite:

- **Maior facilidade de desenvolvimento, instalação, manutenção e expansão**
- **Melhor escalabilidade**, devido à possibilidade de distribuição e replicação de módulos que prestem serviços, sem que isso tenha um impacto significativo nos clientes desses subsistemas/módulos
- **Maior tolerância a falhas**, logo maior robustez, uma vez que a falha de um subsistema/módulo tem um impacto restrito

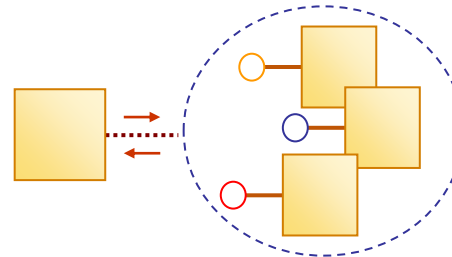
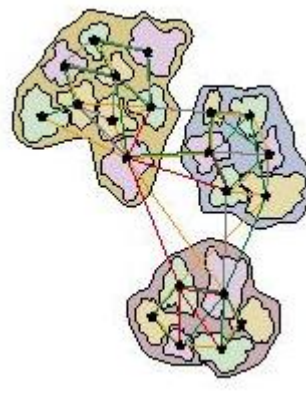
Tipos de Acoplamento



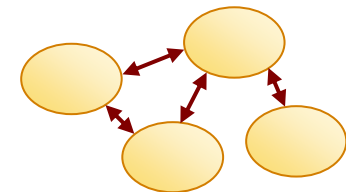
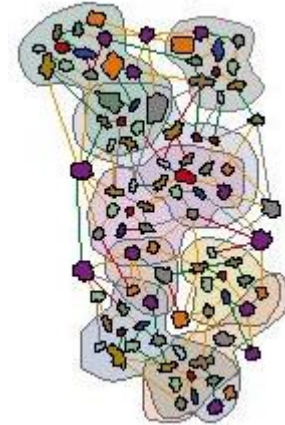
Acoplamento de
natureza estrutural



Acoplamento de
natureza funcional



Acoplamento de
natureza denotacional



Acoplamento de
natureza motivacional



Arquitectura de Software

- Métricas
- **Princípios**
- Padrões

Princípios de Arquitectura

- Abstracção
- Modularidade
- Encapsulamento
- Factorização

Subjacente

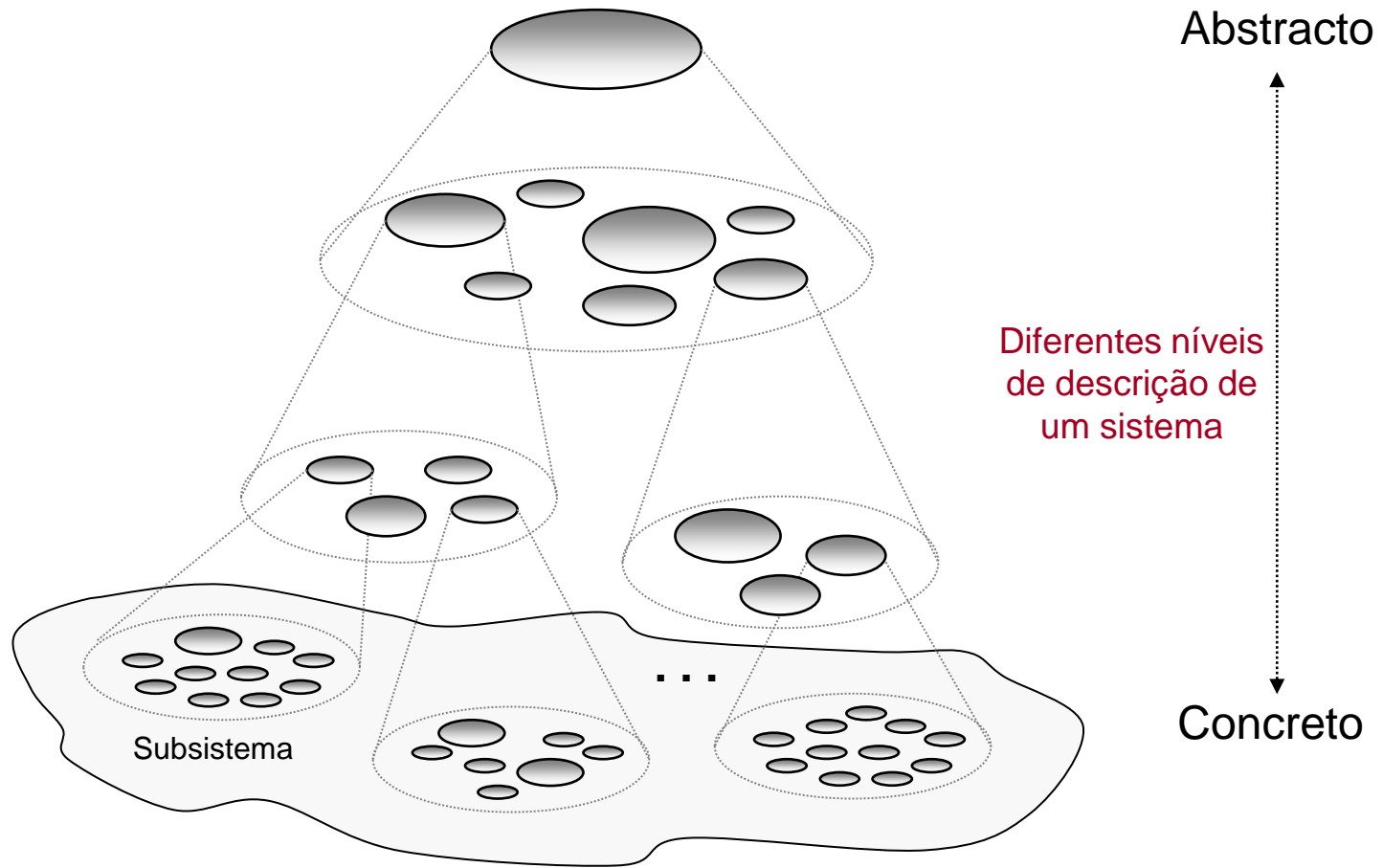
- Maximização da coesão
- Minimização do acoplamento

