Desenvolvimento de Sistemas Software

Aula Teórica 25

Notas finais

Desenvolvimento de Sistemas Software
José Creissac Campos / António Nestor Ribeiro

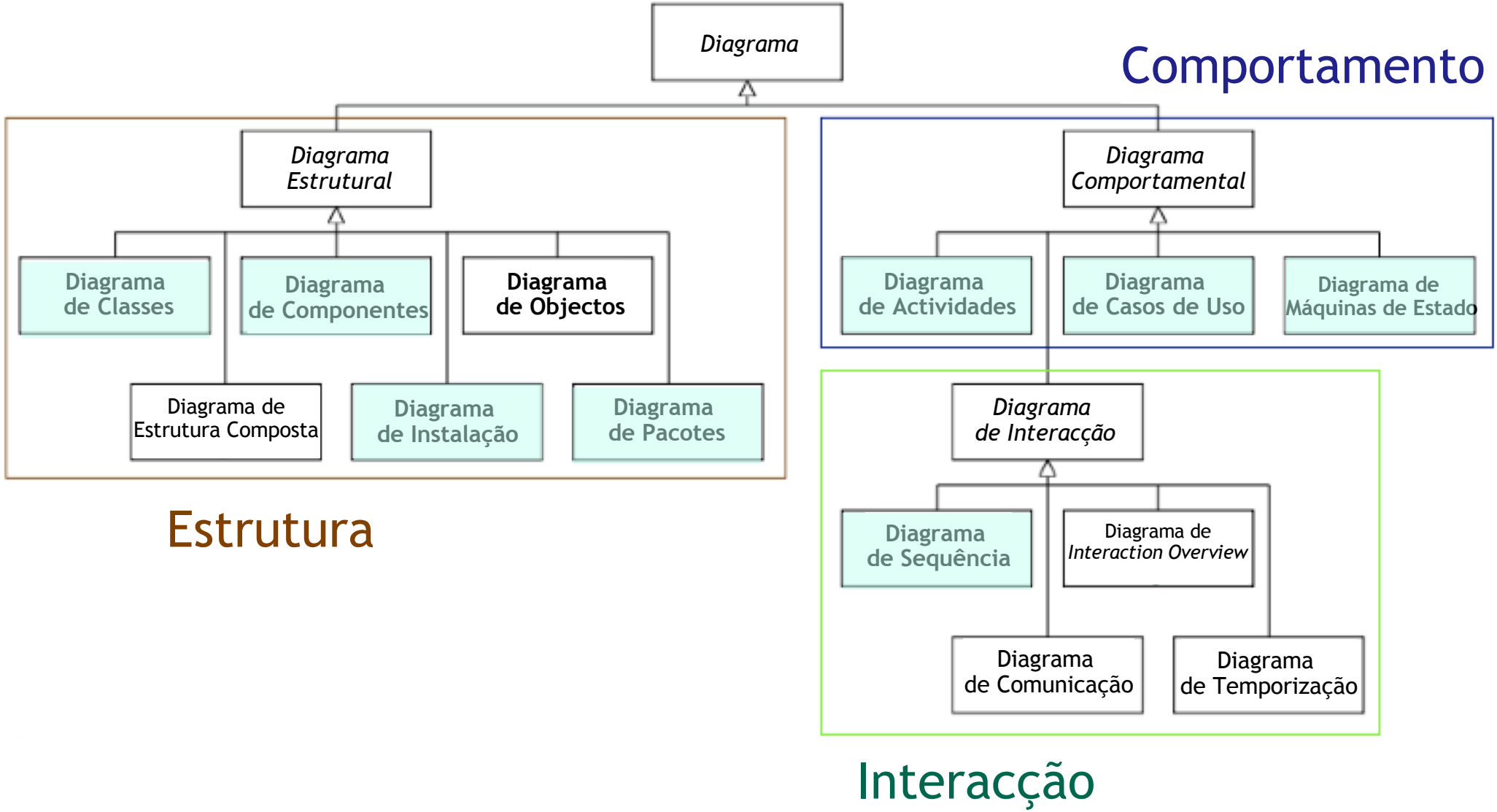
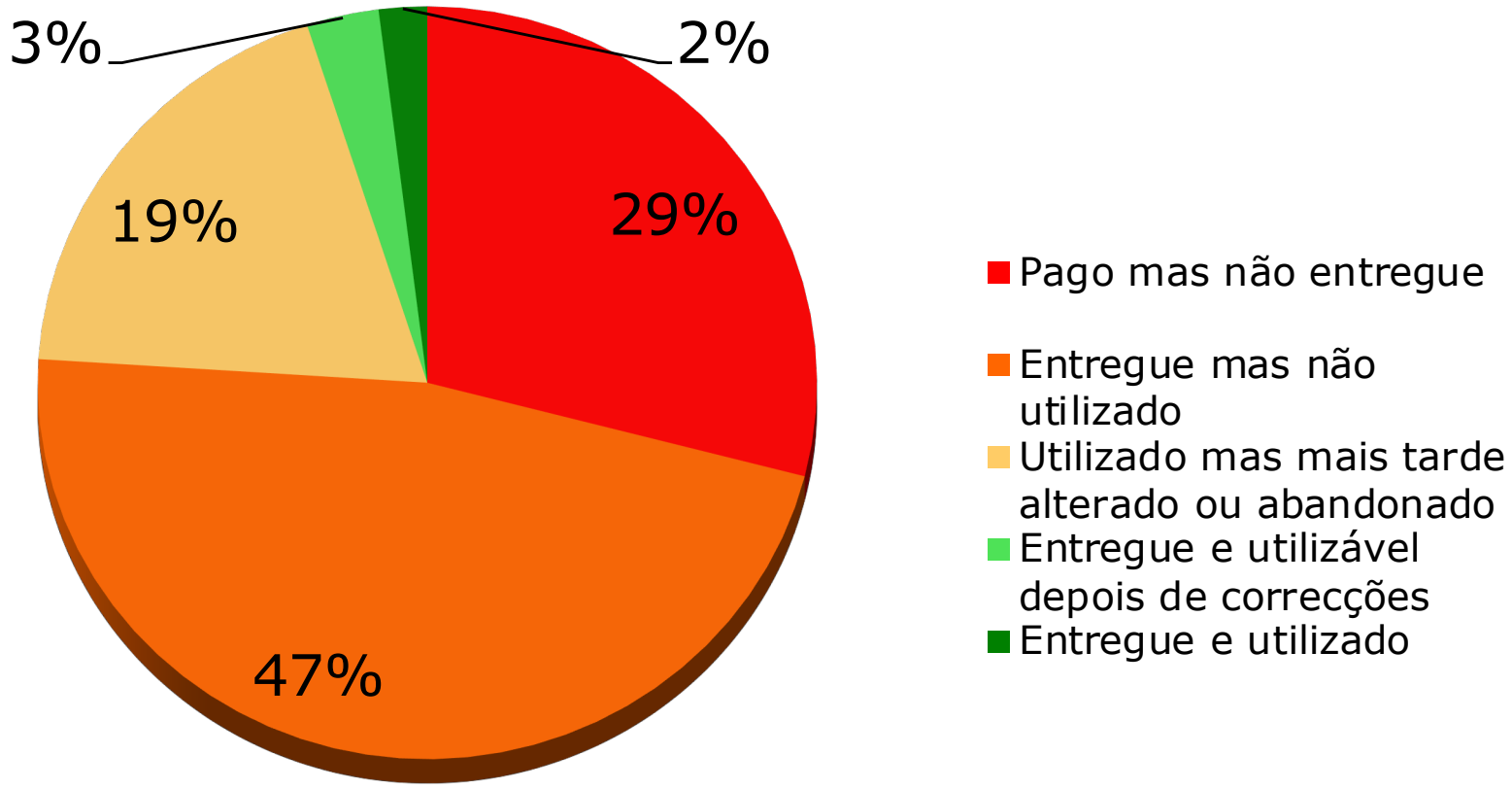


