



Faculdade de Ciências Exatas e Tecnologia
Ciência da Computação

Método ICONIX

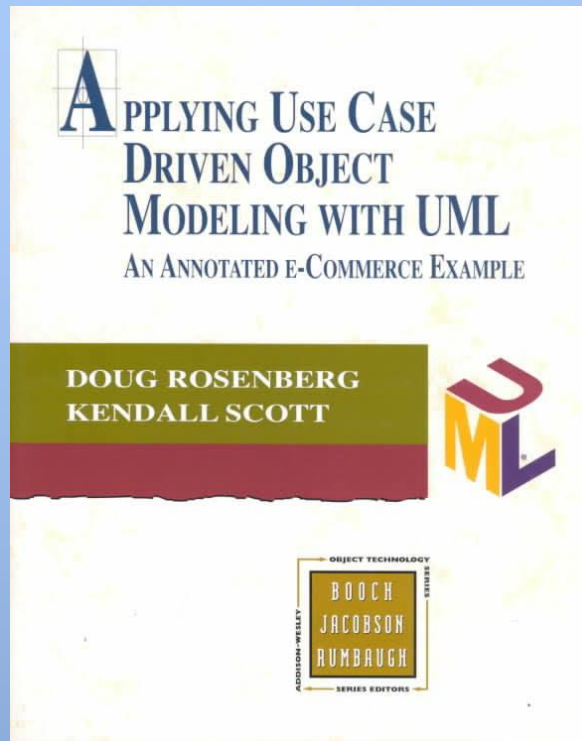
Introdução

Prof. Carlos Eduardo de Barros Paes
Departamento de Computação

ICONIX

- Síntese obtida a partir dos métodos originais de Booch/Runbaugh/Jacobson
- Produto final obtido após cerca de 10 anos de refinamentos
- Conjunto mínimo de modelos retirados do UML, no entanto os suficientes
- Método iterativo e incremental
- Dirigido por casos de uso
- Extremamente simplificado!

