

Engenharia de Software Moderna

Cap. 4 - Modelos

Prof. Marco Tulio Valente

Motivação

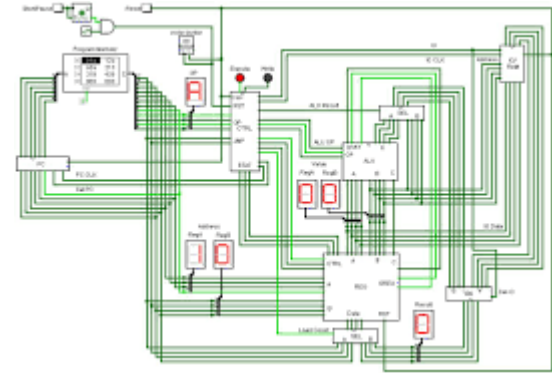
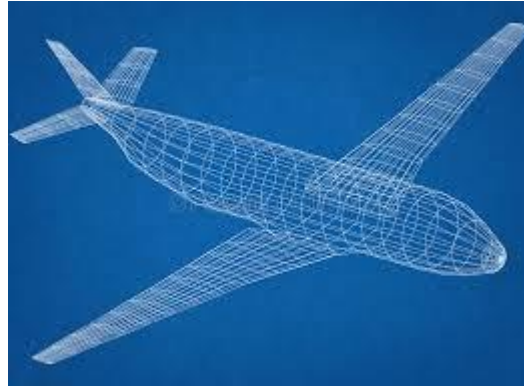
- Existe uma lacuna entre os seguintes mundos:
 - Requisitos: "o que" o sistema faz (nível de abstração mais alto)
 - Código: "como" o sistema opera (nível de abstração mais baixo)

Modelos de Software

- Objetivo: preencher essa "lacuna"
- Via uma notação com um nível de abstração intermediário
- Documentar uma solução para o problema delimitado pelos requisitos

Comuns em outras Engenharias

- Natural que fossem propostos também para software



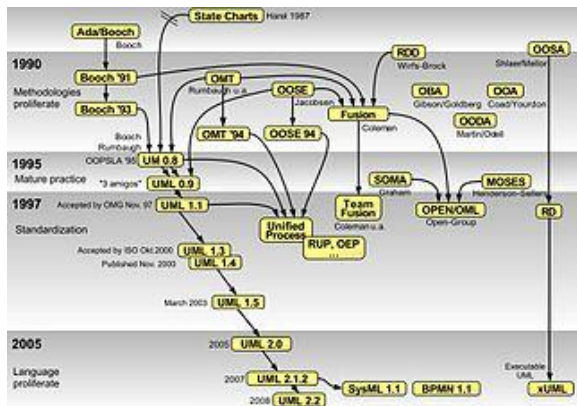
Modelos de Software

- Infelizmente, não são tão efetivos e largamente usados, como em outras engenharias
- Modelos de software podem ser:
 - Formais: menos comuns; não serão estudados aqui
 - Gráficos: UML é a notação mais comum

UML: Unified Modelling Language



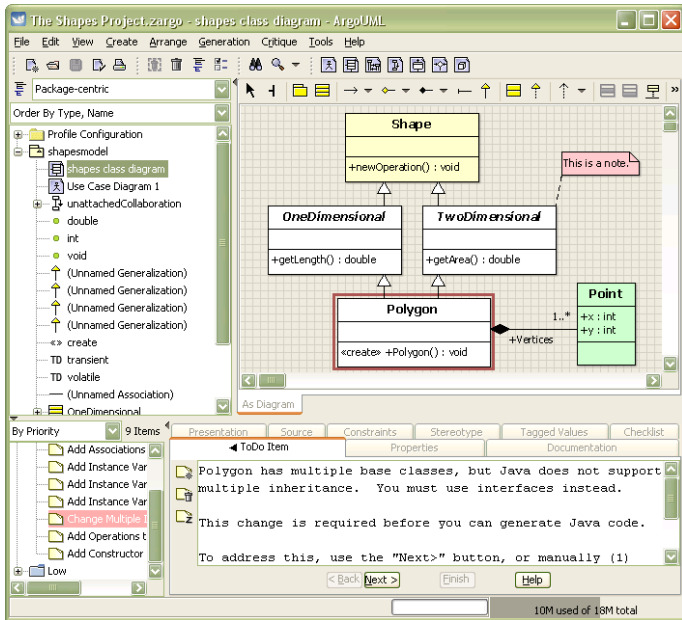
- Proposta em 1995, para fundir outras notações
- Processo mais comum na época: RUP
 - Documentação e planejamento detalhados
 - Código era escrito após meses ou anos de trabalho