Diagrama
Comportamento
Diagrama Estrutural
Diagrama de Classes
Diagrama de Componentes
Diagrama de Objectos
Diagrama de Estrutura Composta
Diagrama de Instalação
Diagrama de Pacotes
Estrutura
Diagrama Comportamental
Diagrama de Actividades
Diagrama de Casos de Uso
Diagrama de Máquinas de Estado
Diagrama de Interacção
Diagrama de Sequência
Diagrama de Interaction Overview
Diagrama de Comunicação
Diagrama de Temporização
Interacção

Estado da arte em 1992...



- Mais de 75% do software pago não chegou a ser utilizado!
- Apenas 5% do software pago foi utilizado continuadamente (deste, 3% necessitou de correcções).

Fonte: GAO, 1992

Estado da arte em 2019?

Forbes / Entrepreneurs / #CuttingEdge

JUL 11, 2016 @ 04:41 AM 22,468

Robots Replacing Developers? This Startup Uses Automation To Build Smart Software

Julian Mitchell, CONTRIBUTOR
I cover entrepreneurs and startups disrupting industries.
FULL BIO

Quora Home 1 Answer Notifications Search Quora

Could artificially intelligent computers replace programmers in the future?

Answer Request Follow 124 Comment 1 Downvote

Promoted by VisionMobile

How well do you know your tools and platforms?
Take the new developer economics survey, along with another 40,000 developers and find out!

Start now at vmob.me

73 Answers

Joscha Bach
Answered Jun 9, 2016

I am surprised that most answers seem to be in the negative, or think that programming is one of the last fields where human labor becomes obsolete. General artificial intelligence will of course replace programmers at some point: the question is not if, but when, and at what level of sophistication of AI.

COMMUNICATIONS OF THE ACM

HOME | CURRENT ISSUE | NEWS | BLOGS | OPINION | RESEARCH | PRACTICE

Home / News / AI Will Replace Coders by 2040, Warn Academics / Full Text

ACM TECHNEWS

AI Will Replace Coders by 2040, Warn Academics

By V3.co.uk
December 8, 2017
Comments

VIEW AS: SHARE:

Display a menu

Coders and programmers could find themselves becoming marginalized by emerging technologies such as artificial intelligence, with humans being replaced in these jobs by 2040, according to a study from academic researchers published by Oak Ridge National Laboratory.

The report predicted by 2040, machine learning and natural language processing technologies will have become so advanced they will be able to write better software code faster than the best human coders.

In addition, "the major technologies that will drive the creation and adoption of machine-generated code already exist, either at

SIGN IN for Full Access
User Name
Password
» Forgot Password?
» Create an ACM Web Account
SIGN IN

MORE NEWS & OPINIONS

As Scrutiny Of Social Networks Grows, Influence Attacks Continue In Real Time
NPR

