Universidade do Minho
Departamento de Informática

ARQUITECTURAS DE SOFTWARE

2007-2008

F. Mário Martins

TÓPICOS :

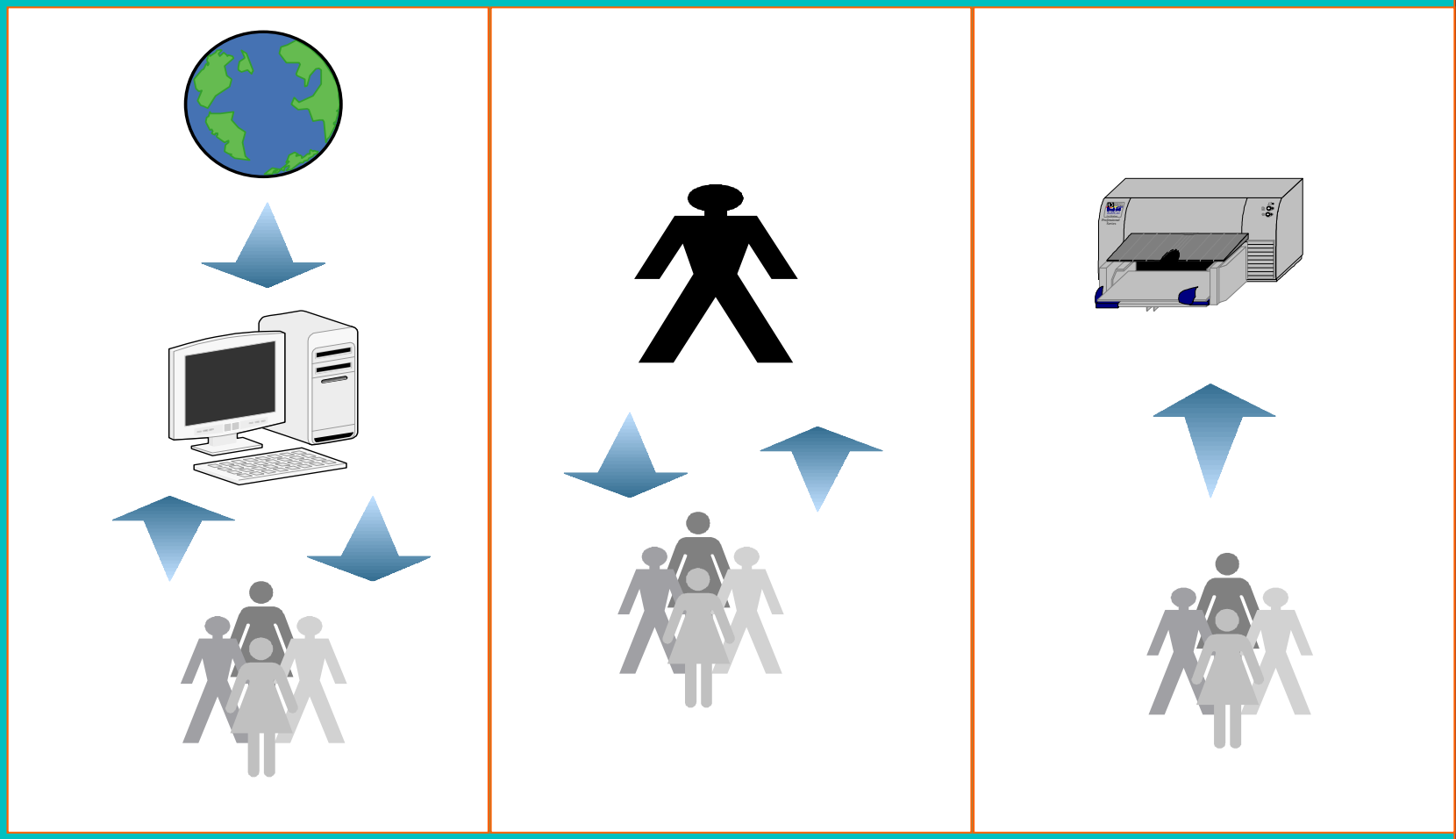
- Concepção de Sistemas Software: Fases, Processos e Métodos
- Model-Driven Software Engineering (MDSE)
- Model-Driven Architectures/Development (MDA/MDD)
- Architecture Description Languages: ADLs
- Modelação de Processos de Negócio (BPM)
- **Modelação Visual Orientada aos Objectos: UML**
- **OCL: Object Constraint Language**

FUNCIONAMENTO (1):

1	12.10.2007	Introduction: Main Concepts and Area Scope Information Systems; Software Engineering; Software Architectures; Component-based Software; Model-Driven SE; Model-Driven Architectures;
2	19.10.2007	Object Oriented Modeling (OMG, UML, OCL); Business Processes (BPM); Software Tools
3	26.10.2007	OO Software Modeling with UML Views: Logic, Structure and Behaviour Diagrams: classification
4	02.11.2007	Use Cases
5	09.11.2007	Class Diagrams and Class Relationships
6	16.11.2007	Interaction Diagrams (Collaboration/Sequence) Object Diagrams
7	23.11.2007	Statechart; Activity and Timming Diagrams
8	30.11.2007	Interfaces; Packages; Deployment
9	07.12.2007	Introduction to the OCL language Extending class diagrams with OCL
10	14.12.2007	OCL (cont.);
11	21.12.2007	Analysis of Small Case Studies
12	11.01.2008	**Overall assessment of produced reports**
13	18.01.2008	Stereotypes; Components Deployment Diagrams;
14	25.01.2008	Implementation Models Implementation Technologies
15	01.02.2008	Talk (to be defined)
16	08.02.2008	Talk (to de defined)
17	15.02.2008	**Software project delivery**
18	22.02.2008	**Final examination**

18 SEMANAS

FUNCIONAMENTO (2):



Leitura Prévia sobre dado tópico

Apresentação formal da matéria

Relatório de Síntese
(15 dias)

AVALIAÇÃO:

Nota: Grupos de no máximo 2 pessoas

TEÓRICA (40%):

4/5 relatórios de síntese (+- entre 2 a 3 semanas)

PRÁTICA (60%):

1 trabalho de modelação usando UML+OCL em ??

Normas:

Ambas as componentes de avaliação são obrigatórias e de nota mínima 10.
A componente teórica não tem recurso, apenas a componente prática.

O recurso consiste em completar o trabalho até à data do exame de recurso.

DESENVOLVIMENTO DE SW

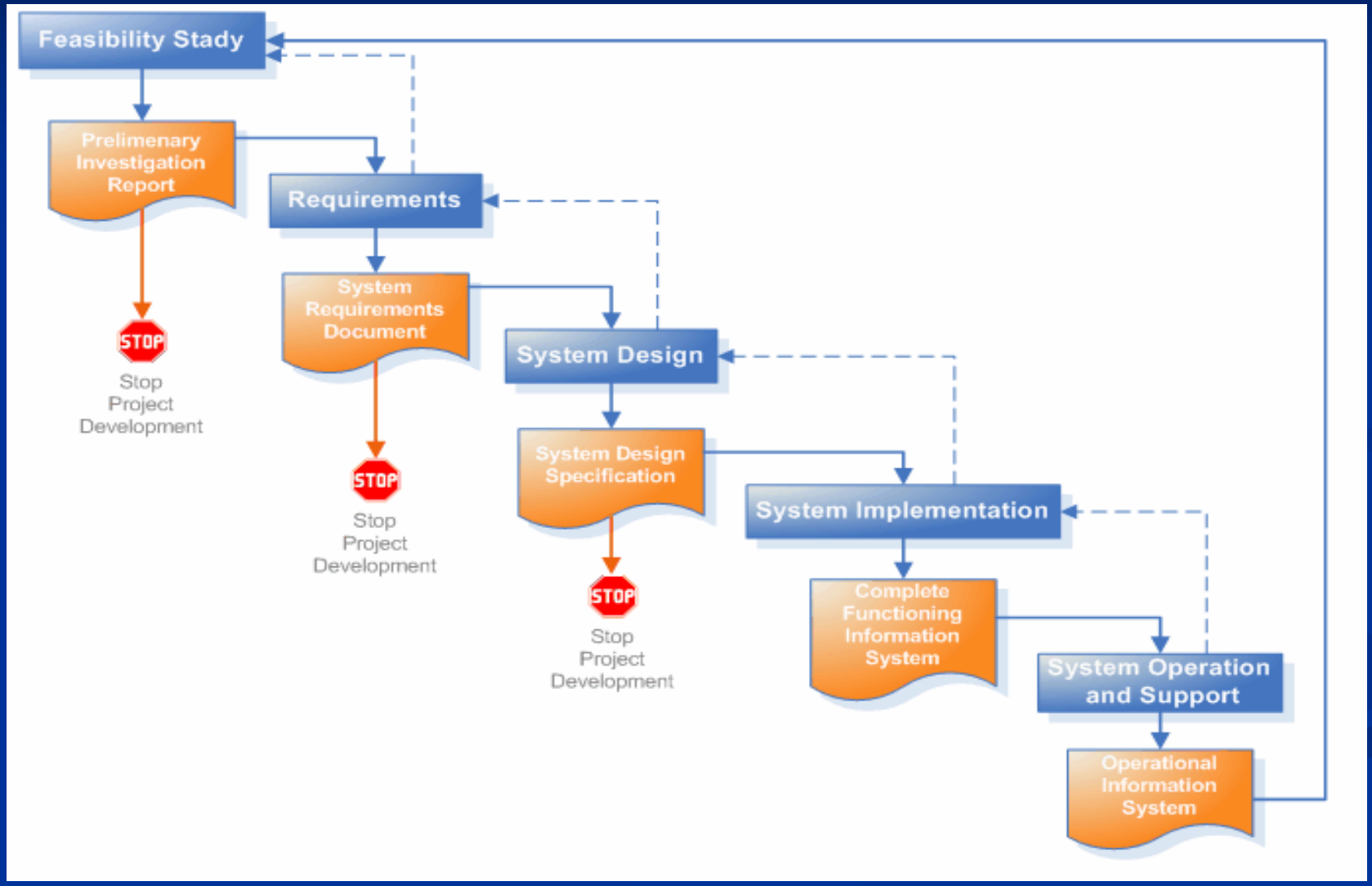
A **software development process** is a **structure** imposed on the development of a software product.

