

Engenharia de Software II

UNISUL

Ciência da Computação

Apresentação 1 – Conceitos de Modelagem

Carlos A S Rocha

carlos.rocha@unisul.br



Plano de Aula

1. Modelo
2. Sistema de Software
3. Porque Modelar?
4. Desafio
5. Modelagem
6. Modelagem Utilizando Objetos
7. Apresentação da UML
8. Motivação para Modelagem com UML



Modelo Definição

Um modelo é uma simplificação da realidade que nos ajuda a entender um problema grande e complexo que não pode ser compreendido como um todo.

Phillipe Krutchen, 2000



Porque Modelar?

1. Criar abstrações para gerenciar complexidade;
2. Comunicação entre os envolvidos no Projeto;
3. Reduzir o tempo de resposta ao mercado a partir de um problema apresentado;
4. Diminuir custo de desenvolvimento;
5. Gerenciar o risco de erros.



Modelagem de Software

É o desenho, esboço, de uma aplicação de software antes da codificação. Modelagem é uma parte essencial de grandes projetos de software, mas também de grande auxílio para médios e pequenos projetos. Usando um modelo garante-se que todas as responsabilidades de um software foram identificadas, que a funcionalidade de negócio está correta e completa e que as necessidades do usuário final foram encontradas OMG (Object Management Group).

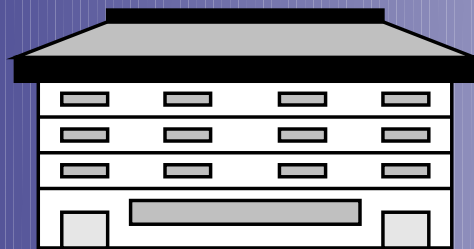


§ Sistema de Software (cont.)

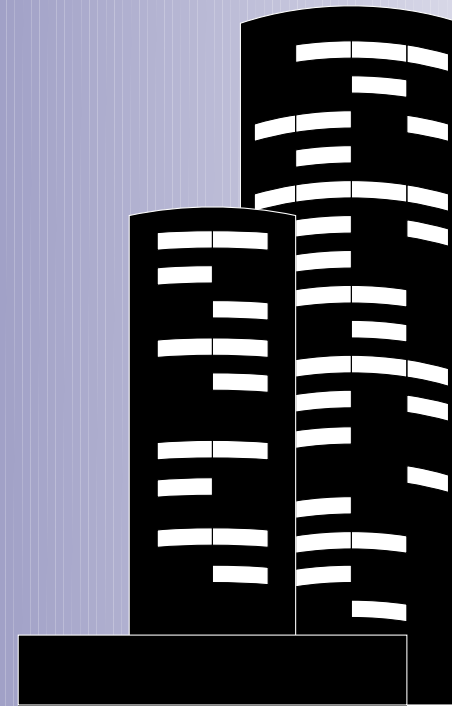
- Desafio...



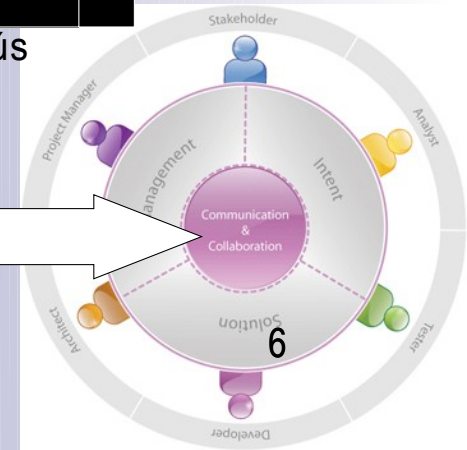
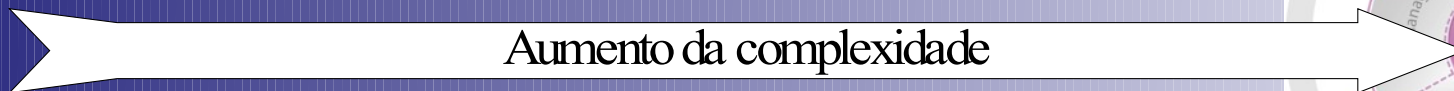
Casa de canhorro



