TRANSFORMAÇÃO DIGITAL E O CICLO DE VIDA NO DESENVOLVIMENTO DE SISTEMAS

MODELAÇÃO E ANÁLISE DE SISTEMAS | TP

ILÍDIO OLIVEIRA ico@ua.pt
v2020-10-08

universidade de aveiro
departamento de eletrónica, telecomunicações e informática
deti

Objetivos de aprendizagem para esta aula

Descrever exemplos da transformação digital dos negócios

Identificar três eixos principais na transformação digital

Explique o que é o Ciclo de Vida do Desenvolvimento de Sistemas

Descreva as principais atividades dentro de cada fase do SDLC

Explique as principais características das ferramentas CASE

Justificar a utilização de modelos no desenvolvimento de software (benefícios)

Justifique as vantagens dos modelos visuais.

Distinguir dos modelos dinâmicos e estruturais (em UML).

A TRANSFORMAÇÃO DIGITAL E A VANTAGEM ESTRATÉGICA DOS SI

"SOFTWARE RUNS THE WORLD"

universidade de aveiro
departamento de eletrónica, telecomunicações e informática
deti

Technological change has never occurred as rapidly, or on as large a scale, as today.

"Technological innovation enables – indeed, requires – companies to boost their agility and thus their competitiveness. That's why CEOs' top priorities in 2016 should be to digitize the core components of their business and rethink organizational design and governance processes. Catching this fast-moving – and rapidly growing – "digital wave" is the only way to avoid getting left behind."

PROJECT SYNDICATE
THE WORLD'S OPINION PAGE

PRINT

DOMINIC BARTON
Dominic Barton is the global managing director of McKinsey & Company.

JAN 15, 2016

Catching the Digital Wave

NEW YORK – Technological change has always posed a challenge for companies. But, as we saw once again in 2015, it has never occurred as rapidly, or on as large a scale, as today. As innovation sweeps across virtually every sector, from heavy industry to services, it is transforming the competitive landscape, with the most advanced companies – rather than the largest or most established players – coming out on top.

For incumbents, the threat of displacement is very real. The average tenure of a company on the S&P 500 has fallen from 90 years in 1935 to less than 18 years today. Disruptive new players like Uber, which has upended the taxi industry, are tough competitors, often staking out market share by shifting more surplus to consumers. This is part of a broader trend of intensifying competition that, according to recent research from the McKinsey Global Institute, could reduce the global after-tax profit pool from almost 10% of global GDP today to its 1980 level of about 7.9% within a decade.

http://prosyn.org/lxXI6OW


caixadirecta
Saldos e movimentos
Transferências
Pagamentos
Produtos Caixa
caixadirecta
Caixadirecta
O SEU BANCO
SEMPRE DISPONIVEL.
Tudo o que precisa, onde estiver, como quiser.
Adesão gratuita.
SAIBA MAIS ›


APP CAIXADIRECTA
UMA EXPERIÊNCIA ÚNICA
DE ACESSO SIMPLES E CÓMODO
AO SEU BANCO.
JÁ DISPONÍVEL PARA TABLET E SMARTPHONE IOS E ANDROID.
INOVAR NA CAIXA. COM CERTEZA.
iOS


SERVIÇO CAIXAUTOMÁTICA
QUALQUER
ALTURA É BOA
PARA IR AO
CAIXAUTOMÁTICA.

Universidade de Aveiro Portal Académico Online PACO

Portal Académico

(3) Universidade de Aveiro : Portal Académico

Imprimir

Serviços Disponíveis

Página Principal

Alunos

Secretaria Virtual

Horários de Turmas do 1ºAno

Horários para 2014/2015

Creditações OnLine

Matrículas OnLine

Candidaturas

Candidaturas M23

Candidaturas Especiais

Candidaturas CET

Candidatura Cursos Livres

Candidaturas EI/internationalstudent

Docentes

Disciplinas

Secretaria Virtual para Estudantes da UA

Sistema de apoio aos **estudantes**, servindo de extensão à Secretaria dos Serviços Académicos.

SGQ

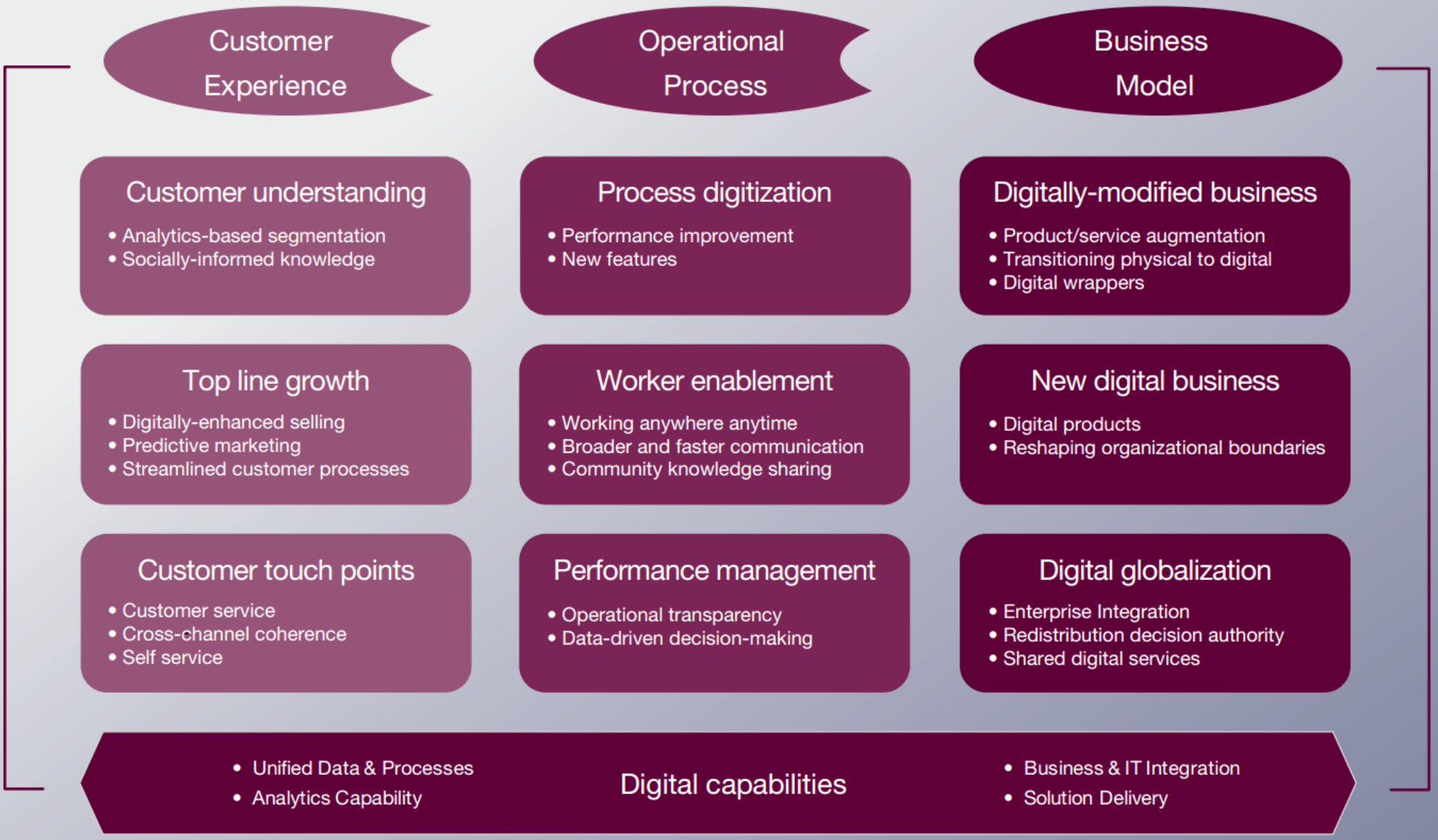
A partir de 26 de Janeiro, a Universidade de Aveiro (UA) implementa o Subsistema para a Garantia da Qualidade das Unidades Curriculares relativo ao 1º semestre do ano letivo 2014/2015.