Abstracção

O processo de descrição de conhecimento a *diferentes níveis de detalhe* (**quantidade de informação**) e *tipos de representação* (**estrutura da informação**)

[Korf, 1980]

Abstracção



- **Abstracção é uma ferramenta base para lidar com a complexidade**
 - Identificação de características comuns a diferentes partes
 - Criação de ordem de forma progressiva
 - realçar o que é essencial, omitir detalhes não relevantes
 - Modelos
- **Desenvolvimento de um sistema complexo**
 - Processo iterativo guiado por conhecimento

Modularidade

- **Decomposição**

- De um sistema em partes coesas
 - Para sistematizar interacções
 - Para lidar com a explosão combinatória

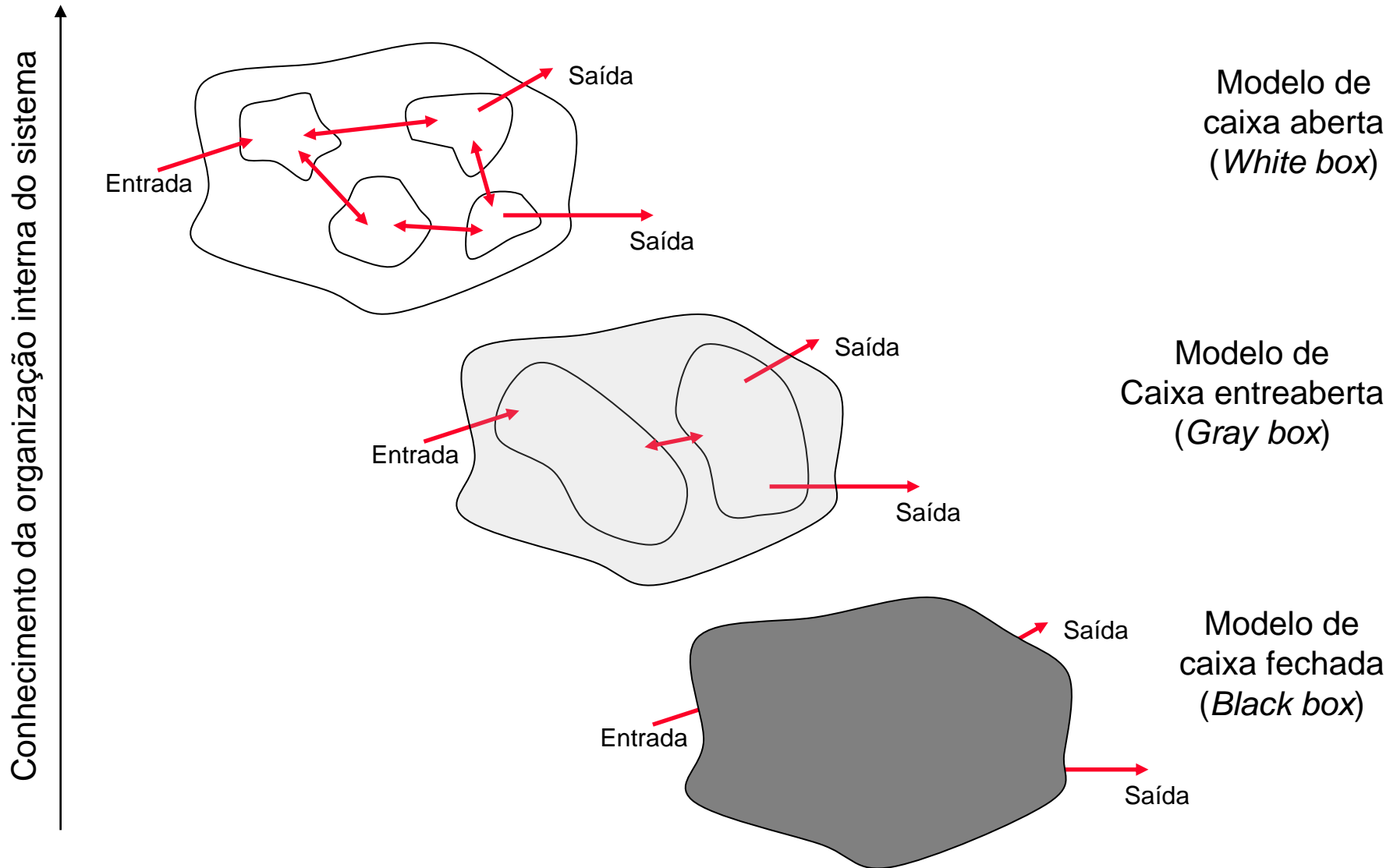
- **Encapsulamento**

- Isolamento dos detalhes internos das partes de um sistema em relação ao exterior
 - Para reduzir dependências (interacções)
 - Relacionar estrutura e função no contexto de uma parte
 - Acesso exclusivo através das interfaces disponibilizadas