Synonyms include **software lifecycle** and **software process**.

There are several **models** for such processes, each describing approaches to a variety of **tasks or activities** that take place during the process.

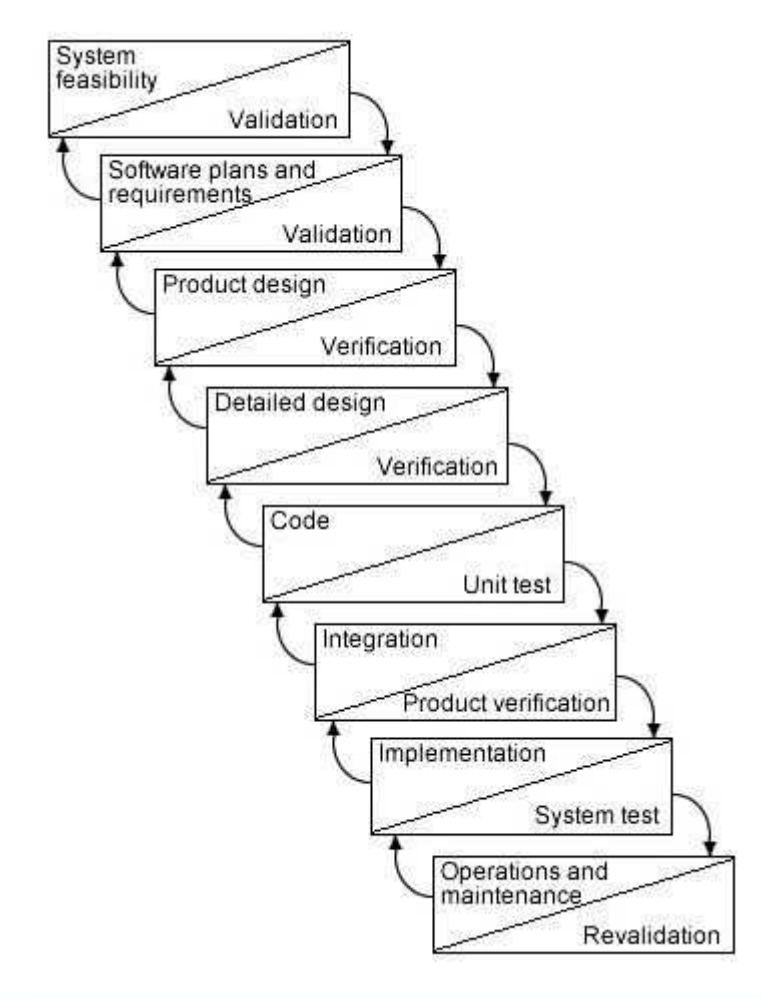
ETAPAS TÍPICAS EM ES

ETAPAS TÍPICAS EM ES

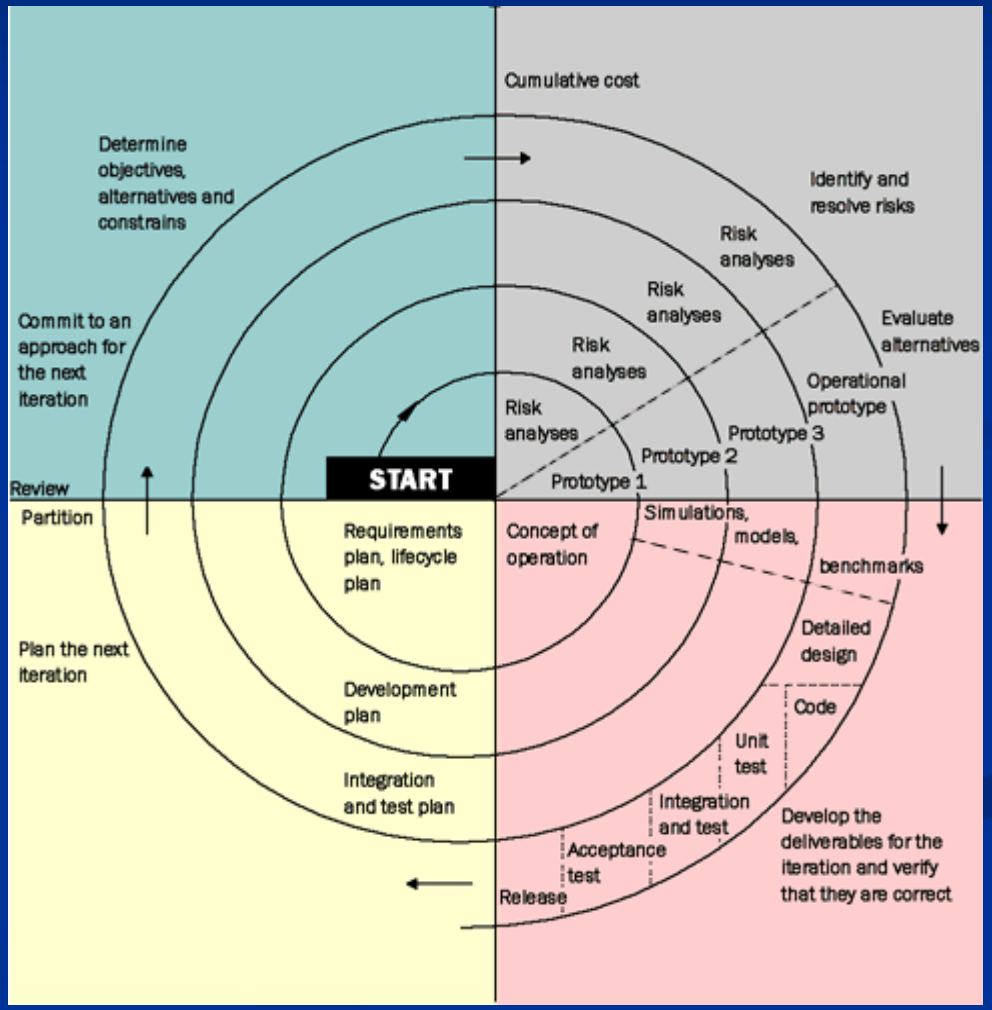


MODELOS DE PROCESSOS

Modelo Waterfall - Cascata

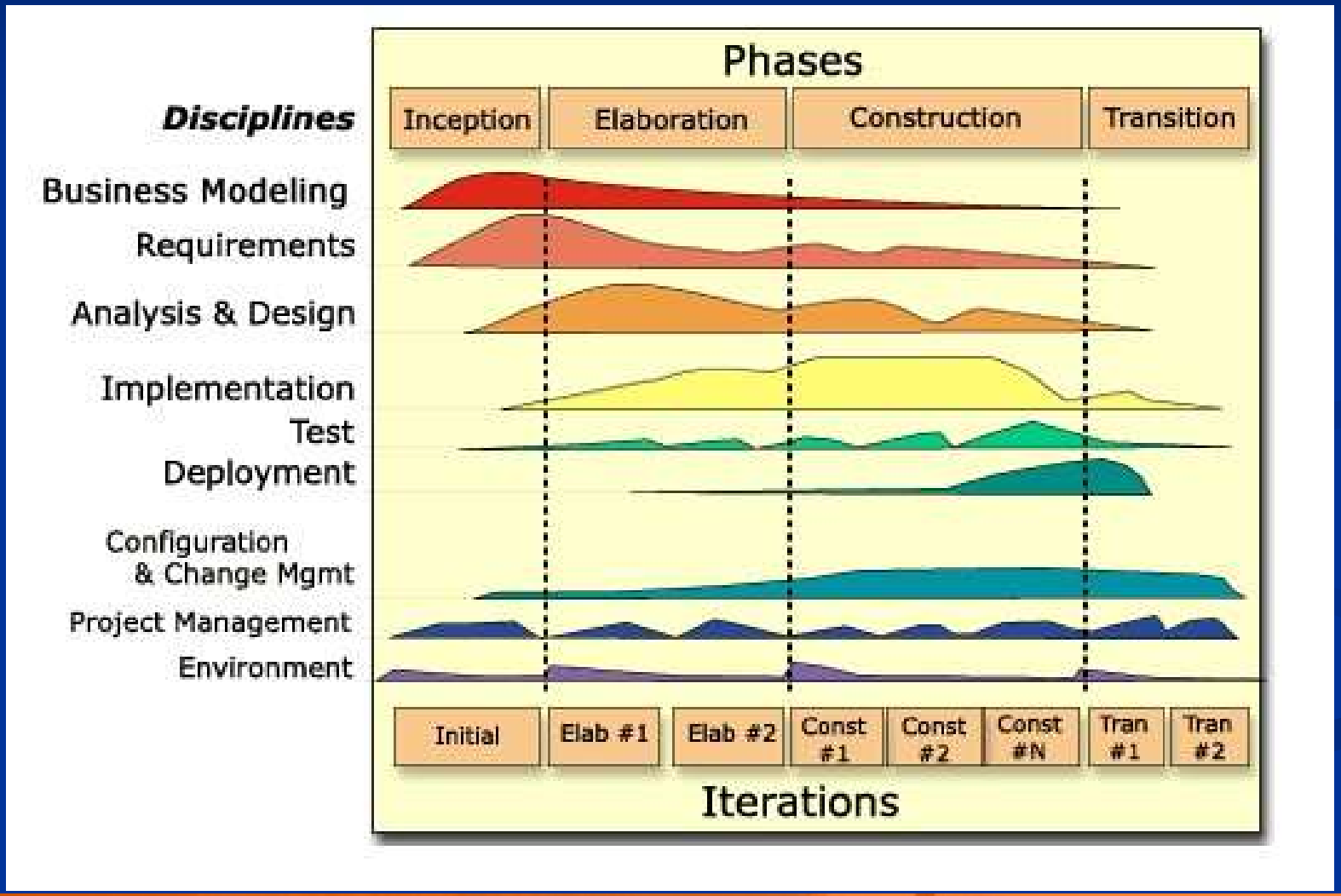


Modelo em Espiral



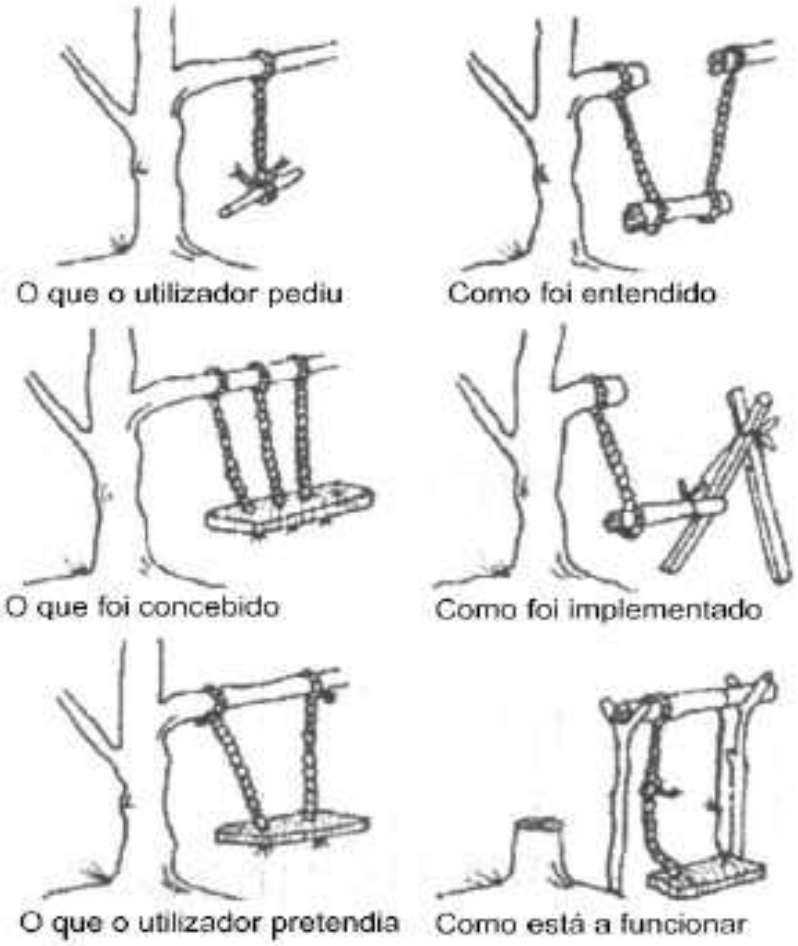