A partir dessa data e até ao dia 22 de fevereiro, a UA promoverá o lançamento dos inquéritos pedagógicos junto dos estudantes. Os inquéritos são preenchidos eletronicamente, via PACO (http://paco.ua.pt/) ou diretamente em http://sgq.ua.pt.

Participa! A tua opinião é fundamental!


UBER

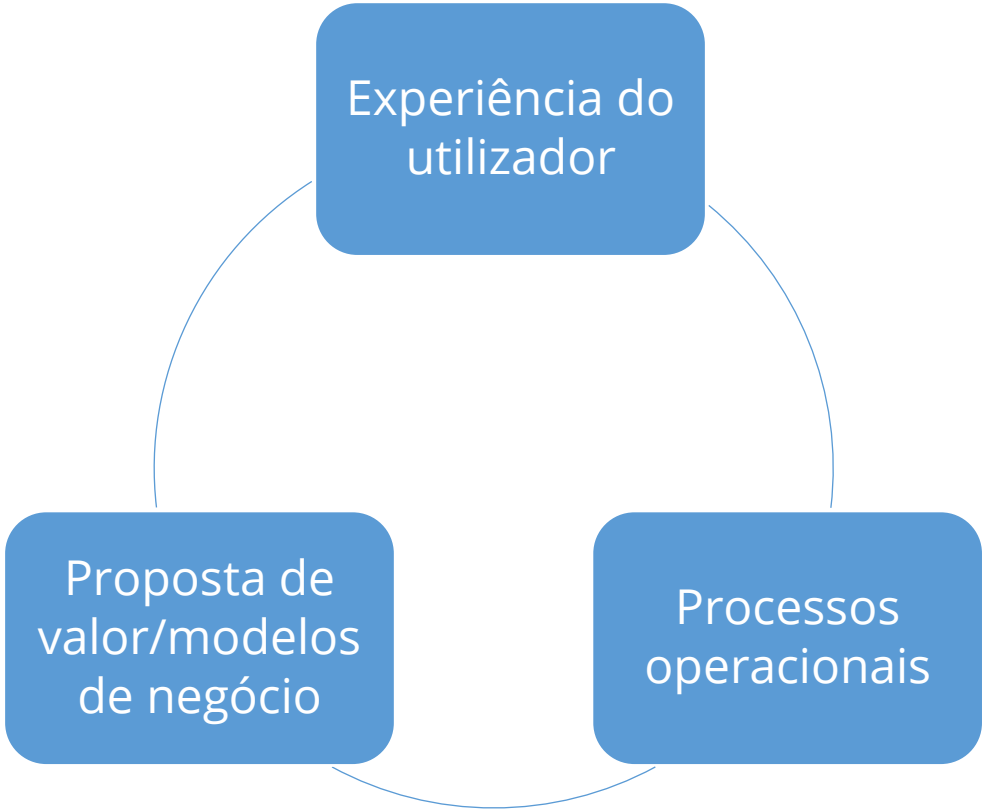
Figure 3: Building blocks of the digital transformation



"Digital Transformation: A Roadmap for Billion-Dollar Organizations", MIT Center for Digital Business

Transformação digital

A utilização de TIC para melhorar de forma decisiva o desempenho ou proposta de valor de uma empresa



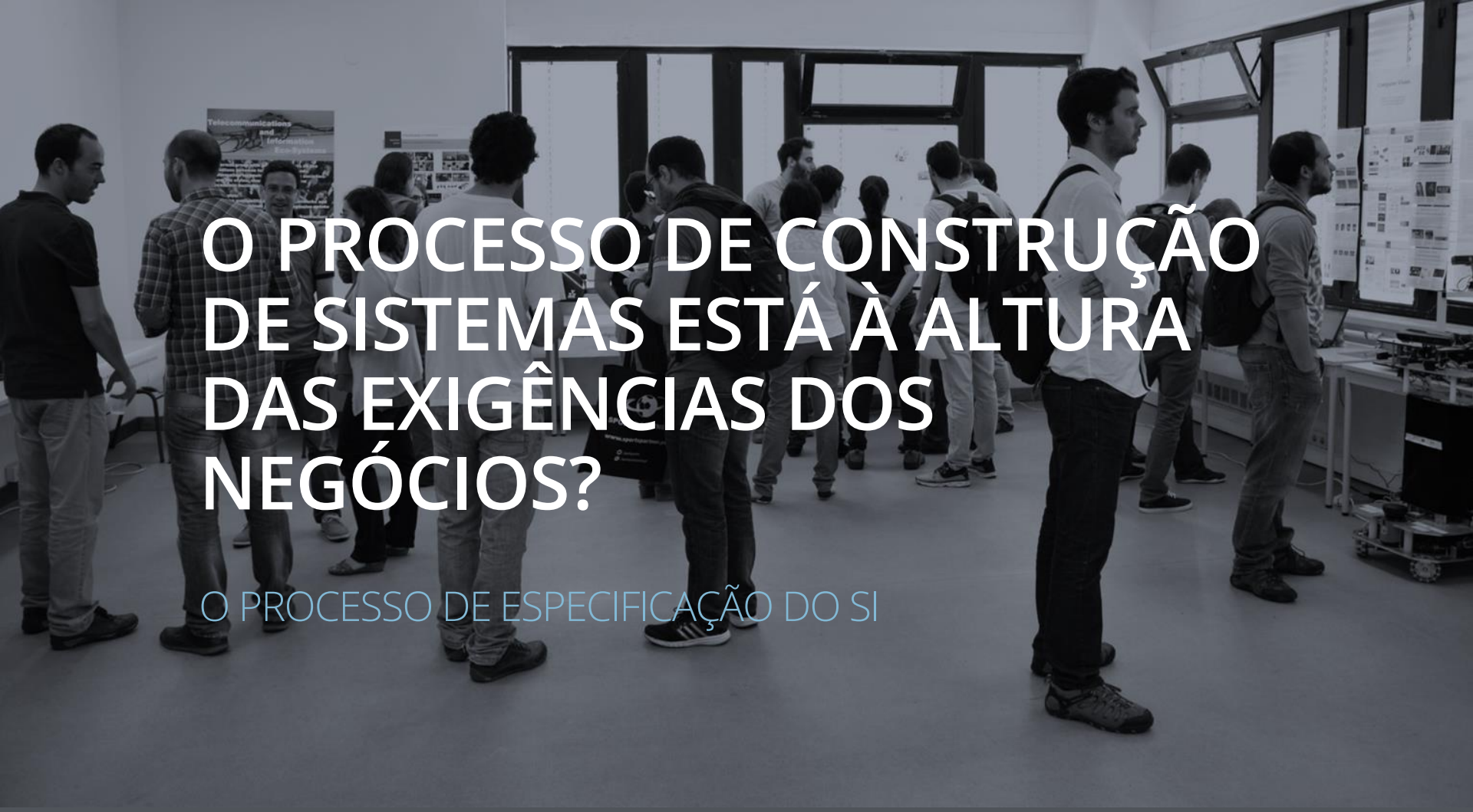
Tecnologias para a próxima década...