- **Interfaces**

- Contractos funcionais para interação com o exterior

Encapsulamento

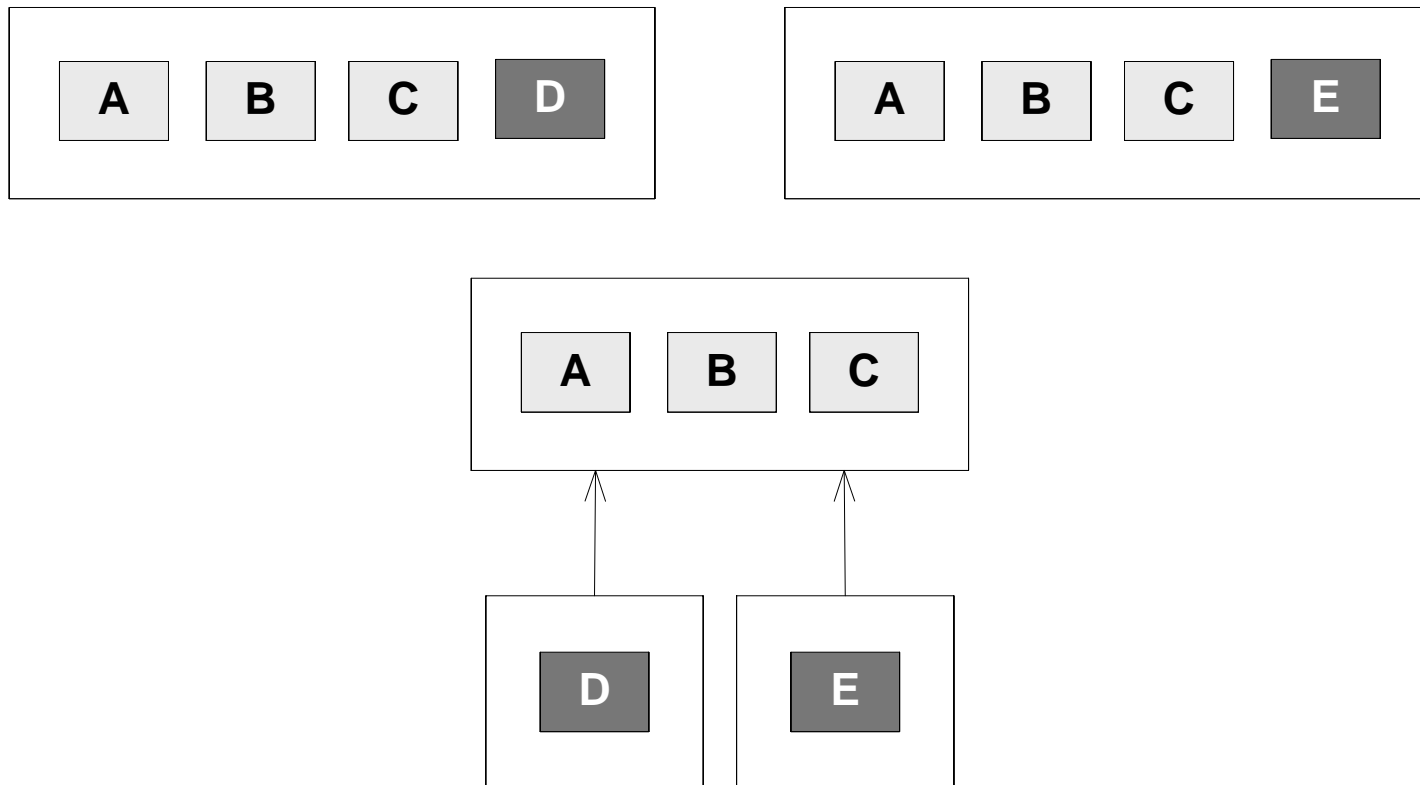


Factorização

Redundância

Uma das principais causas de anomalias no desenvolvimento de software

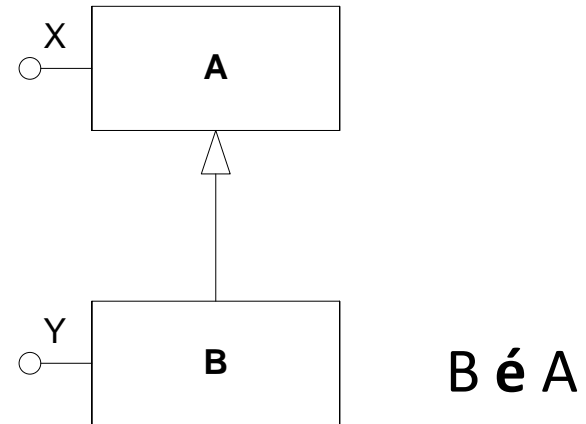
Redução de redundância por factorização



Mecanismos de Factorização

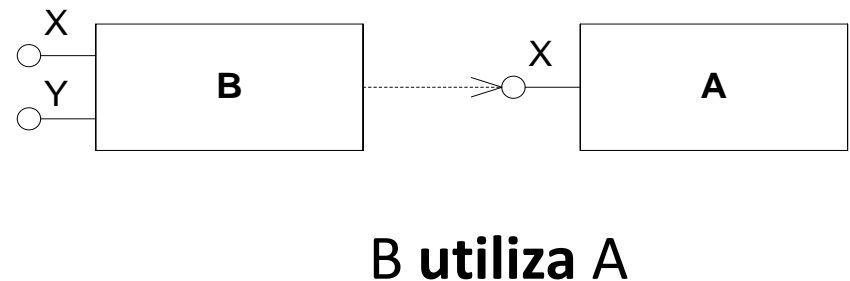
HERANÇA

- Nível de **acoplamento alto**
- B é A



DELEGAÇÃO

- Nível de **acoplamento baixo**
- B utiliza A
- Agregação de partes
- Acoplamento pode variar dinamicamente



Herança / Delegação

Redução de redundância em modelos estruturais

- **Herança**

- **Nível de acoplamento alto**
- **B é A**
- Ênfase na estrutura
- Arquitecturas orientadas a objectos

- **Delegação**

- **Nível de acoplamento baixo**
- **B utiliza A**
- Ênfase na funcionalidade
- Agregação de partes
- Acoplamento pode variar dinamicamente
 - Padrões de interacção podem ser determinados em tempo de execução
 - Arquitecturas orientadas a serviços

Bibliografia

[Pressman, 2003]

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

[Gamma et al., 1995]

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1995.

[Shaw & Garlan, 1996]

M. Shaw, D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice-Hall, 1996.

[Vernon, 2013]

V. Vernon, *Implementing Domain Driven Design*, Addison-Wesley, 2013.

[Parnas, 1972]

D. Parnas, *On the Criteria to Be Used in Decomposing Systems into Modules*, Communications of the ACM 15-12, 1968.

[Kruchten, 1995]

F. Kruchten, *Architectural Blueprints - The "4+1" View Model of Software Architecture*, IEEE Software, 12-6, 1995.

[Schach, 2010]

S. Schach, *Object-Oriented and Classical Software Engineering*, 8th Edition, McGraw-Hill, 2010.

[Booch, 2004]

G. Booch, *Software Architecture*, IBM, 2004.

[Korf, 1980]

R. Korf, *Toward a model of representation changes*, Artificial Intelligence, Volume 14, Issue 1, 1980.

[Laplante, 2007]

P. Laplante, *What Every Engineer Should Know About Software Engineering*, CRC Press, 2007.