Fonte: Wikipédia

Ferramentas CASE

- CASE: Computer-Aided Software Engineering
- Equivalente a ferramentas CAD, mas para Eng. Software



Quais os principais usos de UML?

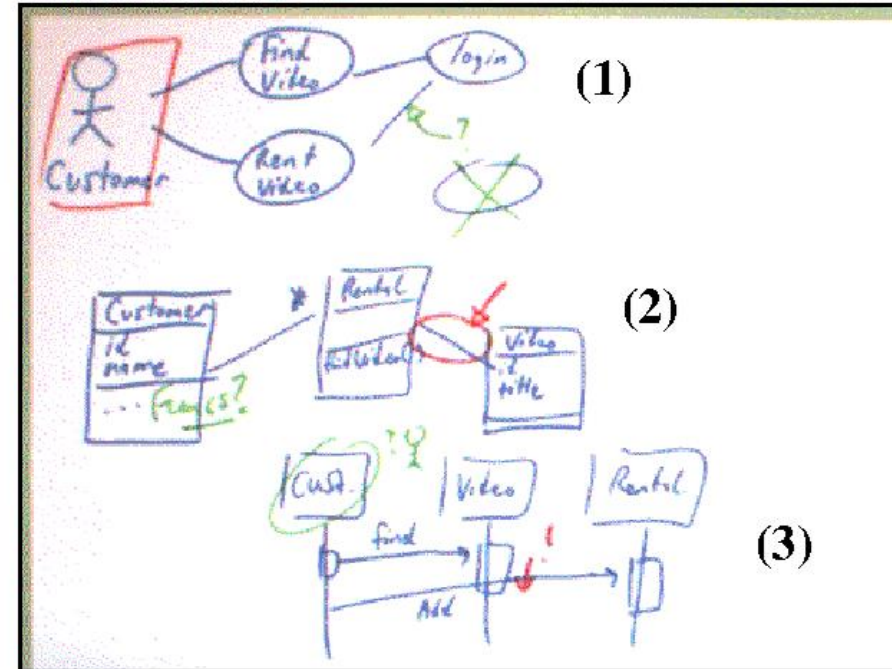
1. Como blueprint (planta detalhada)
2. Como sketches (esboços, rascunhos)

Nesta disciplina, vamos estudar o uso de
UML como sketches

UML como Sketch

- Uso mais comum de UML com métodos ágeis
- UML é usada para:
 - Conversar sobre uma parte do código ou do projeto
 - Documentar uma parte do código ou do projeto
- Uso mais informal e leve da notação
- Objetivo **não** é ter um modelo completo

UML como Sketch



Sketches UML são úteis em
Engenharia Avante e em Engenharia Reversa

Engenharia Avante (“Forward”)