A hora da Europa: resposta à COVID-19

Prioridades:

- 4.1 Pacto Ecológico europeu
- 4.2 Transformação Digital
- 4.3 Solidariedade

https://ec.europa.eu/info/live-work-travel-eu/health/coronavirus-response/recovery-plan-europe_pt#documents



O PROCESSO DE CONSTRUÇÃO DE SISTEMAS ESTÁ À ALTURA DAS EXIGÊNCIAS DOS NEGÓCIOS?

O PROCESSO DE ESPECIFICAÇÃO DO SI

universidade de aveiro
departamento de eletrónica, telecomunicações e informática
deti

Integrated Requirements Engineering: A Tutorial

Ian Sommerville, *Lancaster University*

Before developing any system, you must understand
tem is supposed to do and how its use can suppo
the individuals or business that will pay for that sy
volves understanding the application domain (te
tions, railways, retail banking, games, and so on); the systen
constraints; the specific functionality required by the stakeho
ple who directly or indirectly use the system or the informatic

and essential system characteristics such as performance, security, and dependability. *Requirements engineering* is the name given to a structured set of activities that help develop this understanding and that document the system specification for the stakeholders and engineers involved in the system development.

This short tutorial introduces the fundamental activities of RE and discusses how it has evolved as part of the software engineering process. However, rather than focus on established RE techniques, I discuss how the changing nature of software engineering has led to new challenges for RE. I then introduce a number of new techniques that help meet these challenges by integrating RE more closely with other systems implementation activities.

This tutorial introduces the fundamental activities of requirements engineering and discusses recent developments that integrate it and system implementation.

The fundamental proc

The RE process varies i
ing on the type of applica
oped, the size and culture of
volved, and the software ac
used. For large military a
tems, there is normally a f
the systems engineering pro
tensively documented set o
ware requirements. For sm
veloping innovative softwar
process might consist of b
sions, and the product "req
simply be a short vision stat
software is expected to do.

Whatever the actual proc
tivities are fundamental to all RE processes:

- *Elicitation*. Identify sources of information about the system and discover the requirements from these.
- *Analysis*. Understand the requirements, their overlaps, and their conflicts.
- *Validation*. Go back to the system stake-

0740-7459/05/$20.00 © 2005 IEEE

- *The need for rapid software delivery.* Businesses now operate in an environment that's changing incredibly quickly. New products appear and disappear, regulations change, businesses merge and restructure, competitors change strategy. New software must be rapidly conceived, implemented, and delivered. There isn't time for a prolonged RE process. Development gets going as soon as a vision for the software is available, and the requirements emerge and are clarified during the development process.

Como é que compara com outras "engenharias"?

In 2015

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

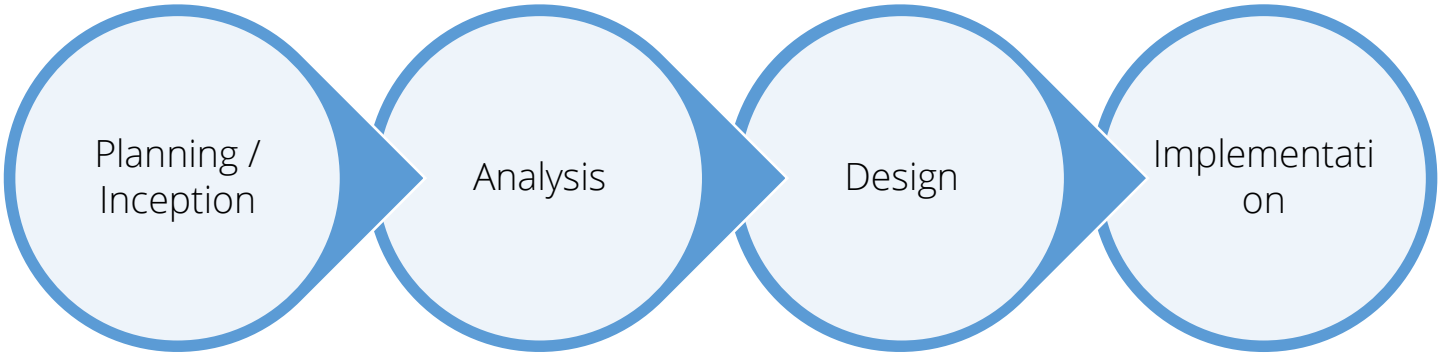
The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

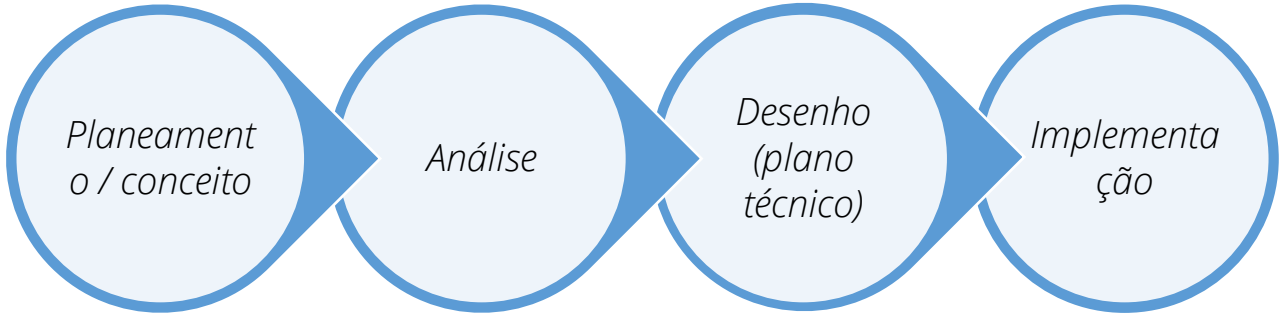
https://www.infoq.com/articles/standish-chaos-2015

SDLC phases

Quatro fases fundamentais: planeamento, análise, desenho e implementação. Diferentes projetos podem enfatizar diferentes partes do SDLC ou abordar as fases SDLC de diferentes maneiras, mas todos os projetos têm elementos destas quatro fases.