Casa



Arranha-Ceús



Desafio

*Como criar um modelo de software claro,
preciso e ao mesmo tempo simples?*

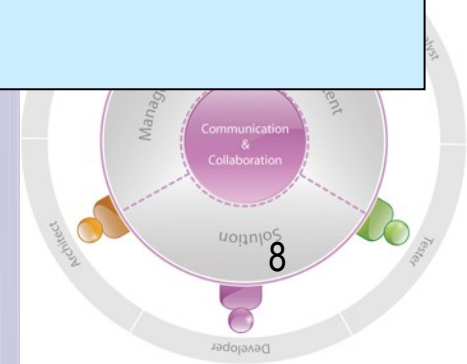
ZINDEL, 2003



Linguagem de Modelagem Unificada – UML

A UML é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de software. Ela oferece uma maneira padronizada para desenvolver projetos de sistemas, incluindo conceitos como processos de negócio e operações de sistemas, assim como coisas concretas como instruções de linguagens de programação, esquemas de banco de dados e componentes reusáveis de software.

OMG (Object Management Group)



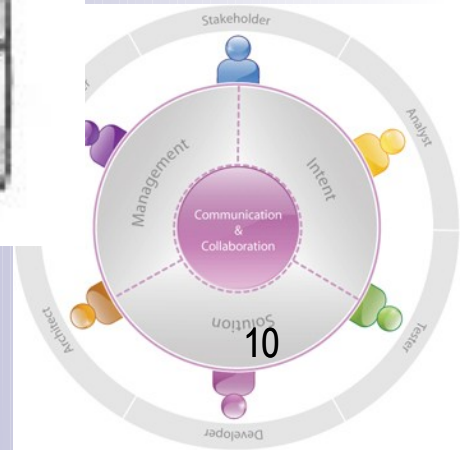
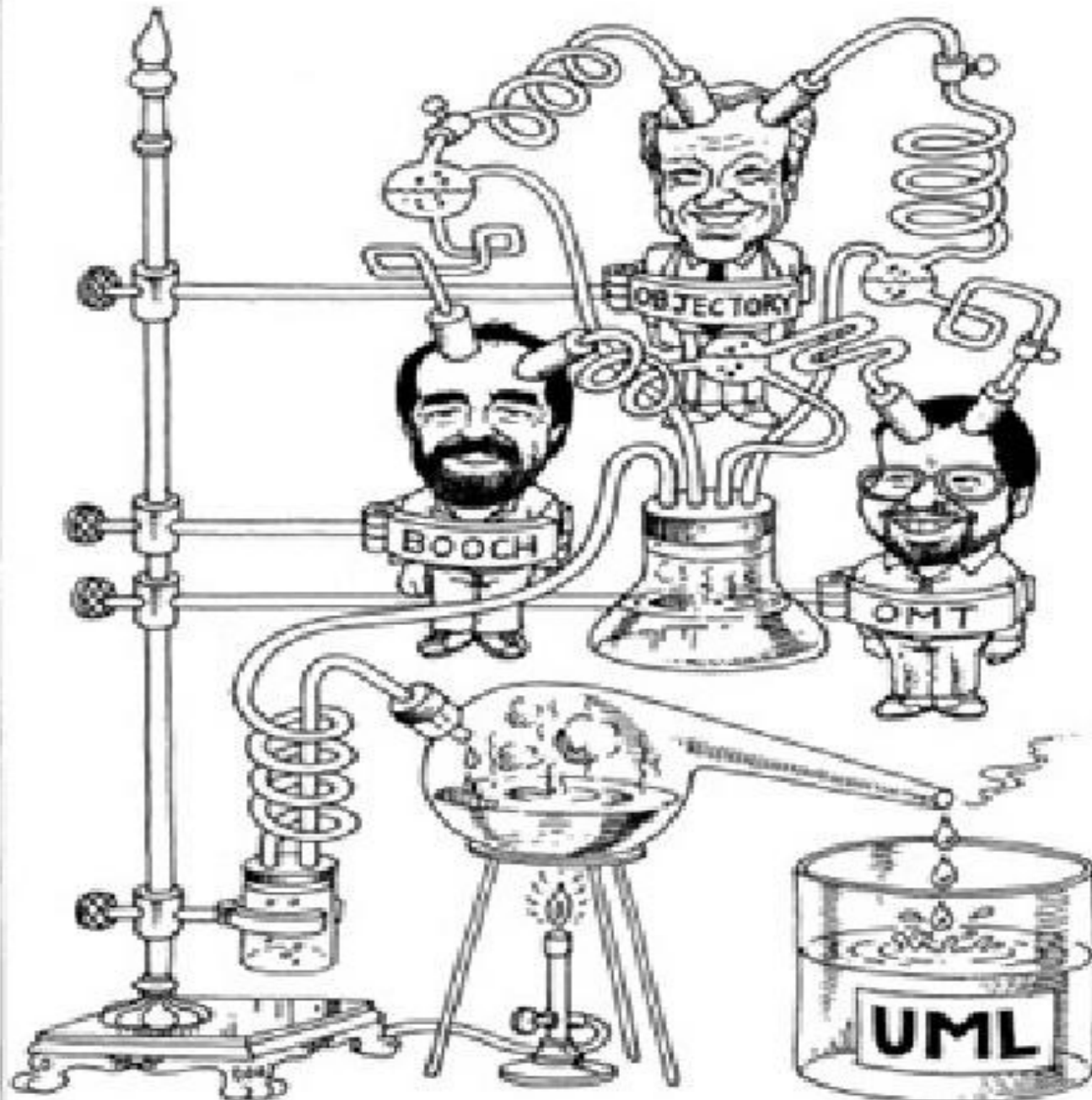
UML