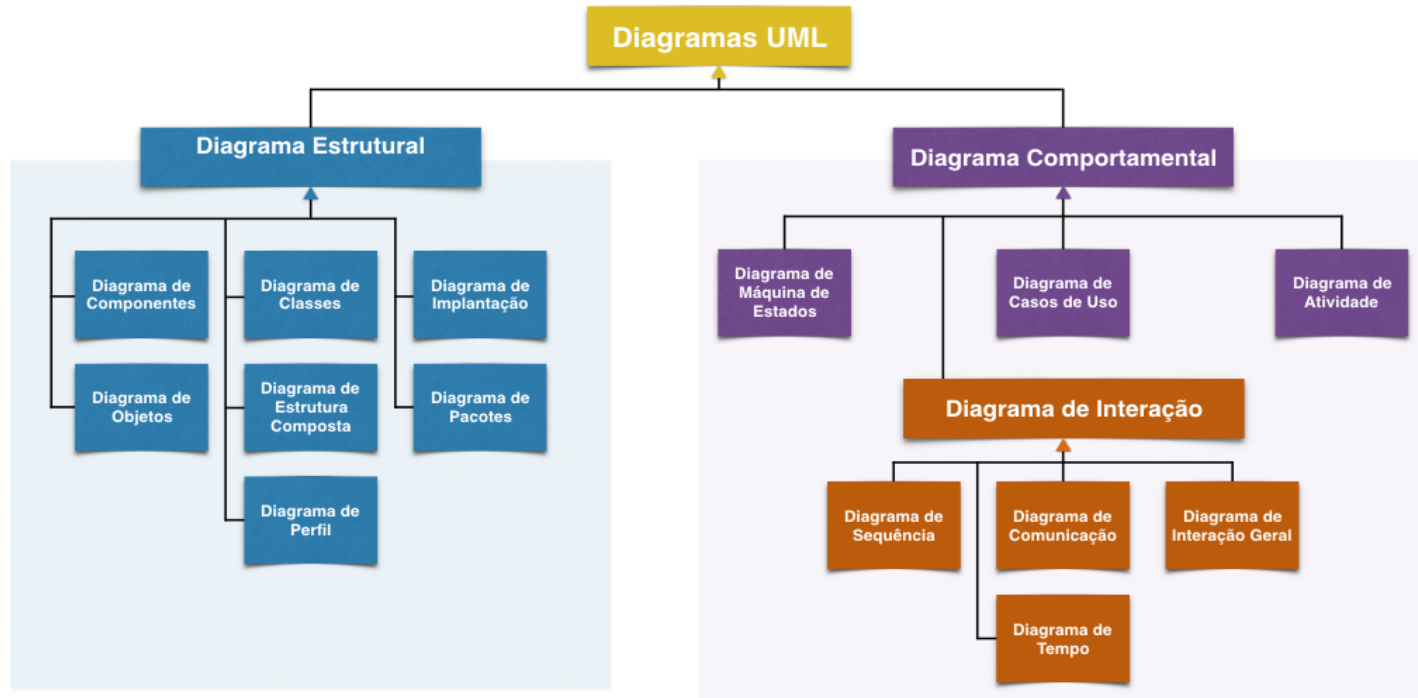
- Modelo é usado para discutir alternativas de projeto
- Antes de qualquer linha de código ser implementada

Engenharia Reversa

- Modelo é usado para explicar um código que já existe
- Contextos de manutenção e evolução de software

Diagramas UML

- UML é uma notação gráfica para modelagem de software
- Dois grupos de diagramas:
 - Diagramas Estáticos: modelam a estrutura do código
 - Diagramas Dinâmicos: modelam a execução do código (o comportamento do sistema)



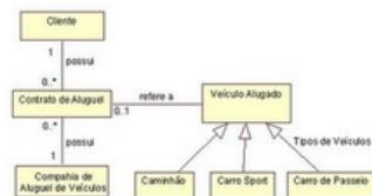


Visão Geral - UML

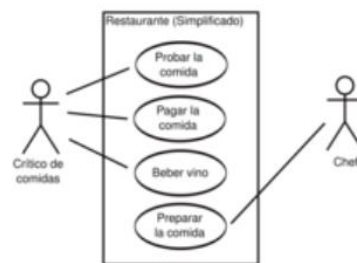
Estudar on-line em https://quizlet.com/_3pq9wn

-
- 1. UML** linguagem gráfica para visualizar, especificar, construir e documentar os artefatos de um sistema de software
 - 2. Diagramas Estruturais** diagramas que definem a arquitetura estática de um modelo
 - 3. Diagramas Comportamentais** diagramas que representam interações e aspectos dinâmicos de um modelo
 - 4. Exemplos de Diagramas Estruturais** classes, objetos, componentes, implantação, perfil, pacotes, estrutura composta
-

25. Diagrama de Classes



26. Diagrama de Casos de Uso



27. Diagrama de Sequência