MODELOS DE PROCESSOS

Rational Unified Process



ETERNO PROBLEMA: REQUISITOS

Desenvolver um bom sistema não é tarefa trivial

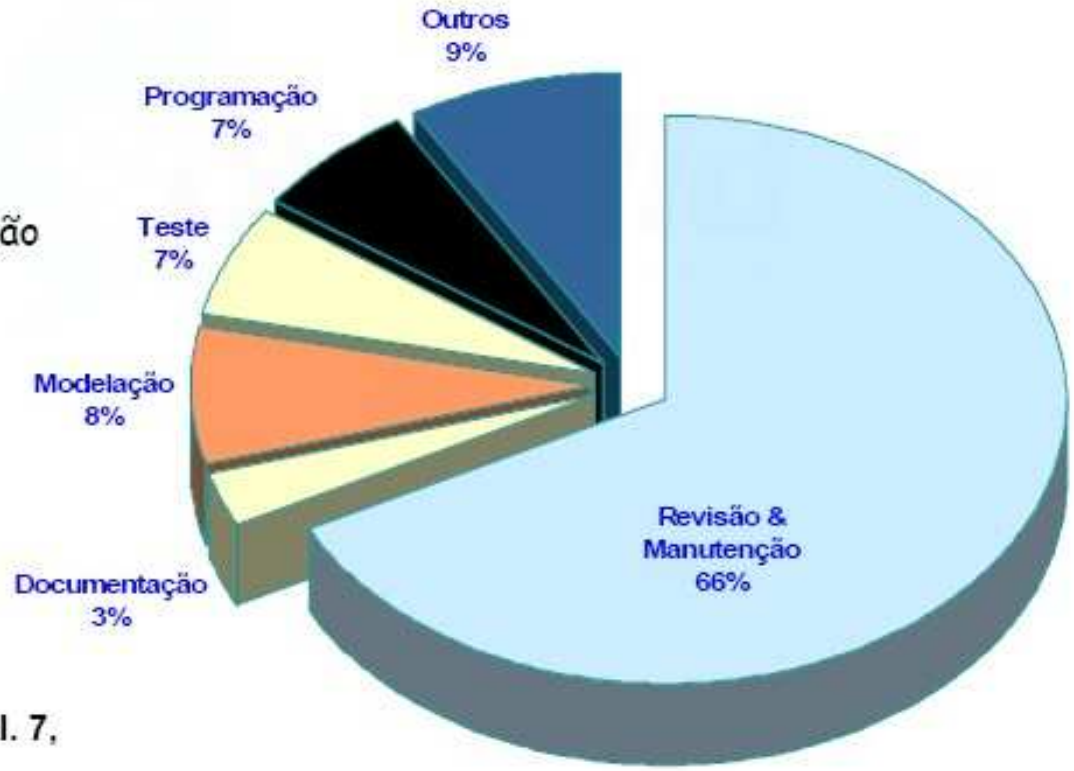


- Riscos associados aos requisitos.
- Riscos tecnológicos.
- Riscos de competência.
- Riscos políticos.

ETERNO PROBLEMA: REQUISITOS

os custos envolvidos num projecto de software

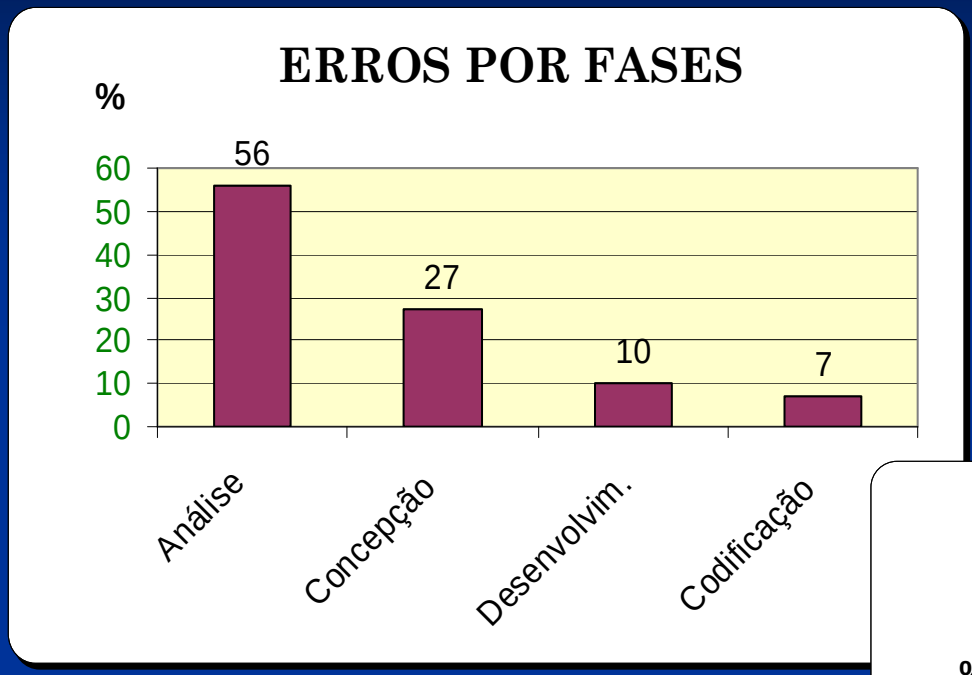
- Modelação
- Programação
- Teste
- Doumentação
- Revisão e Manutenção
- Outros