Connectivity Hacking Back Makes a Comeback—But It's Still

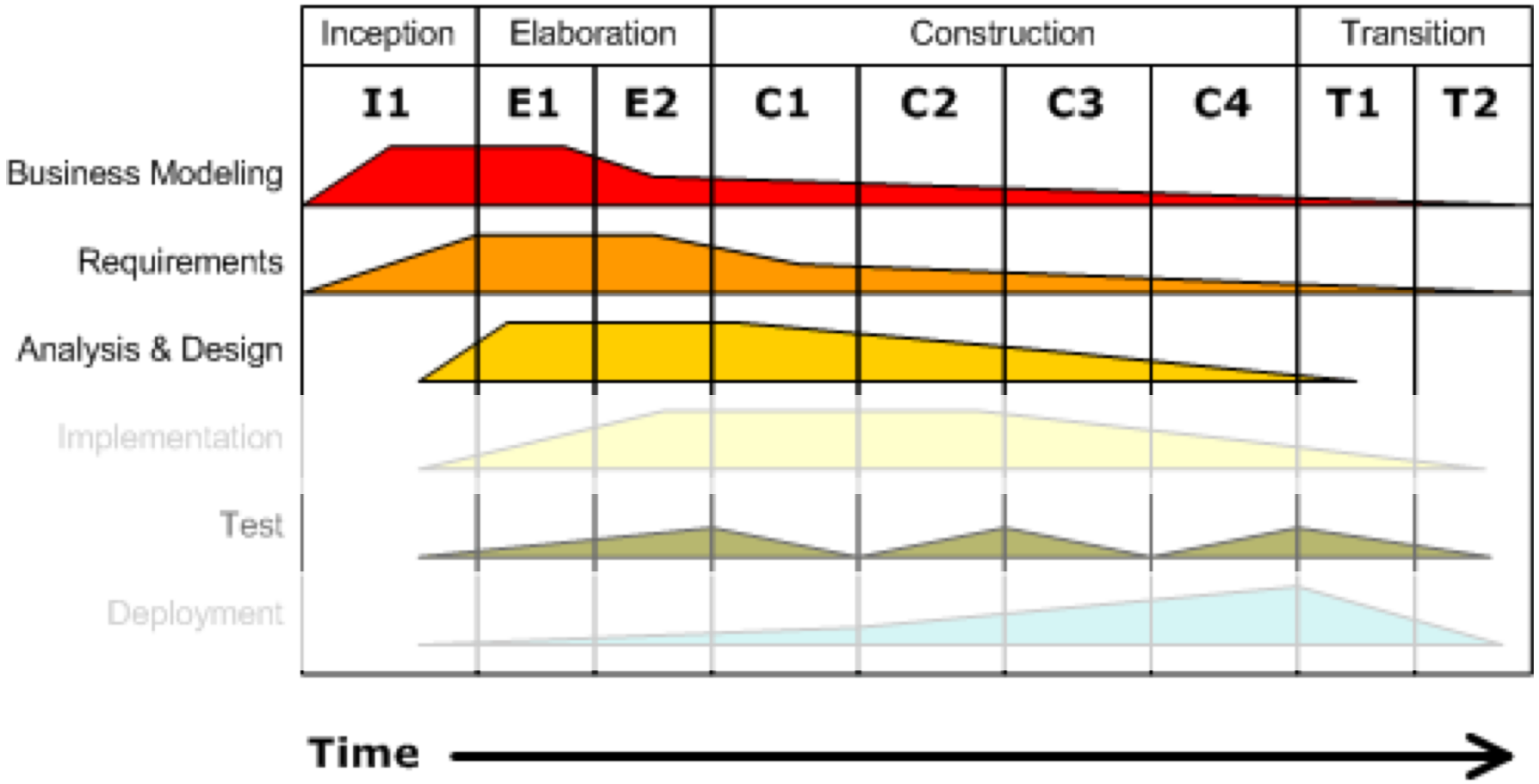