Cada fase é composta por uma série de etapas, conforme as técnicas que vão gerar as entregas necessárias.





Fundamental phases: **planning**, analysis, design, and implementation

The planning phase is the fundamental process of **understanding why an information system should be built** and determining how the project team will go about building it. Key steps:

Project initiation

the business value of the system to the organization is identified.

Most ideas for new systems come from outside the IS area (e.g., from the marketing department, accounting department).

A system request presents a brief summary of a business need, and it explains how a system that supports the need will create business value.

The system requests and feasibility analysis are presented to an information systems approval committee (sometimes called a steering committee), which decides whether the project should be undertaken.

Project management

The project manager creates a workplan, staffs the project, and puts techniques in place for the team to control and direct the project through the entire SDLC.

Fundamental phases: planning, **analysis**, design, and implementation

The analysis phase answers the questions of who will use the system, **what the system will do**, and where and when it will be used.

During this phase, the project team investigates any current system(s), **identifies opportunities for improvement, and develops a concept** for the new system.

Key steps:

analysis of existing systems,
requirements gathering,
solution concept (system proposal)

Fundamental phases: planning, analysis, **design**, and implementation

The design phase **decides how the system will be constructed**, in terms of the hardware, software, and network infrastructure; the user interface, forms, and reports; and the specific programs, databases, and files that will be needed.

Key steps:

system architecture design

data model design

program design

selection of frameworks

Fundamental phases: planning, analysis, design, and **implementation**

In the implementation phase, the system is **actually built** (or purchased, in the case of a packaged software design).

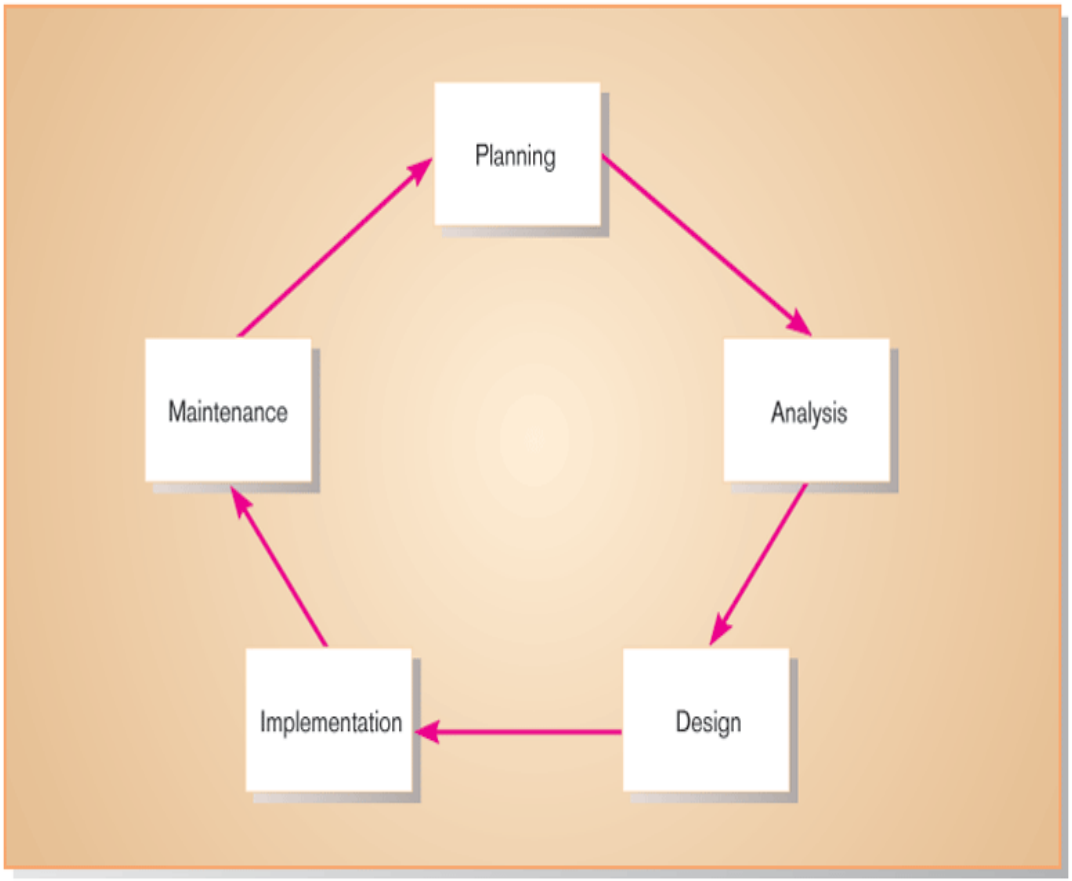
Includes also the transition to the production environment.

Key steps:

1. System construction (build and Quality Assurance)
2. Installation and transition
3. Support plan (post-install review and change management)

Alguns autores representarão a fase de manutenção

Figure 1-3 The systems development life cycle



Credit: Hoffer et al, "Modern Systems Analysis and Design", 5th ed.

Table 1-2 Products of SDLC Phases

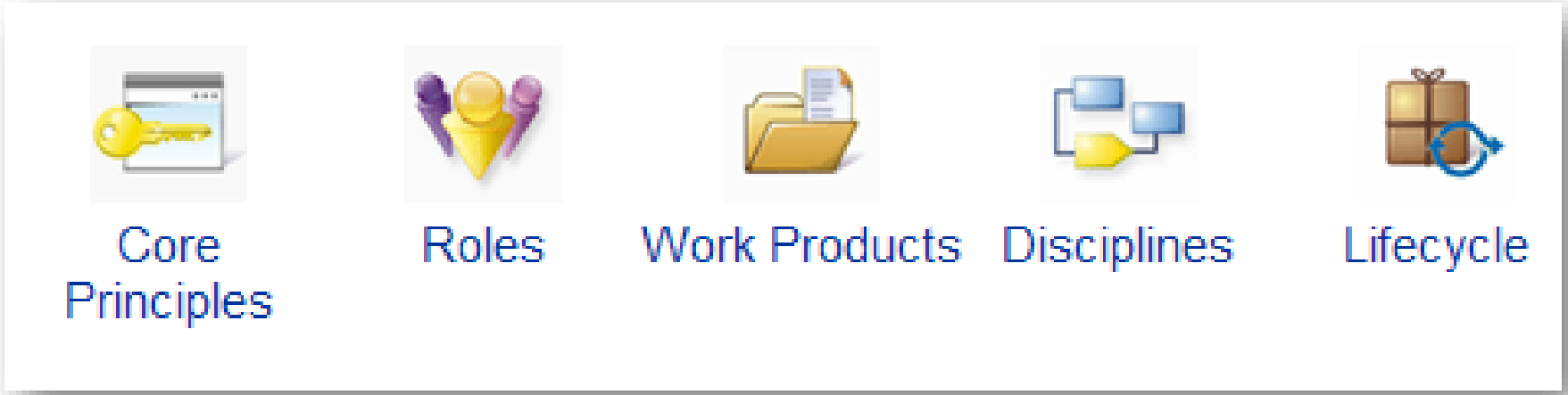
Phase	*Products, Outputs, or Deliverables*
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and IS management are the result of associated systems;
	Detailed steps, or work plan, for project;
	Specification of system scope and planning and high-level system requirements or features;
	Assignment of team members and other resources;
	System justification or business case
Analysis	Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system;
	Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs);
	Technical, detailed specifications of all system elements (programs, files, network, system software, etc.);
	Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

Credit: Hoffer et al, "Modern Systems Analysis and Design", 5th ed.