Ela é a síntese de muitas notações: Grady Booch, James Rumbaugh, Ivar Jacobson, entre outros. Originada a partir das três metodologias OO mais populares na década de 90.





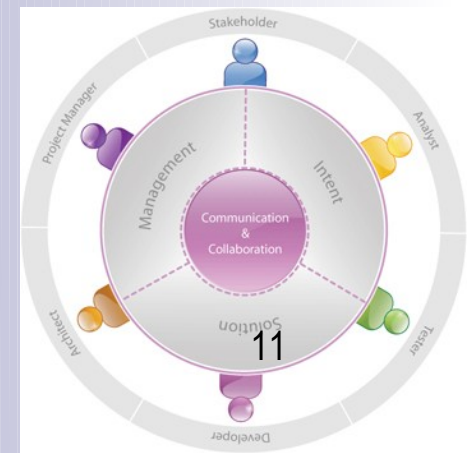
UNISUL



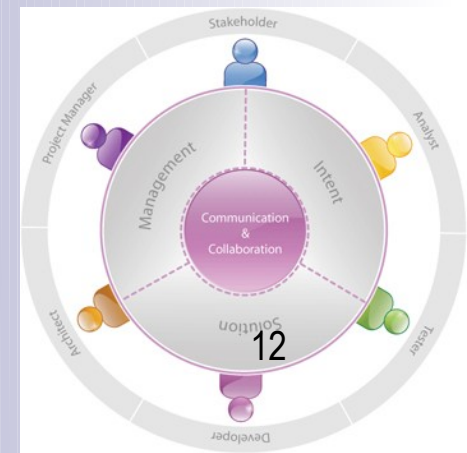
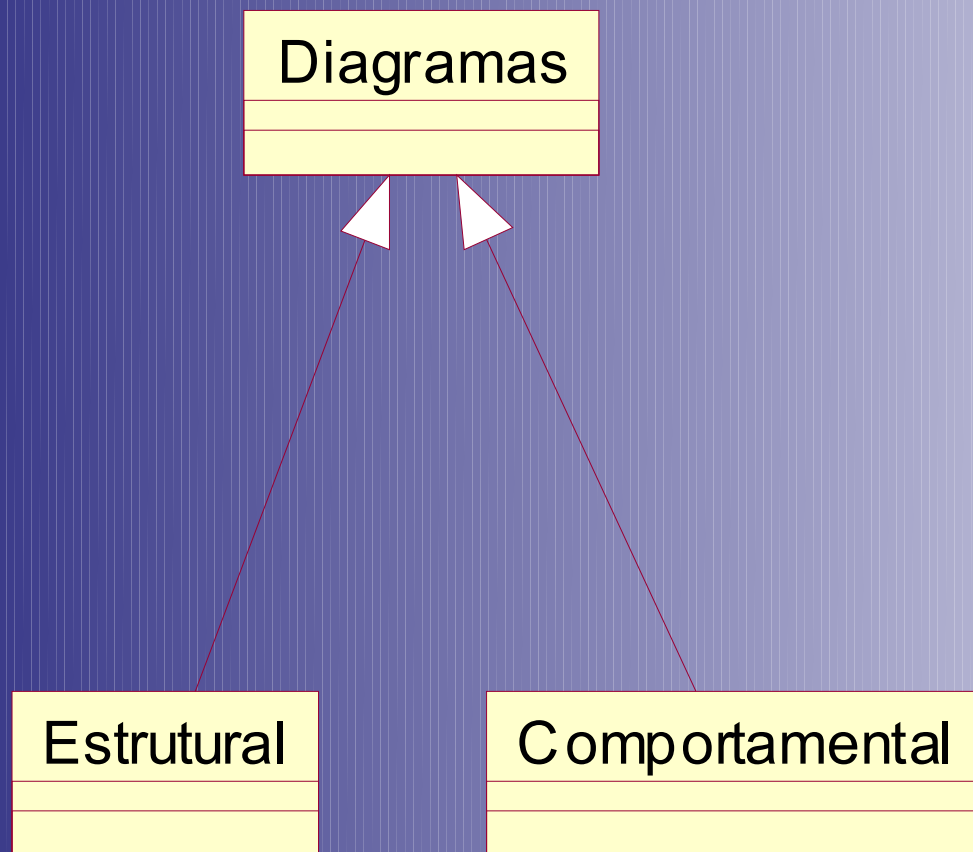
Divisão Geral dos Diagramas

A UML 2.0 define 13 tipos de diagramas, divididos em duas categorias:

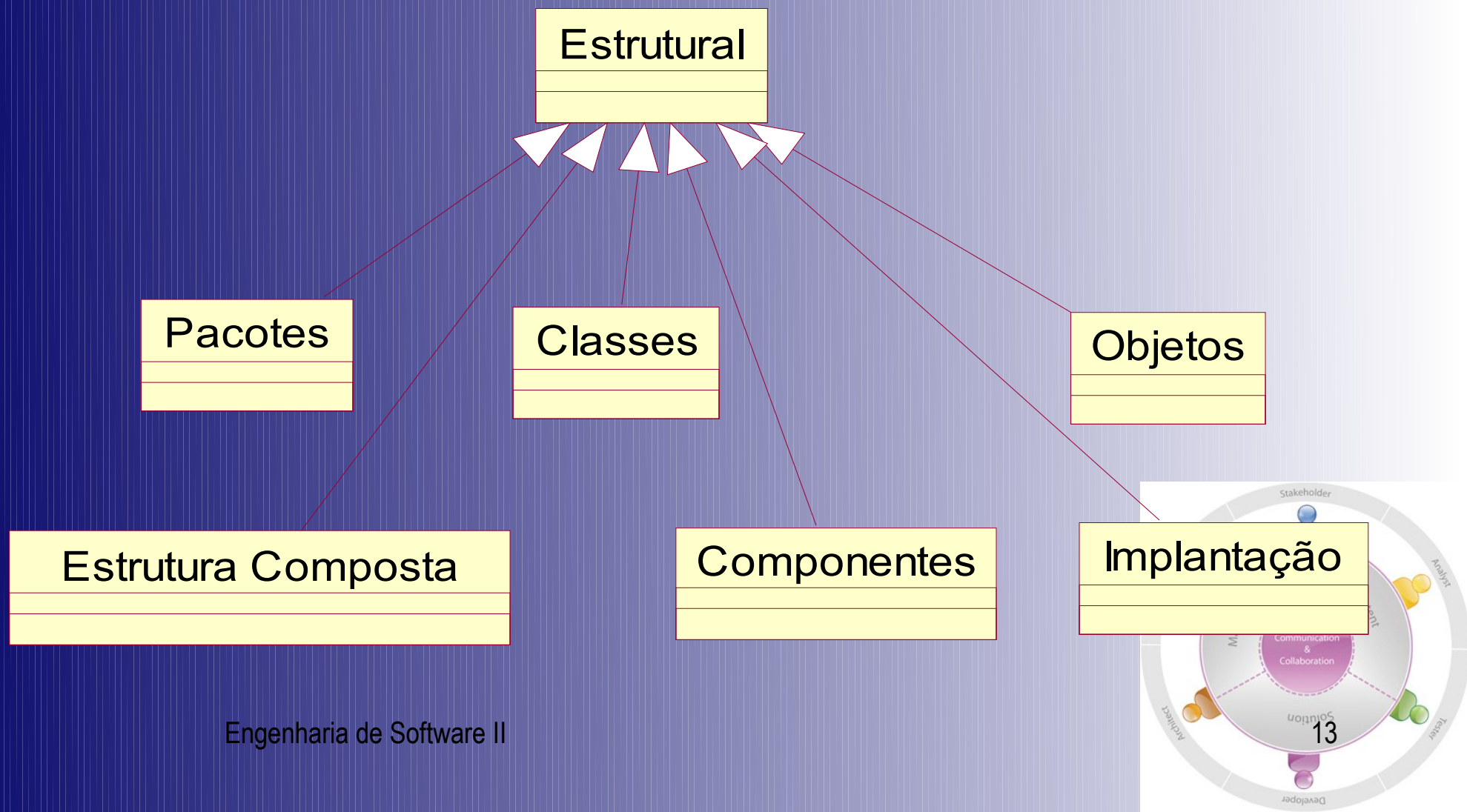
1. Diagramas Estruturais.
2. Diagramas Comportamentais.