OVERVIEW | LEARN | HIGHEST RATED | FEATURES

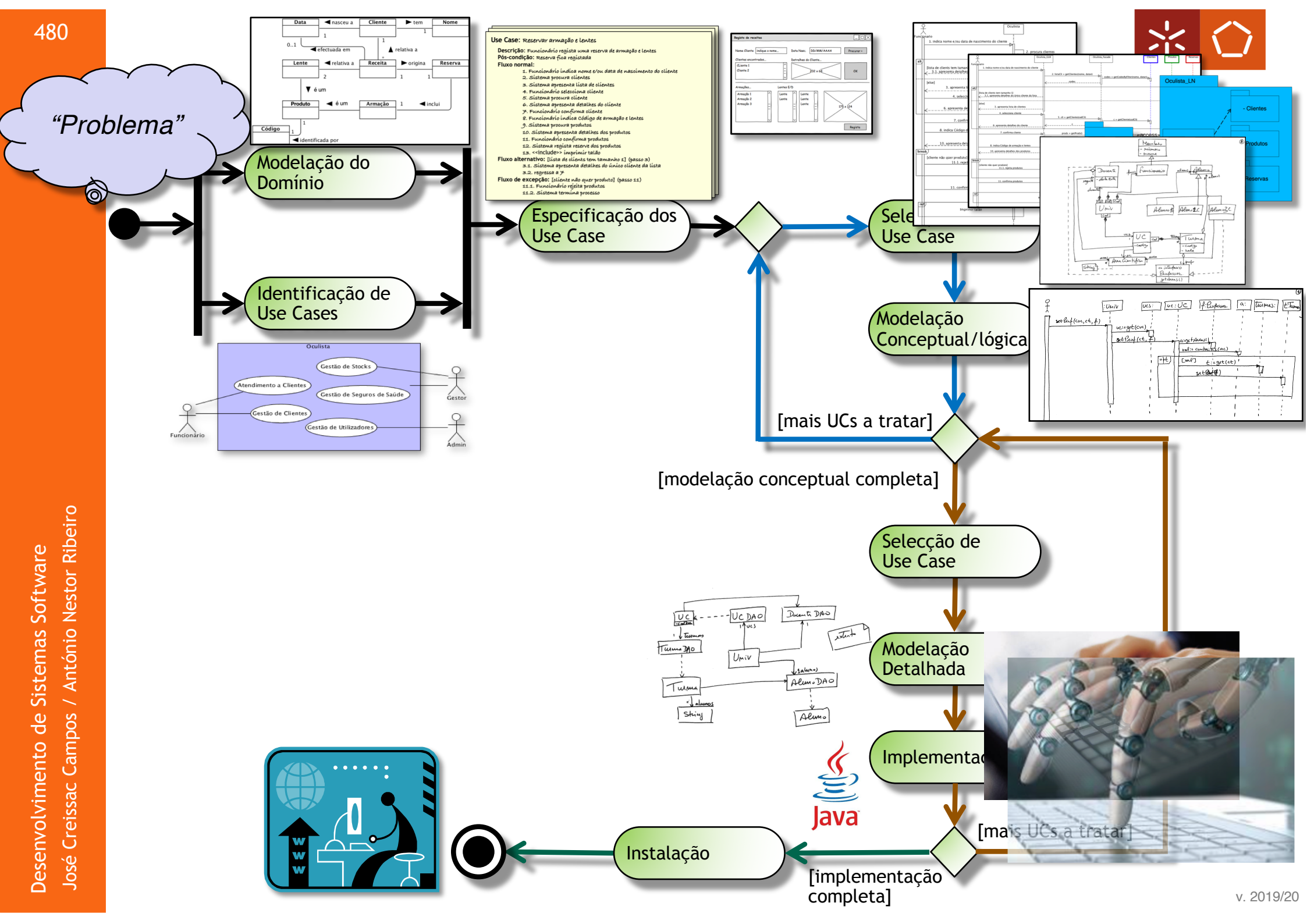
Best No-Code Development Platforms Software