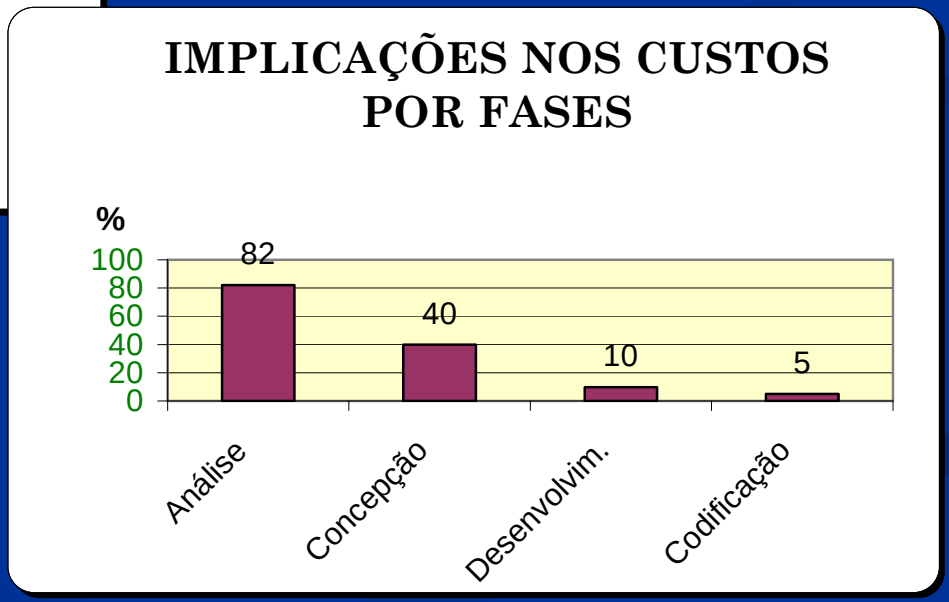


Source: DP Budget, Vol. 7, No. 12, Dec. 1998

cf. Teresa Galvão Dias (FEUP)

ETERNO PROBLEMA: REQUISITOS





ARQUITECTURAS DE SOFTWARE

Definition of software architecture

"An **architecture** is a set of significant decisions about the organization of a software system:
- the selection of the **structural elements** and
- their **interfaces** by which the system is composed,
- together with their behavior as specified in the collaborations among those elements,
- the **composition** of these structural and behavioral elements into progressively larger subsystems, and
- the **architectural style** that guides this organization, these elements and their interfaces, their collaborations, and their composition"

Kruchten: The Rational Unified Process (RUP).

ARQUITECTURAS DE SOFTWARE

ou, de outra forma:

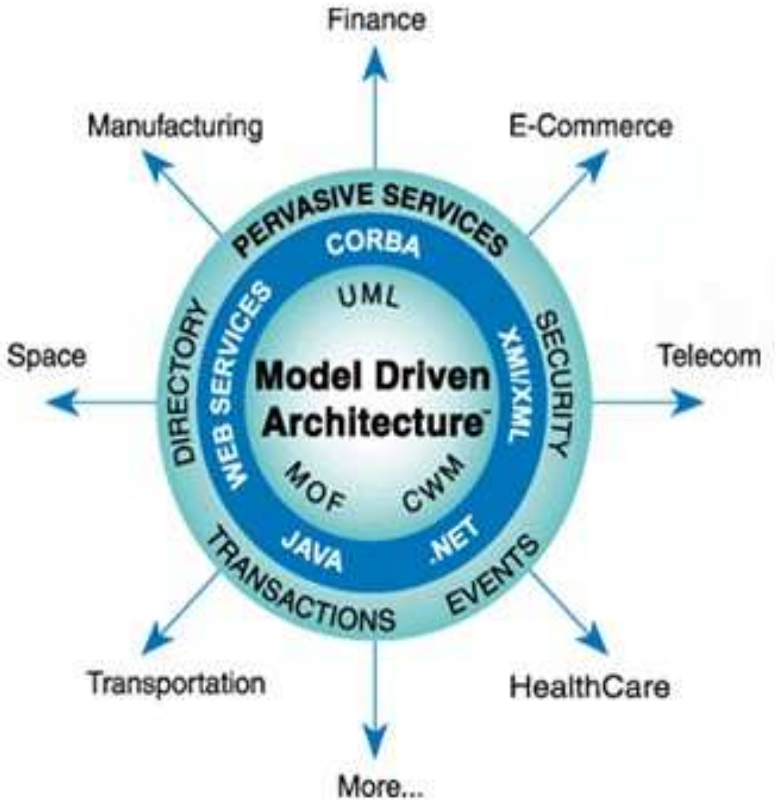
"The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them."

– from Software Architecture in Practice,
Bass, Clements, and Kazman

Comum: Componentes, Interfaces e Ligações

ARQUITECTURAS DE SOFTWARE

OMG Model Driven Architecture

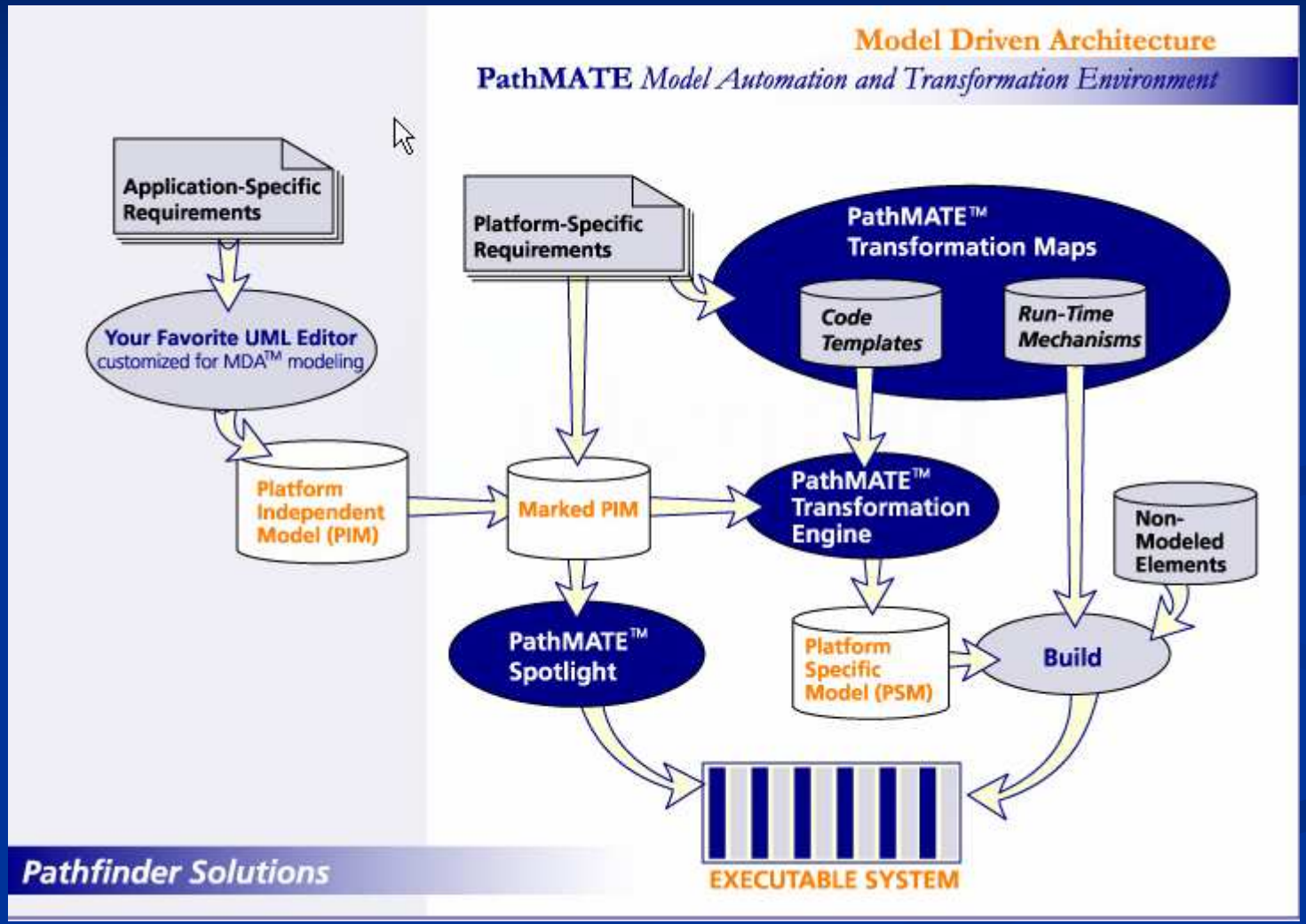


How Systems Will Be Built

OMG's Model Driven Architecture ® (MDA ®) provides an open, vendor-neutral approach to the challenge of business and technology change. Based on OMG's established standards, the MDA separates business and application logic from underlying platform technology. Platform-independent models of an application or integrated system's business functionality and behavior, built using UML and the other associated OMG modeling standards, can be realized through the MDA on virtually any platform, open or proprietary, including Web Services, .NET, CORBA ®, J2EE, and others. These platform-independent models document the business functionality and behavior of an application separate from the technology-specific code that implements it, insulating the core of the application from technology and its relentless churn cycle while enabling interoperability both within and across platform boundaries. No longer tied to each other, the business and technical aspects of an application or integrated system can each evolve at its own pace - business logic responding to business need, and technology taking advantage of new developments - as the business requires.