O SDLC é concretizado em **processos de desenvolvimento**

Adotar um processo de engenharia testado & (a)provado

O que é que inclui um processo?



http://epf.eclipse.org/wikis/openup/

Ciclo de vida de desenvolvimento de sistemas (SDLC)

o processo de determinação de como um sistema de informação (SI) pode suportar as necessidades das empresas, projetando um sistema, construindo-o e entregando-o aos utilizadores.

Papel do Analista

Um papel-chave no SDLC é o **analista** de sistemas, que analisa a situação do negócio, identifica oportunidades de melhorias e projeta um sistema de informação para implementá-los. Ser analista de sistemas é um dos trabalhos mais desafiantes na eng.a de software.

O principal objetivo de um analista de sistemas não é criar um sistema "topo de gama", mas **criar valor para a organização.**

The Systems Analyst: Skills

Agents of change

Identify ways to improve the organization
Motivate & train others

Skills needed

Technical: must understand the technology

Business: must know the business processes

Analytical: must be able to solve problems

Communications: technical & non-technical audiences

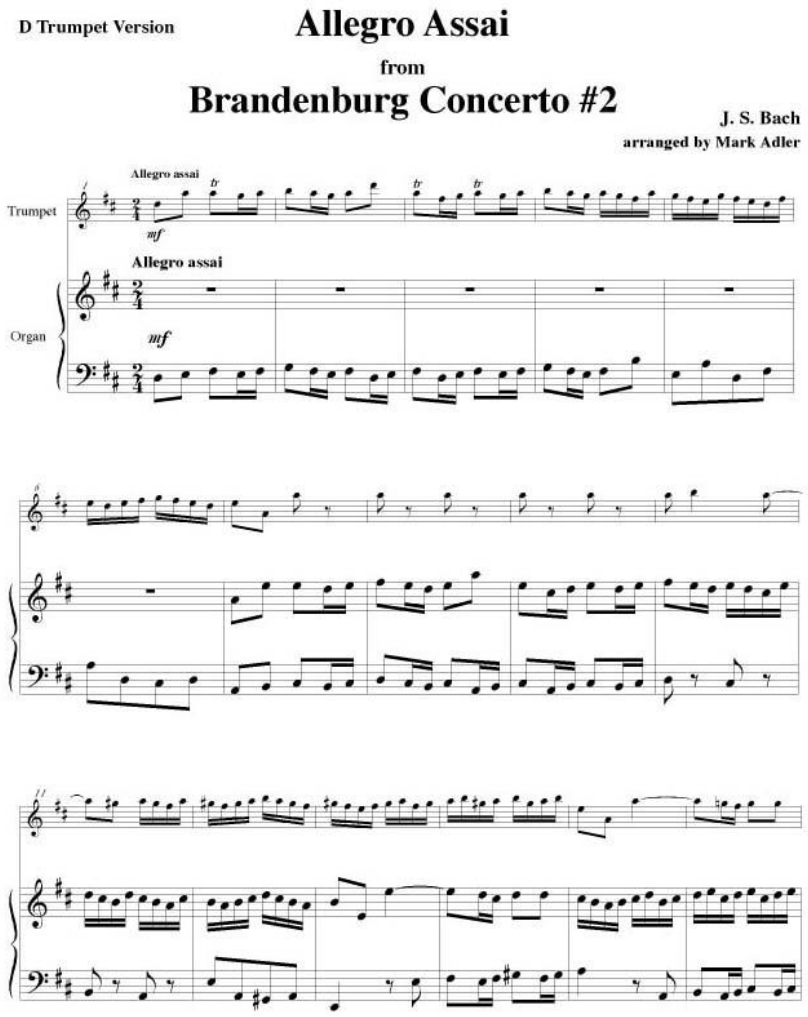
Interpersonal: leadership & management

Ethics: deal fairly and protect confidential information

Credit: Dennis et al, "*Systems Analysis and Design: An Object Oriented Approach with UML*", 5th ed.

PAPEL DOS MODELOS NO DESENVOLVIMENTO DE SW

Usamos **modelos visuais para captar partes do mundo/realidade**



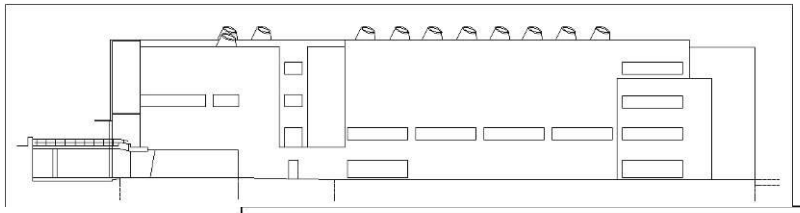
Uma linguagem comum (escrever, ler)

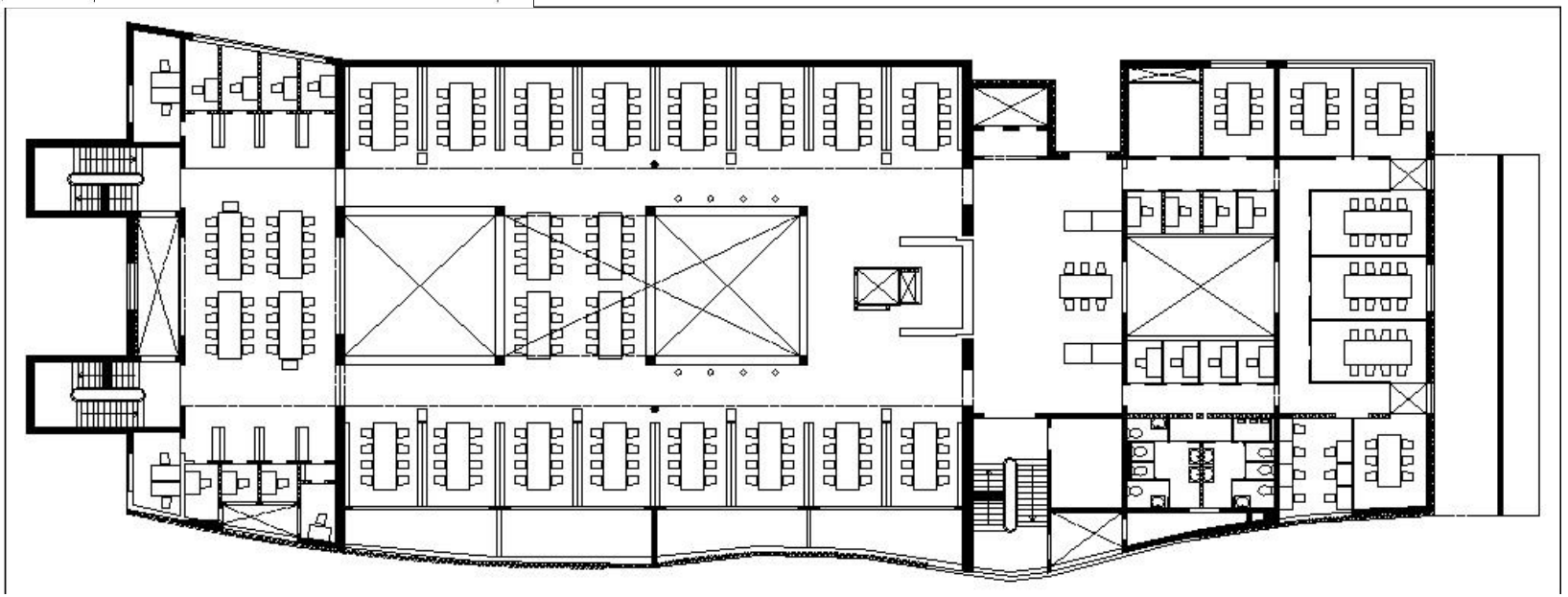
Especificações visuais são mais ricas a mostrar relações entre os conceitos