No-Code Development Platform provide drag-and-drop tools that that allow businesses to develop software quickly without coding. The platforms provide WYSIWYG editors and drag-and-drop components to quickly assemble and design applications. Both developers and nondevelopers can use these tools to practice rapid application development with customized workflows and

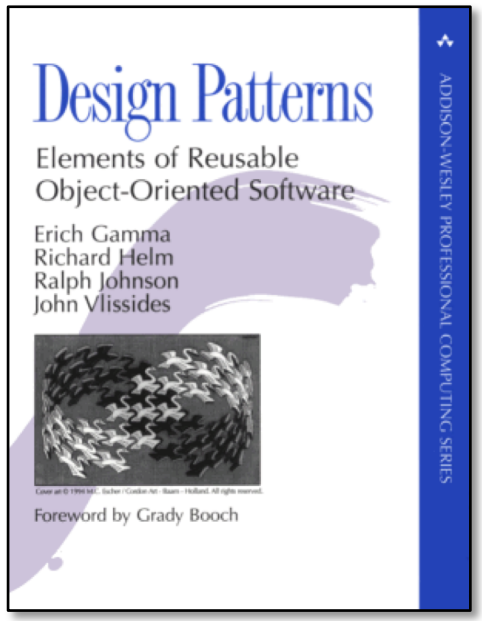
SHOW MORE

(Rational) Unified Process



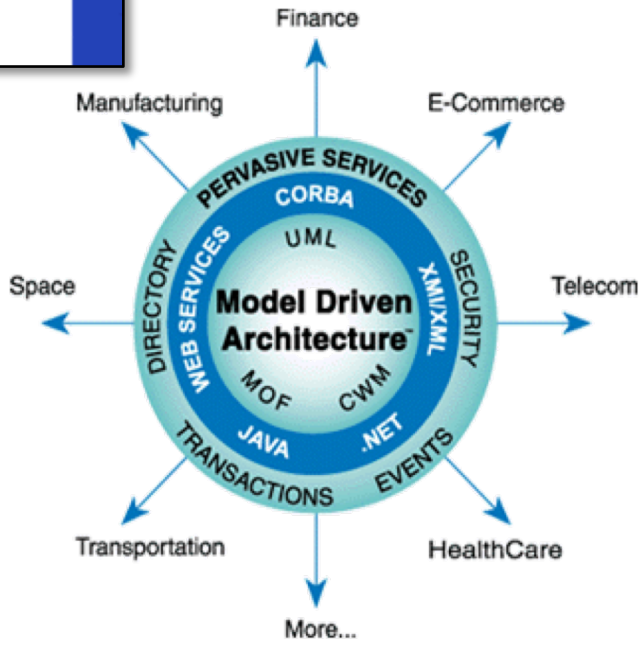

"Problema"
Modelação do Domínio
Identificação de Use Cases
Especificação dos Use Case
Use Case: Reservar armação e lentes
Descrição: Funcionário regista uma reserva de armação e lentes
Pós-condição: Reserva fica registada
Fluxo normal:
1. Funcionário indica nome e/ou data de nascimento do cliente
2. Sistema procura clientes
3. Sistema apresenta lista de clientes
4. Funcionário selecciona cliente
5. Sistema procura cliente
6. Sistema apresenta detalhes do cliente
7. Funcionário confirma cliente
8. Funcionário indica Código de armação e lentes
9. Sistema procura produtos
10. Sistema apresenta detalhes dos produtos
11. Funcionário confirma produtos
12. Sistema regista reserva dos produtos
13. <<include>> imprimir talão
Fluxo alternativo: [lista de clientes tem tamanho 1] (passo 3)
3.1. Sistema apresenta detalhes do único cliente da lista
3.2. regressa a 7
Fluxo de excepção: [cliente não quer produto] (passo 11)
11.1. Funcionário rejeita produtos
11.2. Sistema termina processo
Oculista
Gestão de Stocks
Atendimento a Clientes
Gestão de Seguros de Saúde
Gestão de Clientes
Gestão de Utilizadores
Funcionário
Gestor
Admin
Sele Use Case
Modelação Conceptual/lógica
[mais UCs a tratar]
[modelação conceptual completa]
Selecção de Use Case
Modelação Detalhada
Implementa
[mais UCs a tratar]
[implementação completa]
Instalação
Java
Oculista_LN
- Clientes
Produtos
Reservas

Mais...



- Software design patterns

- Model Driven Engineering
  - Model Driven Arquitecture (OMG)



- <pub>Engenharia de Aplicações</pub>
  - Desenvolvimento de Aplicações Multi-camada (Web)

Avaliação

- Exame (≥ 9.0) - uma prova escrita sobre a matéria leccionada
  - Exame de consulta
- Trabalho Prático (≥ 10.0)
- Objectivos
  - Reconhecer os diferentes tipos de diagramas da UML ;
  - Compreender modelos (de requisitos/estruturais/comportamentais) descritos em UML;
  - Conceber sistemas de software utilizando UML;
  - Implementar sistemas de software a partir de modelos UML.
- Classificação Final (**≥ 10.0**)

  .6 Exame + .4 Trabalho

Sobre o trabalho prático

- Dois deliverables:
  - Relatório com análise e modelação
  - Uma aplicação informática que permita executar os use cases solicitados

- Na apresentação:
  - Avalia-se a qualidade do processo e como é que cada grupo executou as diferentes fases de análise e modelação
  - Avalia-se a construção da aplicação, na sua lógica multi-camada e na arquitectura apresentada

Sobre o trabalho prático…

- Estrutura do relatório:
  1. Introdução (objectivos e principais decisões tomadas)
  2. Modelação
     1. Modelo de domínio
     2. Modelo de Use Cases (diagrama e especificação textual de cada UC)
     3. Protótipo da interface
     4. Considerando os 10 use case solicitados na Fase 2
        1. Diagramas de sequência com subsistemas para cada Use Case
        2. Diagrama de packages
        3. Diagrama de classe
        4. Diagrama de sequência das operações de cada subsistema
  3. Implementação – Considerando os 5 use case solicitados na Fase 2
     1. Diagrama de classe com DAOs / indicação das tabelas do modelo relacional
     2. Detalhes relevantes da implementação
     3. Descrição da interface
  4. Conclusões
     1. Como avalia o resultado obtido, o processo de modelação seguido e o de construção da aplicação

Sobre o trabalho prático…

- Relatório:
  - Itens opcionais:
    - Diagrama de estado para as entidades mais relevantes (interface?)
    - Diagrama de sequência de nível de implementação para métodos mais relevantes (se necessário)
    - Diagrama de instalação

Modelação Estrutural/Modelação Comportamental

Sumário

- Notas finais