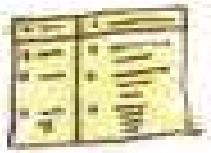
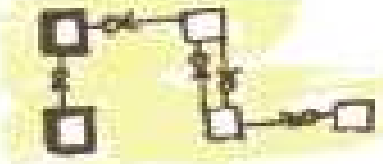
MODEL DRIVEN ARCHITECTURES (MDA)

P
I
M

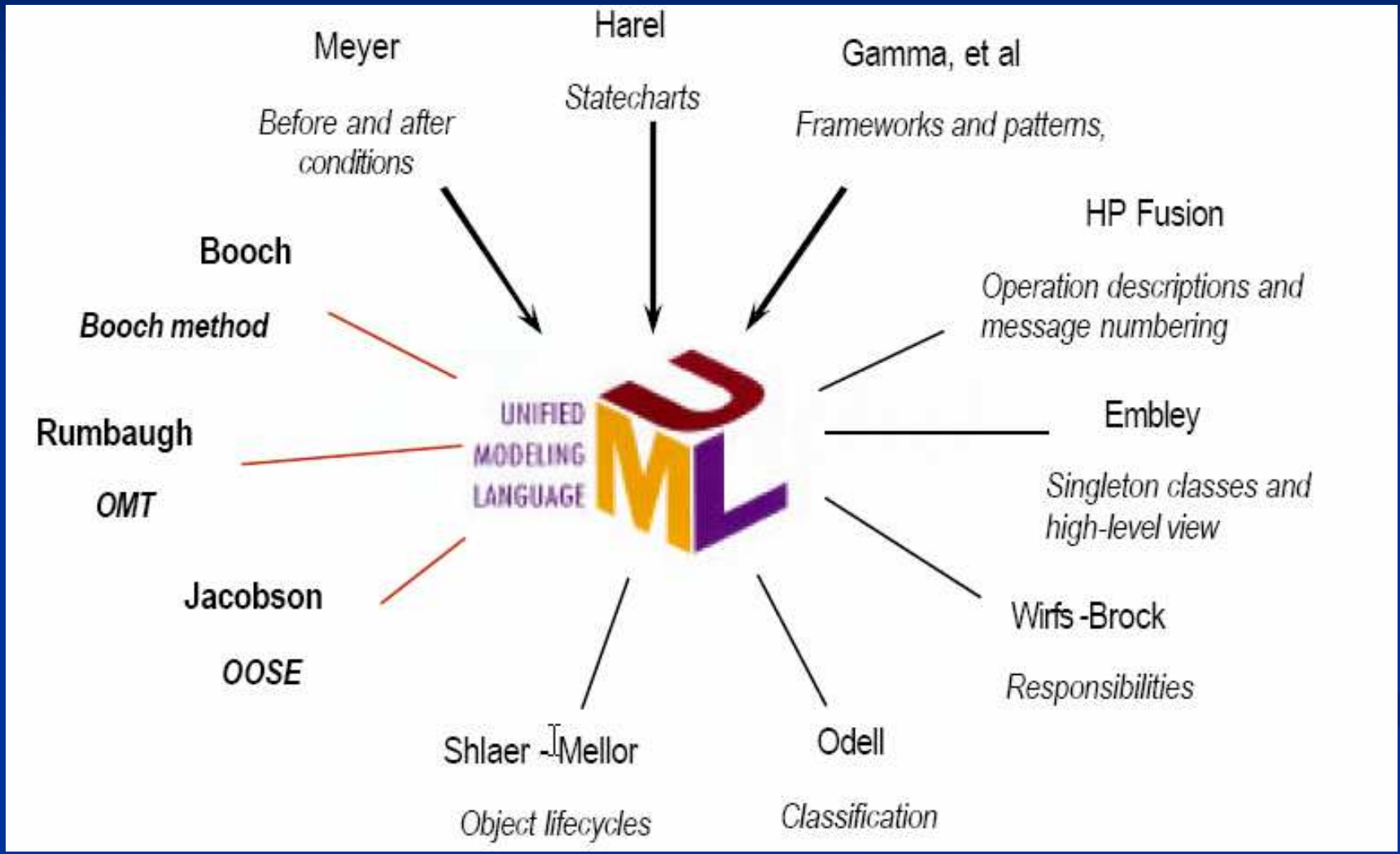


P
S
M

ARQUITECTURA DE SOFTWARE

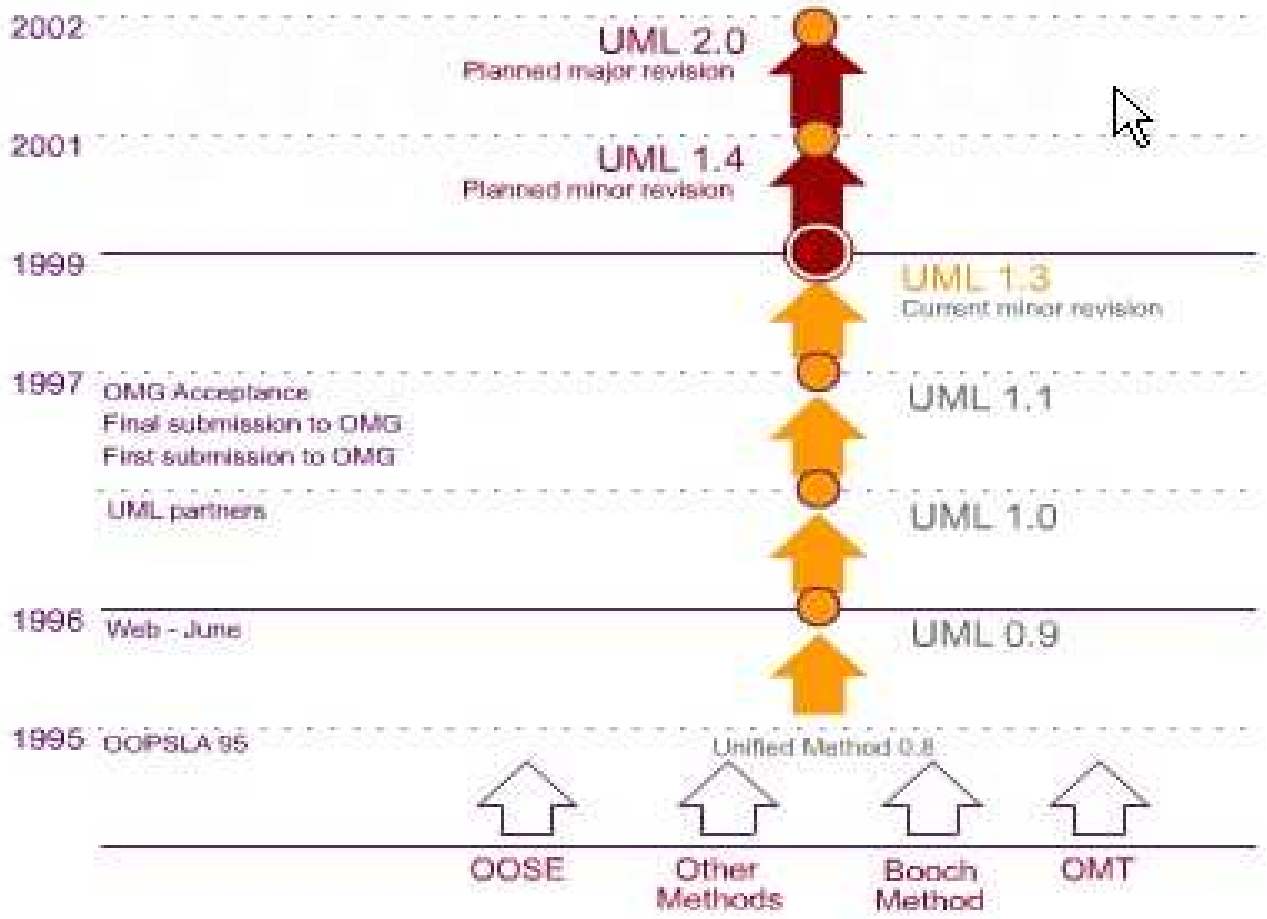
	Behavioral View	Structural View
Conceptual Architecture (abstract)	 Collaboration trace	 Architecture Diagram  Informal Component Specs (CRC-R)
Logical Architecture (detailed)	 Collaboration Diagrams	 Architecture Diagram with I/Fs  Interface Specs
Execution Architecture (Process View and Deployment View)	 Collaboration Diagrams showing processes	 Architecture Diagram showing Active Components

UML: ORIGENS


Meyer
Before and after conditions
Harel
Statecharts
Gamma, et al
Frameworks and patterns,
Booch
Booch method
HP Fusion
Operation descriptions and message numbering
Rumbaugh
OMT
UNIFIED MODELING LANGUAGE
UML
Embley
Singleton classes and high-level view
Jacobson
OOSE
Wirfs-Brock
Responsibilities
Shlaer-Mellor
Object lifecycles
Odell
Classification