Compor: aplicar talento e disciplinas técnicas

Orquestra: a prova que os modelos funcionam!

Um modelo é uma simplificação da realidade

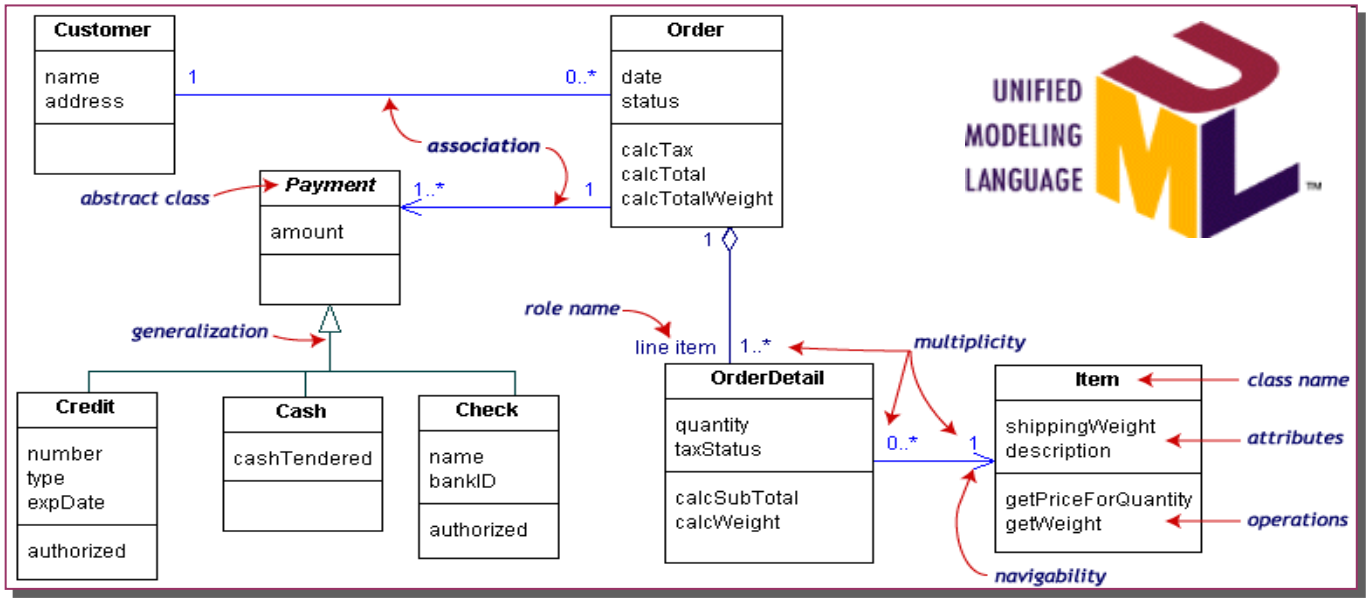




Os modelos ajudam a **gerir a complexidade**

Grady Booch apresenta 4 razões para usar modelos:

- Ajudar a **visualizar um sistema**, como se pretende que venha a ser
- Especificar a **estrutura e o comportamento** do sistema (antes de implementar)
- Serve como **referência / orientação** para a construção (“planta”)
- **Documentar** as decisões (de desenho) que foram feitas