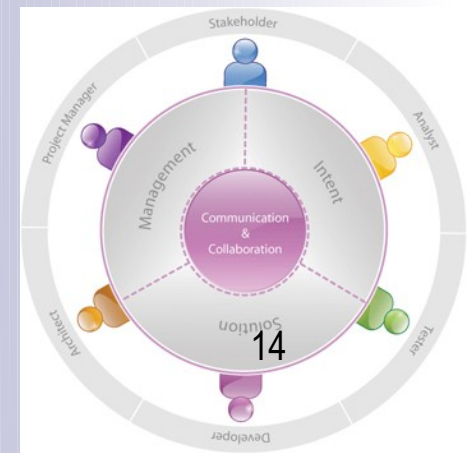
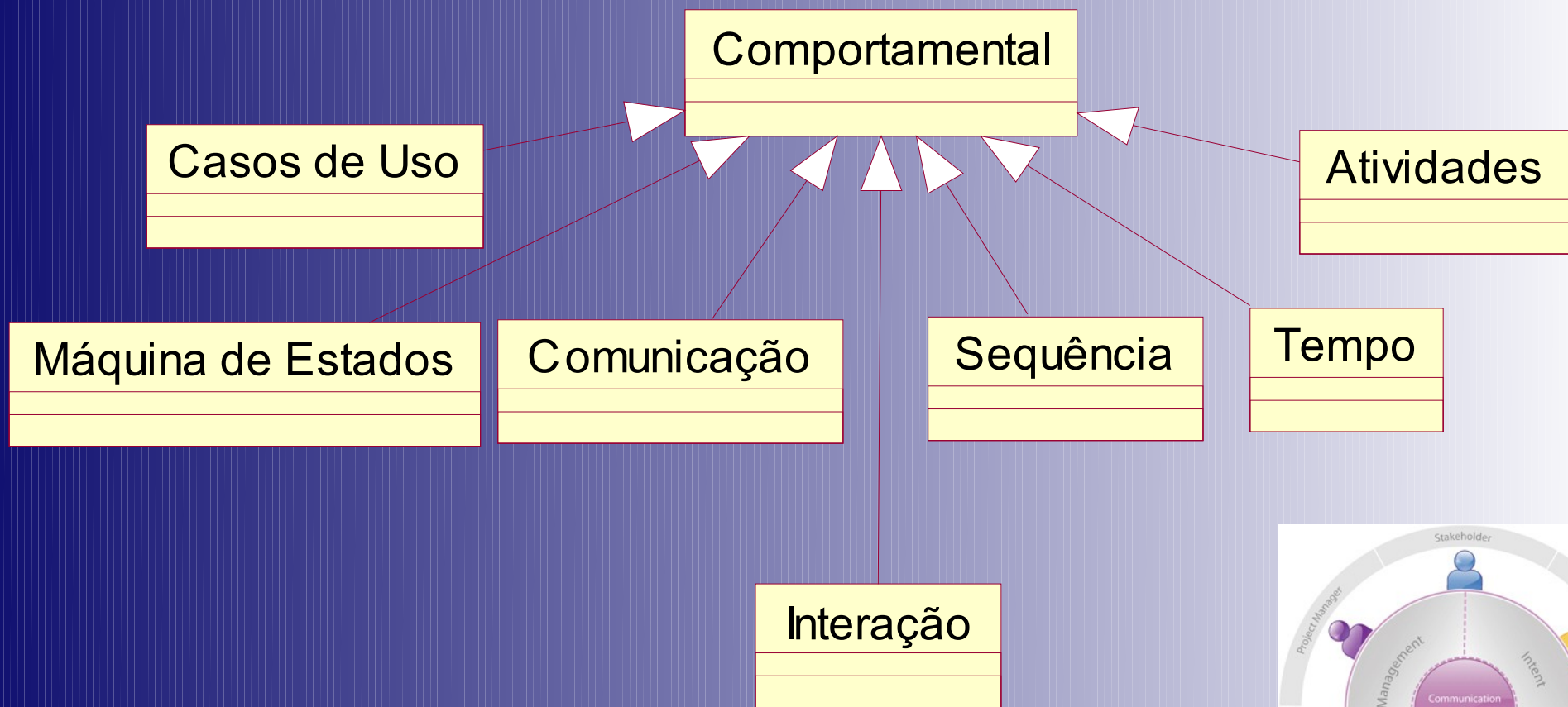
Síntese Geral dos Diagramas



Síntese Geral dos Diagramas (cont.)

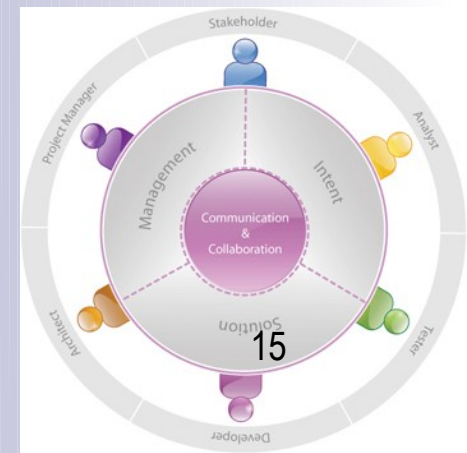


Síntese Geral dos Diagramas (cont.)