UML: EVOLUÇÃO

Timeline



Diagramas de UML2

Diagramas de Estrutura

- Diagrama de Objectos
- Diagrama de Classes
- Diagrama de Componentes
- Diagrama de Instalação
- Diagrama de Pacotes
- Diagrama de Estrutura Composta

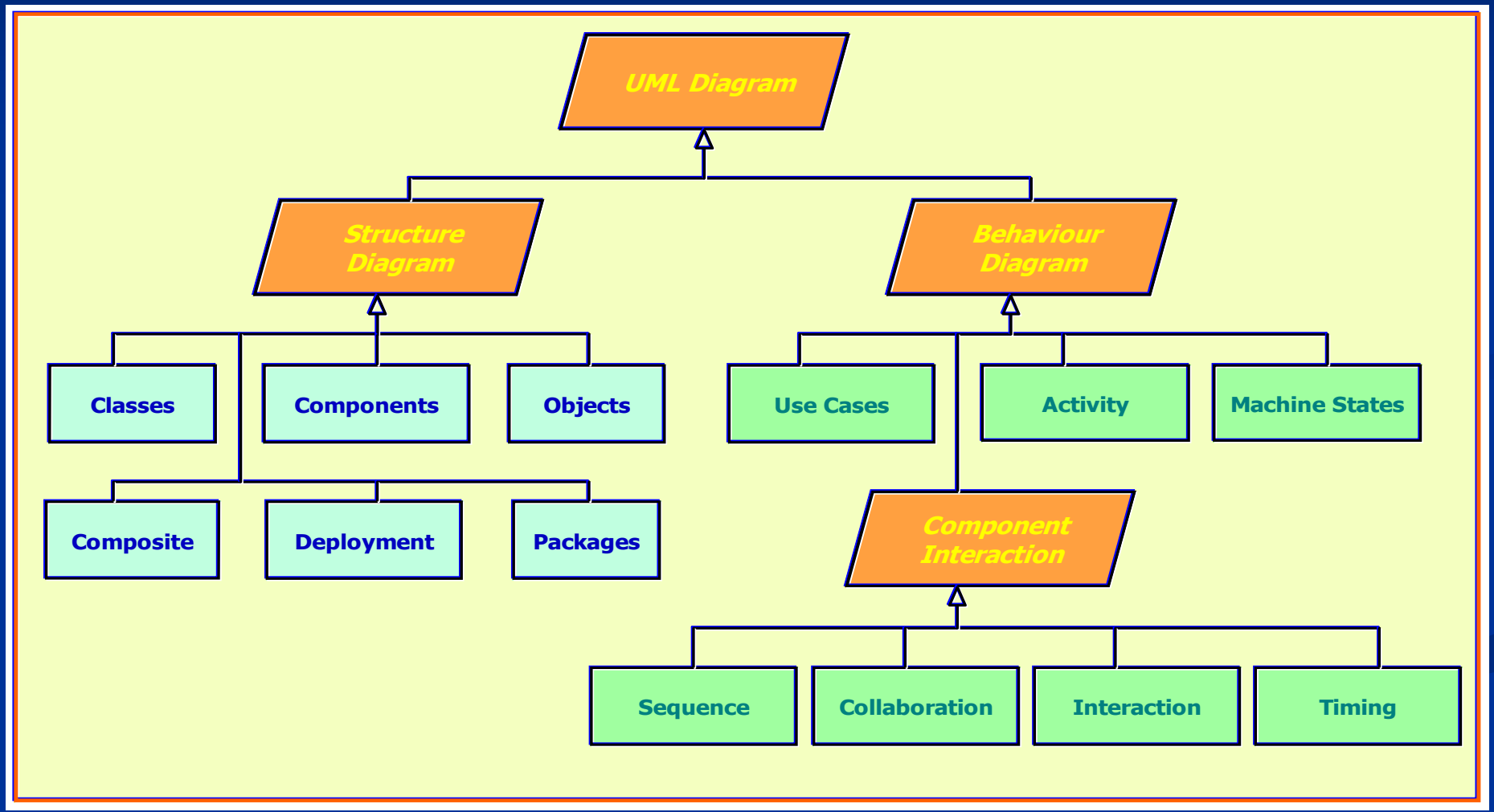
Diagramas de Comportamento

- Diagrama de Casos de Uso
- Diagrama de Actividade
- Diagrama de Estados

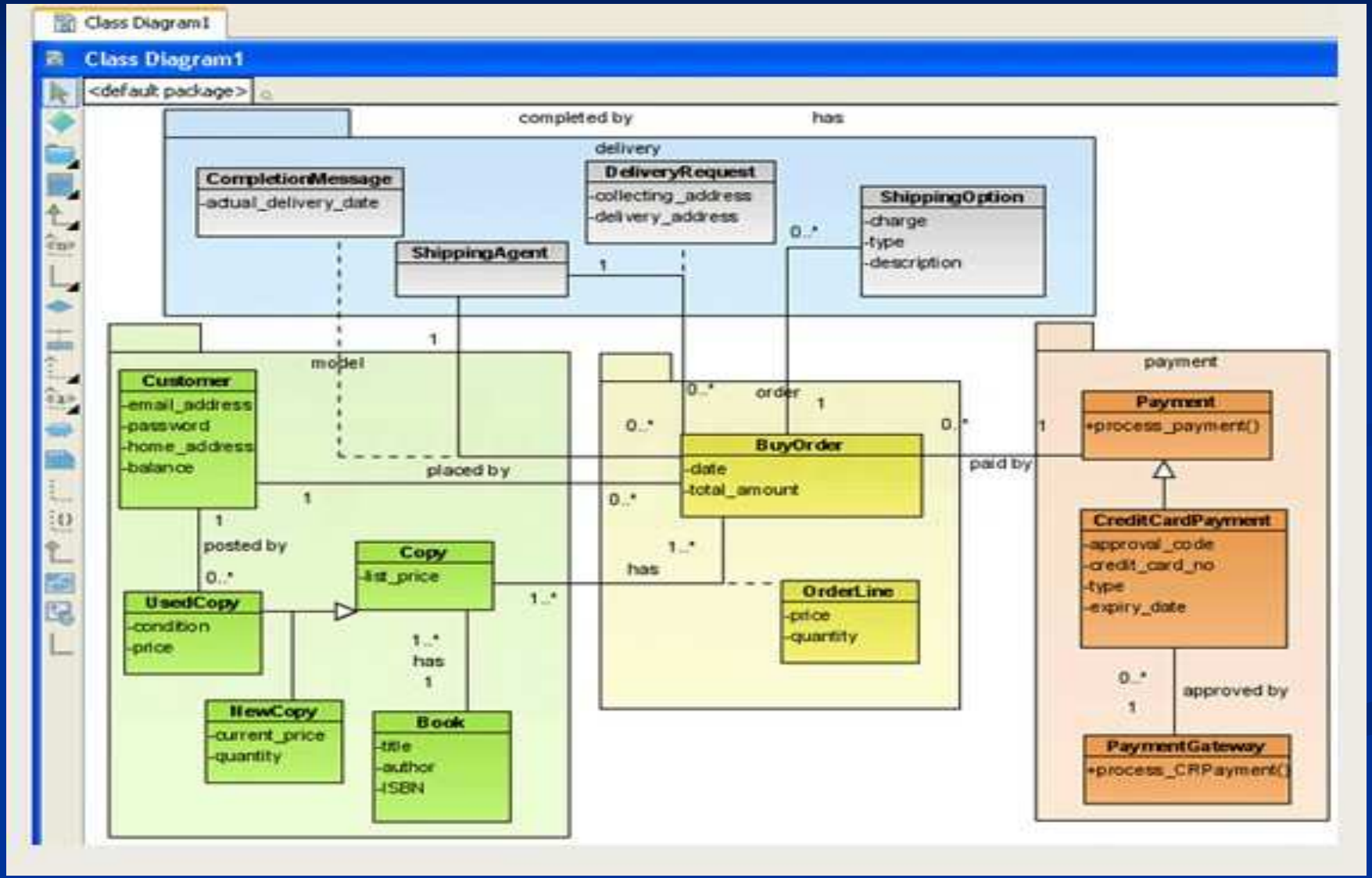
Diagramas de Interacção

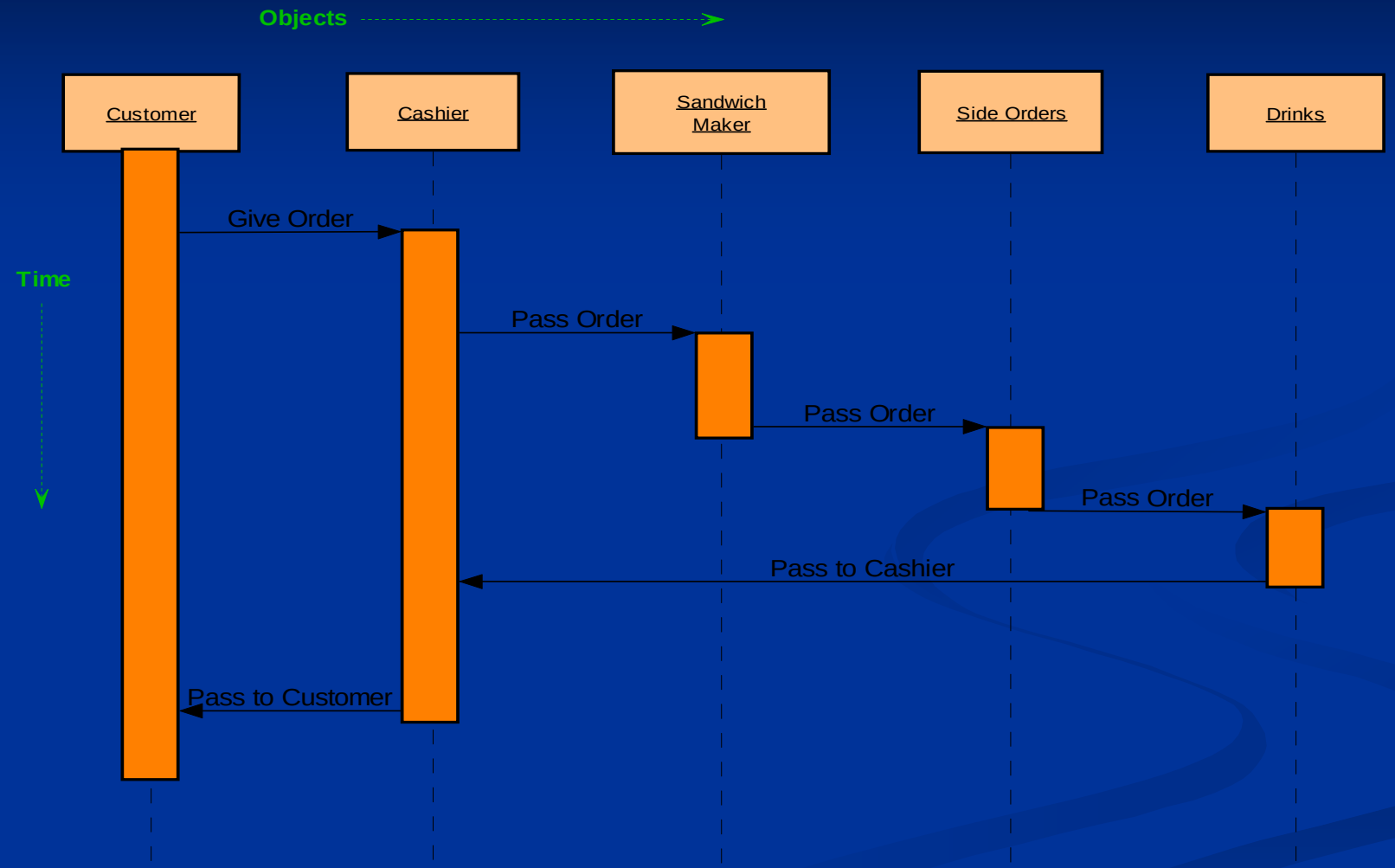
- Diagrama de Sequência
- Diagrama de Interacção
- Diagrama de Colaboração
- Diagrama de Temporização