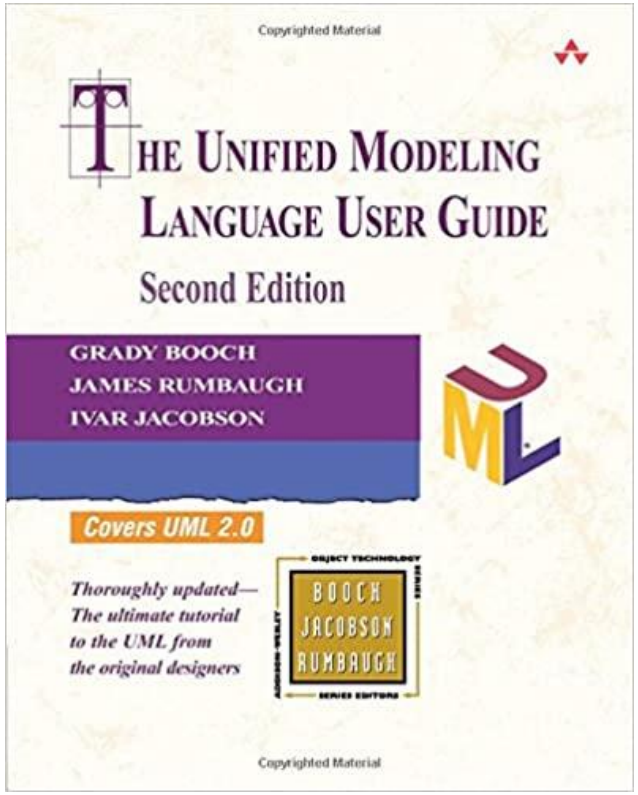
Modelação visual no desenvolvimento

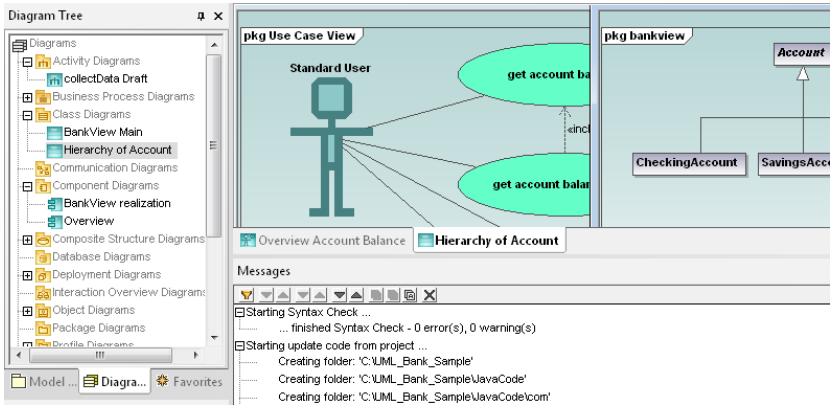
UML 2: Unified Modeling Language

Linguagem de modelação normalizada, orientada para a área de eng.a de software

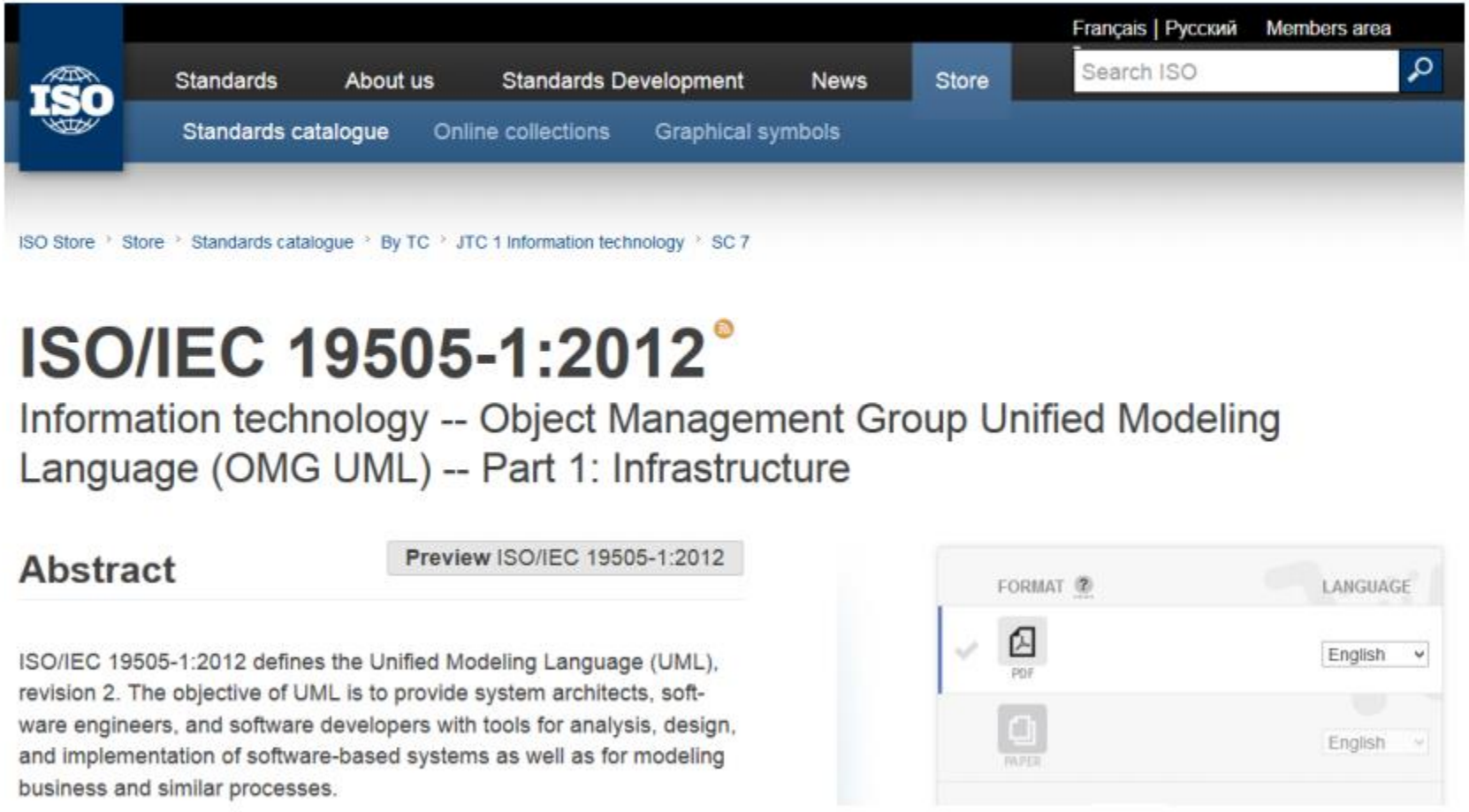
Benefícios

- Promover a comunicação mais clara e sucinta
- Manter o desenho (planeamento) e a implementação (construção) coerentes
- Mostrar ou esconder diferentes níveis de detalhe, conforme apropriado
- Pode suportar, em parte, processos de construção automática (gerar a solução a partir do modelo)

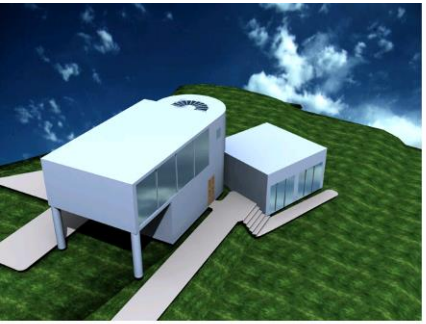
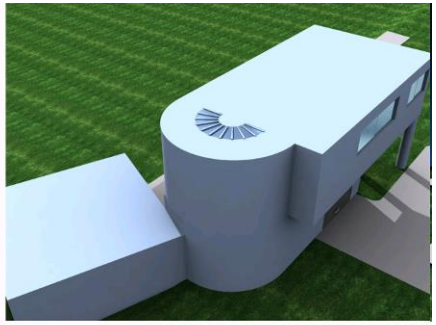




Também reconhecida como um standard internacional ISO



Não há uma vista única, mas várias e complementares para descrever um sistema

Para que serve o sistema?

→ análise funcional

Quais são as estruturas de informação?