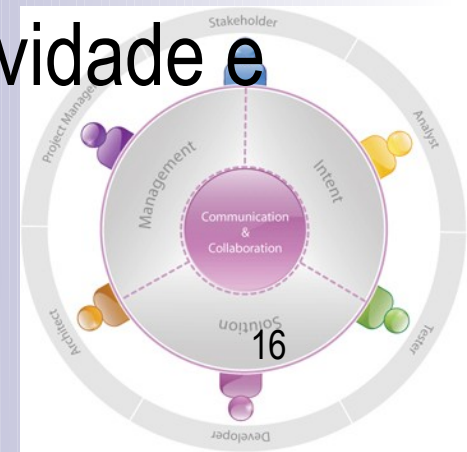
Diagramas Estruturais

Definem a arquitetura estática de um modelo. Eles são usados para modelar as coisas que compõem o modelo: classes, objetos, *interfaces* e componentes físicos. Eles são utilizados para modelar os relacionamentos e dependências entre os elementos. São eles: Diagramas de Pacotes, de Classes, de Objetos, de Estrutura Composta, de Componentes e Implantação.



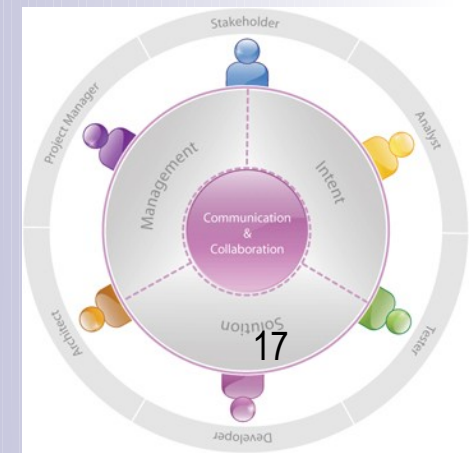
Diagramas Comportamentais

Definem diferentes aspectos do comportamento dinâmico do modelo. Eles capturam a variedade de interações e estados instantâneos dentro do modelo ao longo de sua execução. São eles: Diagramas de Casos de Uso, de Atividades, de Máquina de Estados, de Comunicação, de Seqüência, de Tempo (união dos diagramas de seqüência e estado), de Interação (união dos diagramas de atividade e seqüência).



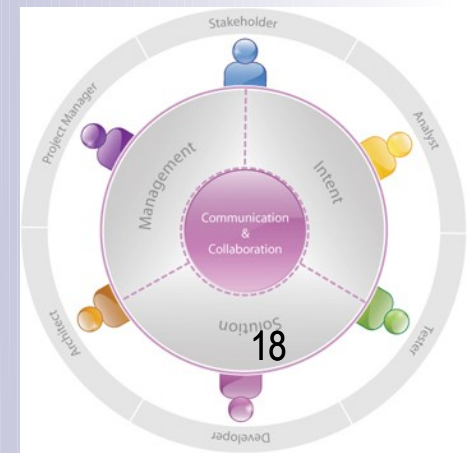
Por que tantos Diagramas?

Permite múltiplas visões do sistema. Desta forma o sistema pode ser modelado em camadas, permitindo o gerenciamento de complexidade. Imaginemos um avião composto pela sua partes elétrica, aerodinâmica, mecânica, etc. De maneira análoga cada diagrama da UML oferece uma perspectiva do sistema, estrutural e comportamental.



Motivação

- **Paradigma:** Utilização do paradigma Orientado a Objetos.
- **Notação Comum:** UML permite que analistas e desenvolvedores comuniquem-se entre si e que os programadores do banco, antes excluídos uma vez que utilizam um diferente tipo de notação, comuniquem-se com o restante do grupo.



Importante

UML não é Análise ou Projeto Orientado a Objetos ou um Método, ela é uma linguagem com notações baseadas em diagramas. ***A UML precisa estar inserida no contexto de Um Processo de Desenvolvimento.***