Hierarquia dos Diagramas



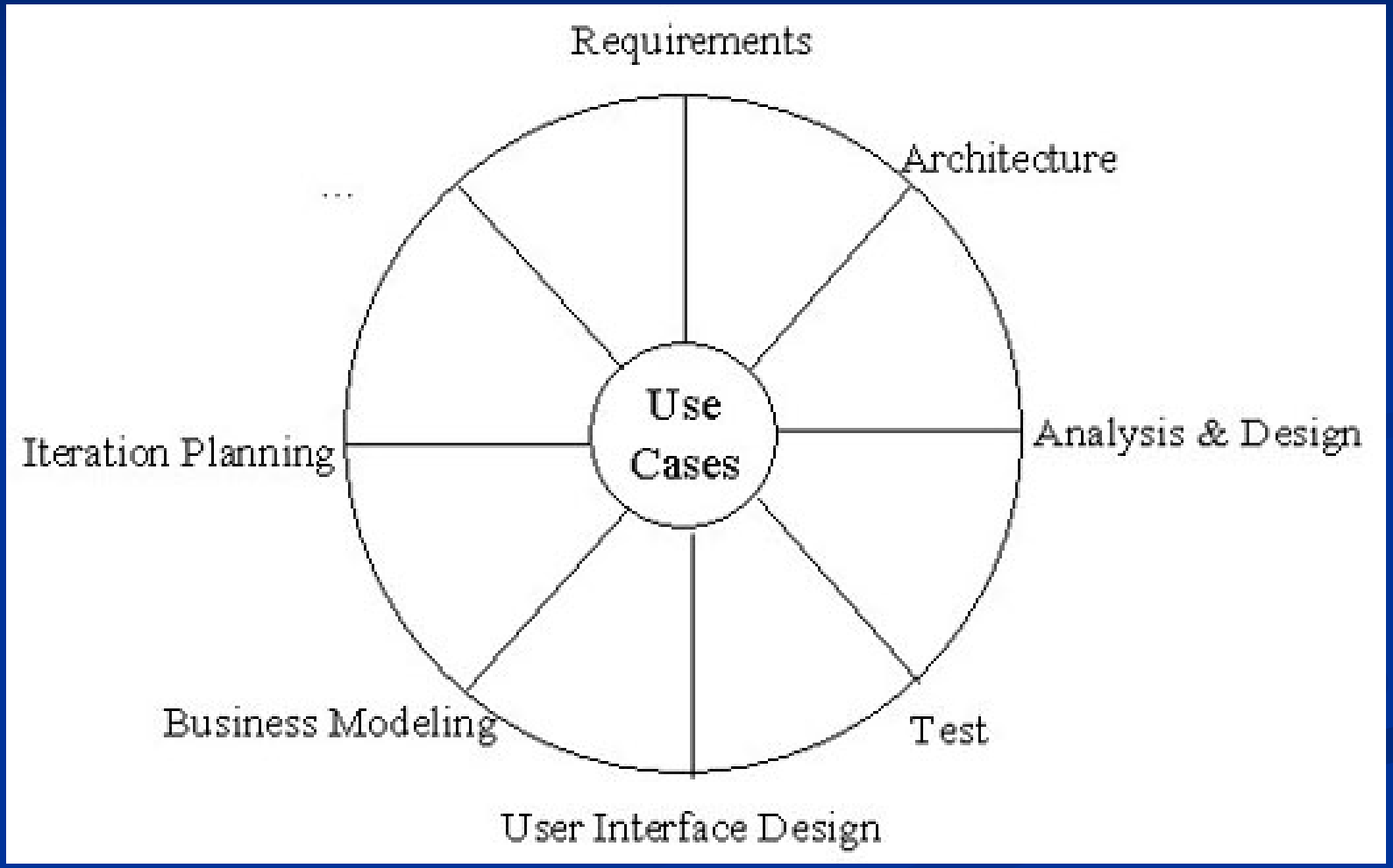
EXEMPLOS DE DIAGRAMAS

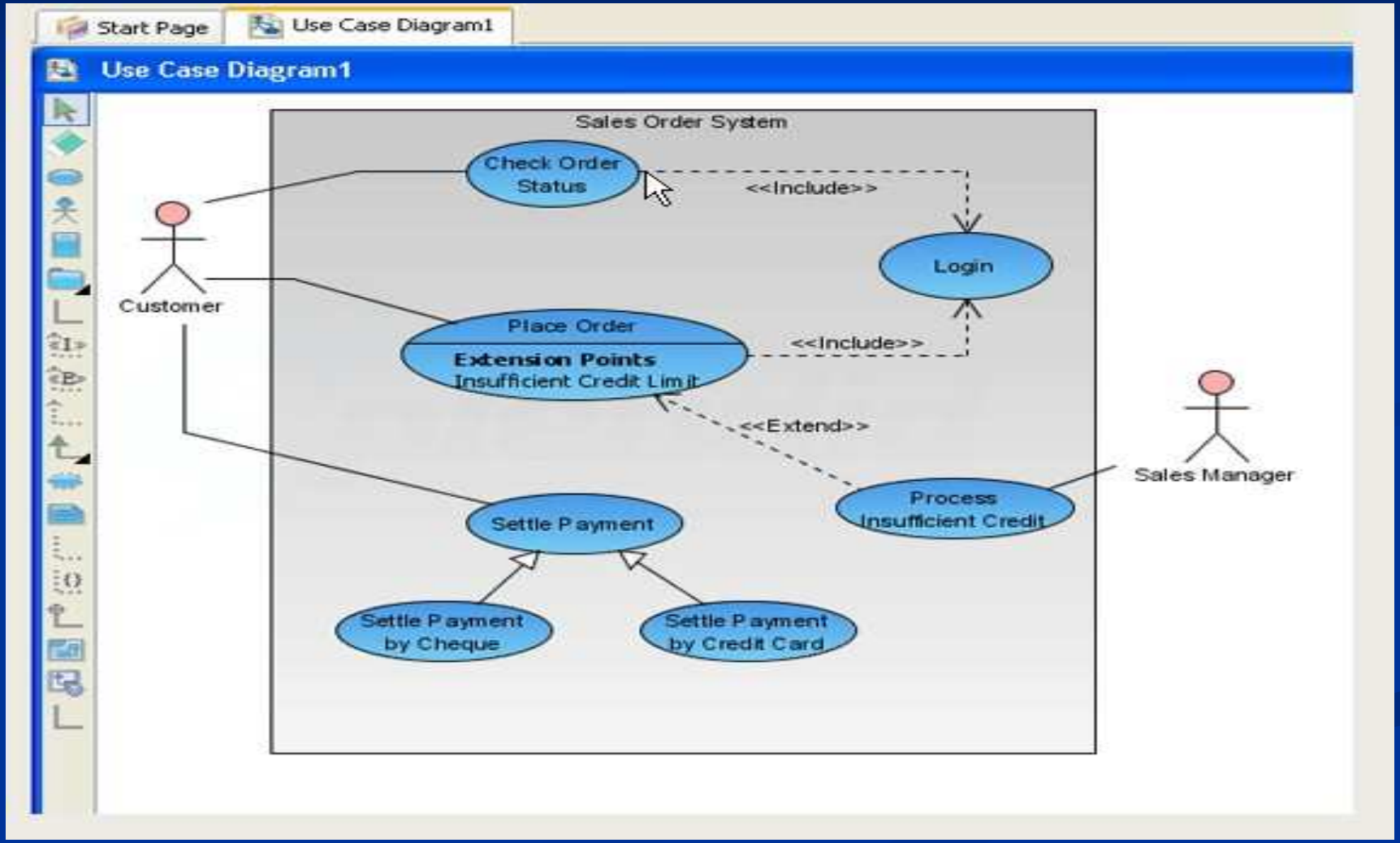




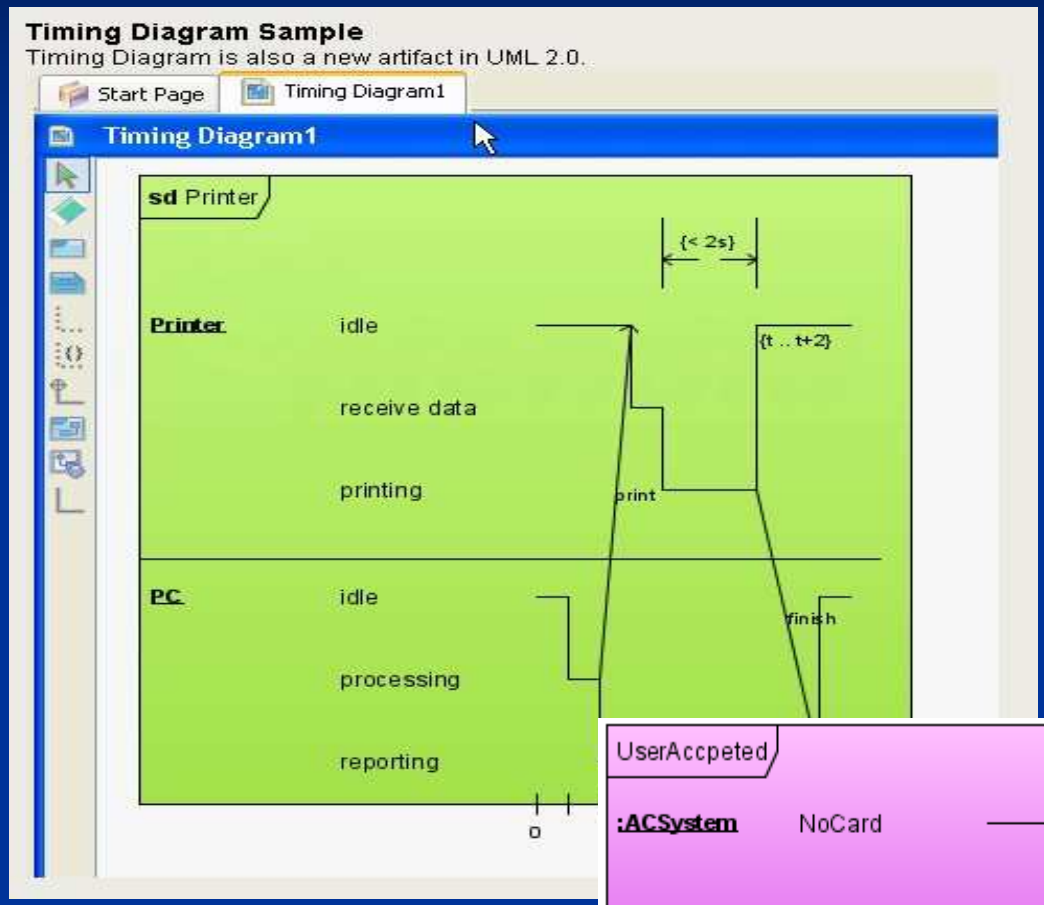
Sequence of Time: Order Preparation

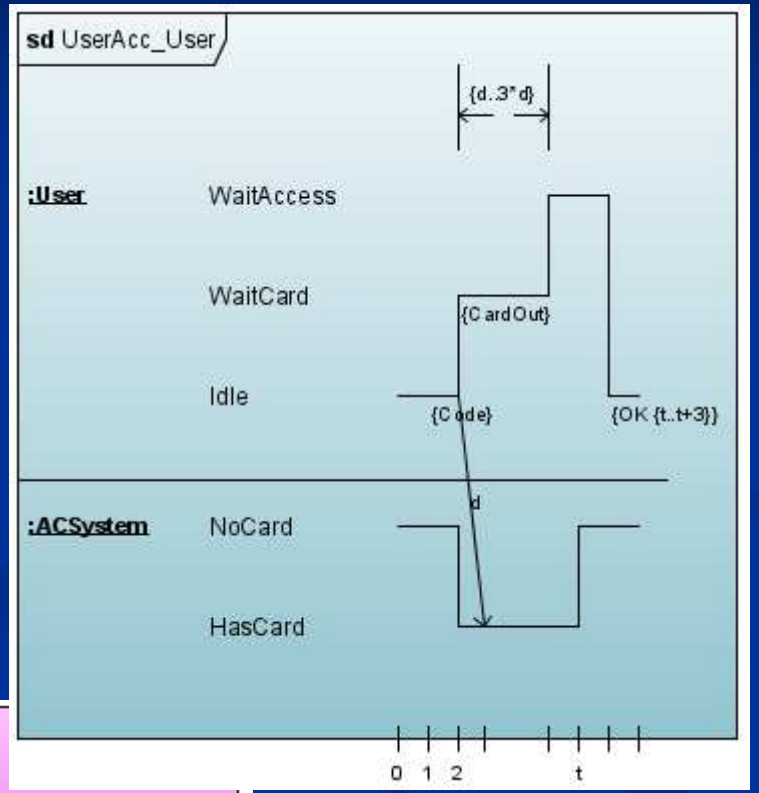
USE CASES

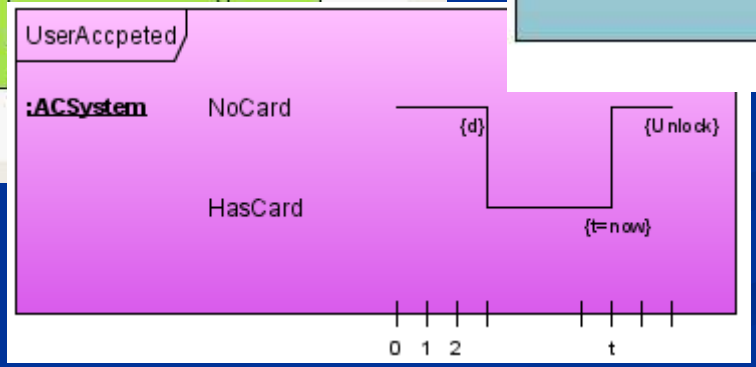




em www.visual-paradigm.com







em www.visual-paradigm.com

A maioria das ferramentas para modelação visual em UML permitem a geração automática de parte ou de todo o código final

Generate and Reverse 10+ languages

Visual Paradigm for UML supports a set of languages both in Code Generation and Reverse Engineering on Java, C++, CORBA IDL, PHP, XML Schema, Ada and Python. In addition, Code Generation supports C#, VB .NET, Object Definition Language (ODL), Flash ActionScript, Delphi, Perl, Objective-C, and Ruby. Reverse Engineering also supports Java class, .NET dll and exe, JDBC, and Hibernate mapping files.

em www.visual-paradigm.com

Hipóteses: Poseidon, VisualParadigm, Jude, Eclipse/UML, etc.

Referências (para já)

UML

The UML Resource Page: http://www.uml.org/, OMG.
The Unified Modeling Language User Guide, G. Booch, I. Jacobson, and J. Rumbaugh, Addison-Wesley, 1998.
Software Engineering Institute: http://www.sei.cmu.edu

OMG MDA: Contains white papers and presentations on OMG's MDA vision.
OMG UML Standard: Contains the UML standard and links to other sites on OO and UML modeling.
UML resource Center: Contains information on the UML including the official standard documents.
Alhir's Links to UML stuff : A source of information on the UML.
www.cetus-links.com
Links to WWW sites on OO frameworks, patterns, C++, and Java.
www.visual-paradigm.com http://visualcase.com
www.sparxsystems.com

Trabalho de Síntese 1 :

Investigação sobre Model Driven Development (MDD)

- O que é Model Driven Development ?
- Como surge e o que pretende resolver na Engenharia de Software ?
- Ferramentas actuais de MDD
- Perspectivas futuras em MDD
- Bibliografia consultada (e outra, mas actual)

- Máximo de 12 paginas A4 com tipo de letra de tamanho 11