→ Análise das estruturas de dados

Decomposição funcional de atividades complexas

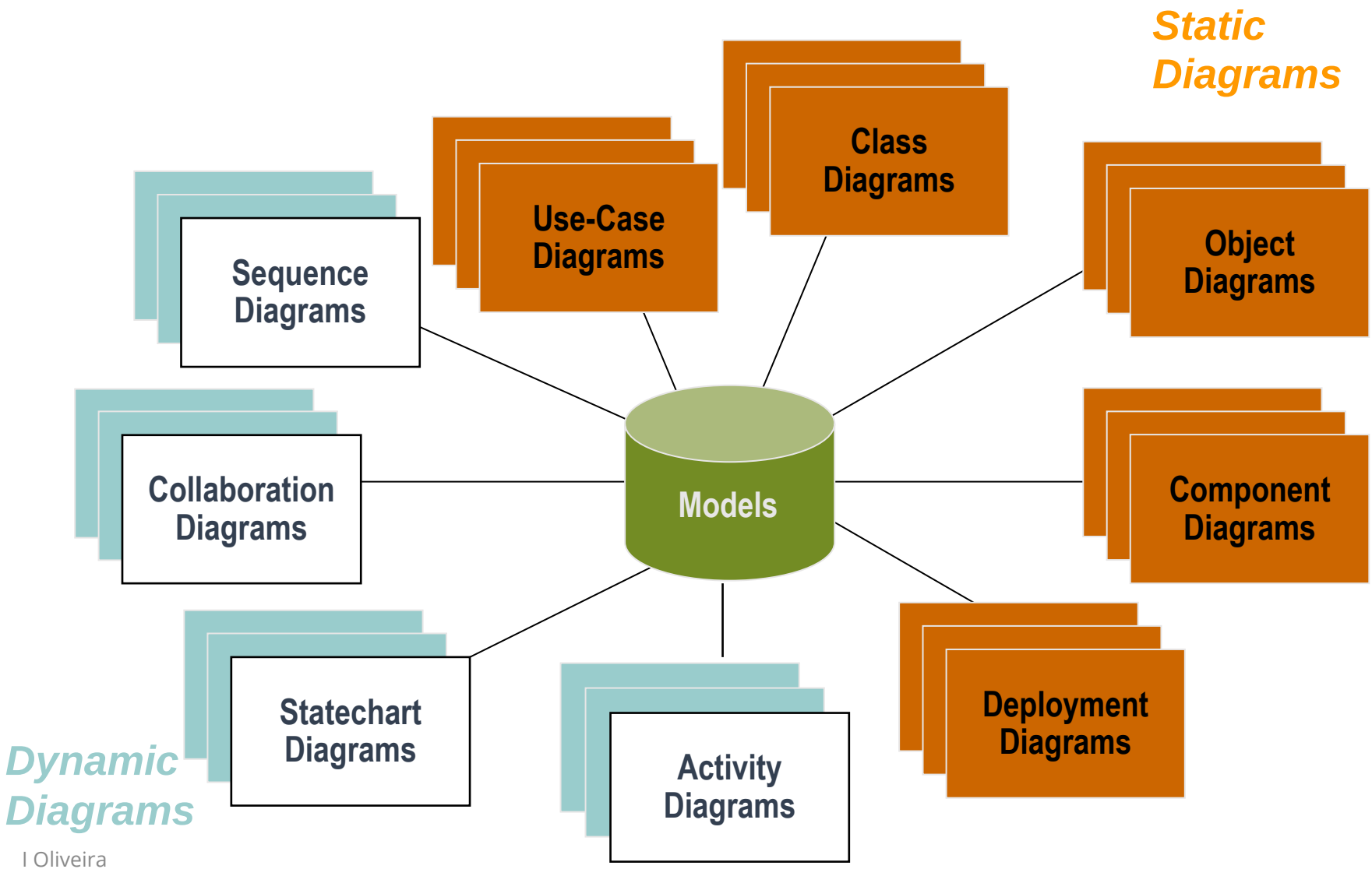
→ Colaboração entre peças/módulos

Visualizar a organização do software em partes e as suas interações

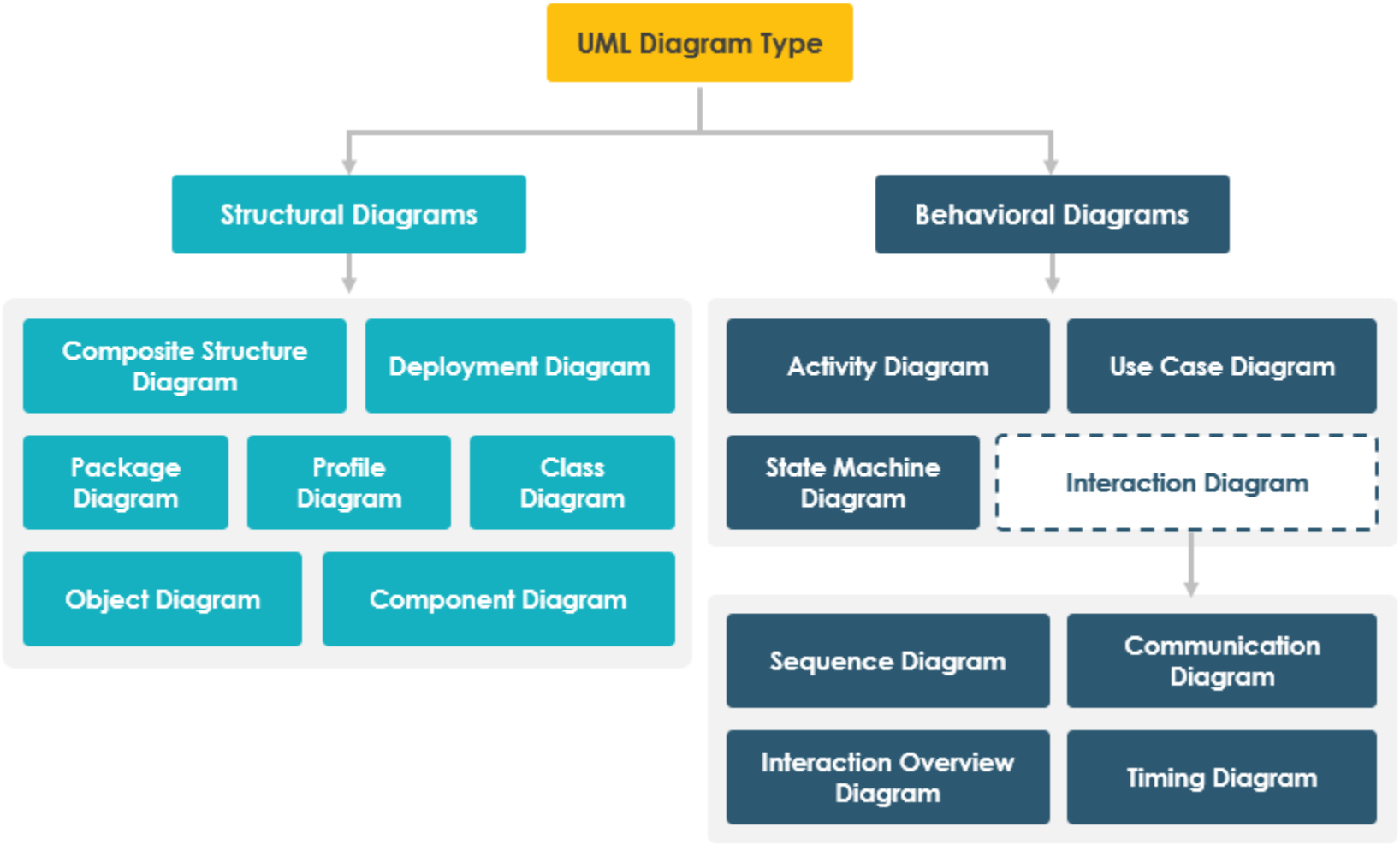
→ Explicitar arquitetura, componentes

Etc.

A UML fornece a sintaxe e a semântica para modelar diferentes aspetos de um SI



Diagramas da UML 2.x



https://www.visual-paradigm.com/guide/uml-unified-modeling-language/overview-of-the-14-uml-diagram-types/

Referências

Fowler, Martin. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 2003.

Pressman, Roger. *Software Engineering: A Practitioner's Approach*. 7th ed. McGraw-Hill, 2009.

Silva, Alberto, and Carlos Videira. *UML - Metodologias E Ferramentas CASE (vol. I E II)*. 2a ed. Centro Atlântico, 2008.