Referências ICONIX

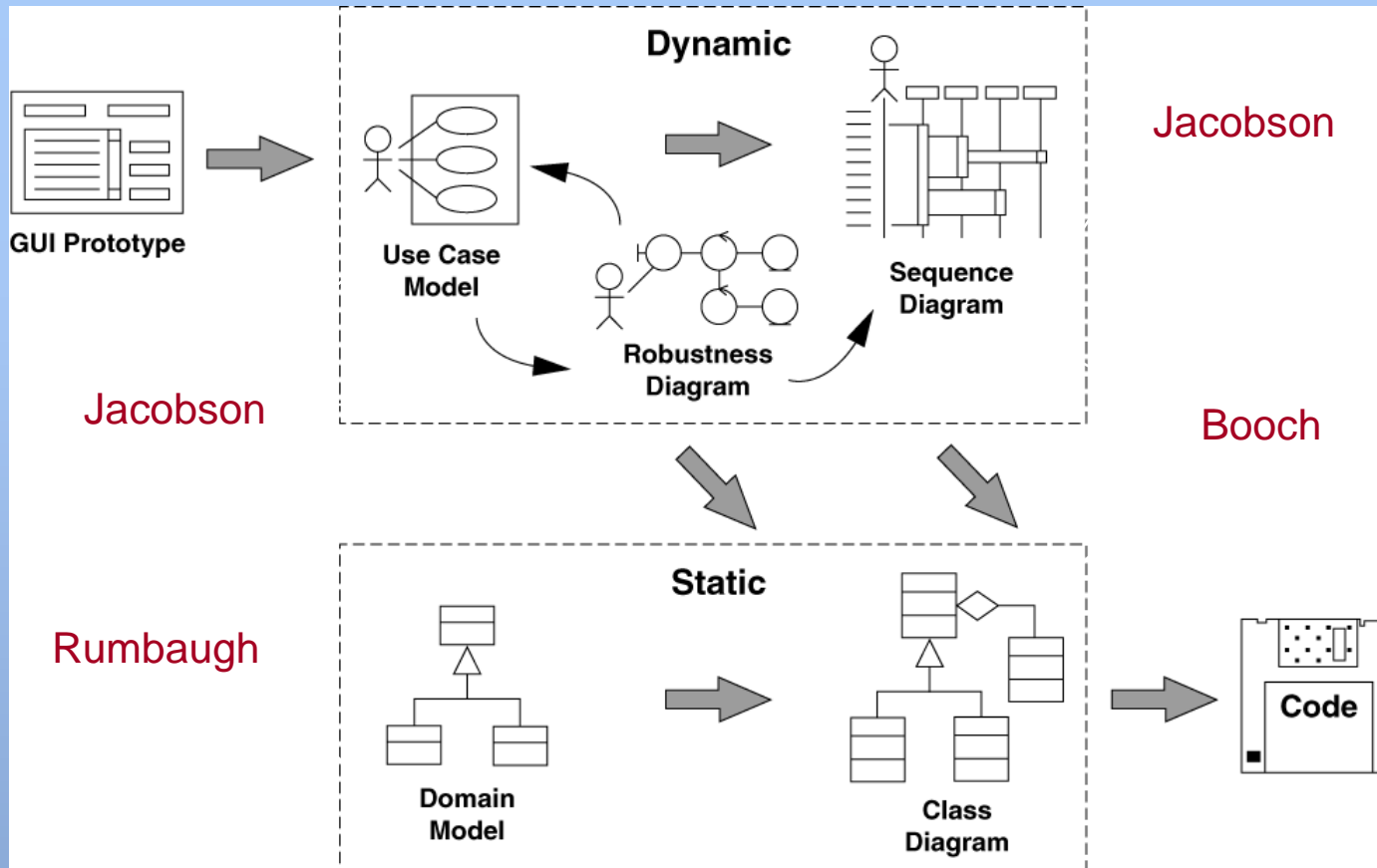


Mantido pela ICONIX Software Engineering



<http://www.iconixsw.com/>

Visão Geral do ICONIX



Visão Geral do ICONIX

- Modelos do métodos é dividido nas seguintes visões:
 - Estática
 - Dinâmica
- Principais características:
 - Iterativo e Incremental
 - Utiliza UML
 - Dirigido por Casos de Uso
 - Rastreabilidade (?)

Modelos e Diagramas

- Modelo de domínio;
- Modelo de Caso de Uso;
- Diagrama de Robustez;
- Diagrama de Sequencia;
- Digrama de Classe

Estrutura do ICONIX

- Análise de Requisitos
- Análise e Projeto preliminar
- Projeto
- Implementação

Análise de Requisitos

- Identificar use cases, objetos e desenvolver protótipos de interface com o usuário (GUI)
- Cada requisito deve focar apenas um ponto, ser claro e conciso

Análise de Requisitos

- Como iniciar??
 - Levantamento de Documentos
 - Entrevistas e questionários
 - Seminários...

Análise e Projeto Preliminar

- Descrever use cases por meio de cenários e diagrama de classes
- Necessário realizar um diagrama de robustez para cada um dos casos de uso
- Tipo de operação é o que permite ilustrar graficamente as interações entre os diversos objetos participantes num determinado caso de uso
- Finalmente, deve-se efetuar uma atualização do diagrama de classes que representa a análise realizada até então

Análise e Projeto Preliminar

- Diagrama de Robustez → Resultado da Análise OO
- Fornece uma visão dos objetos que participam de um cenário (descrito no caso de uso) e com esses objetos interagem entre si
- Estereótipos de classes propostos por Jacobson:
 - Classe Fronteira (Boundary)
 - Classe Entidade (Entity)
 - Classe Controle (Control)

Estereótipos das Classes

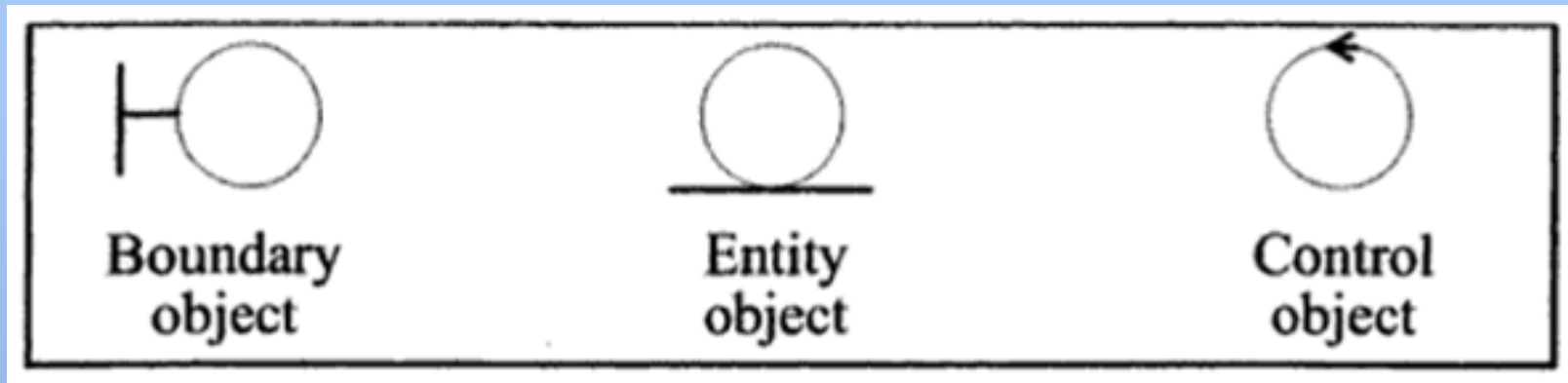
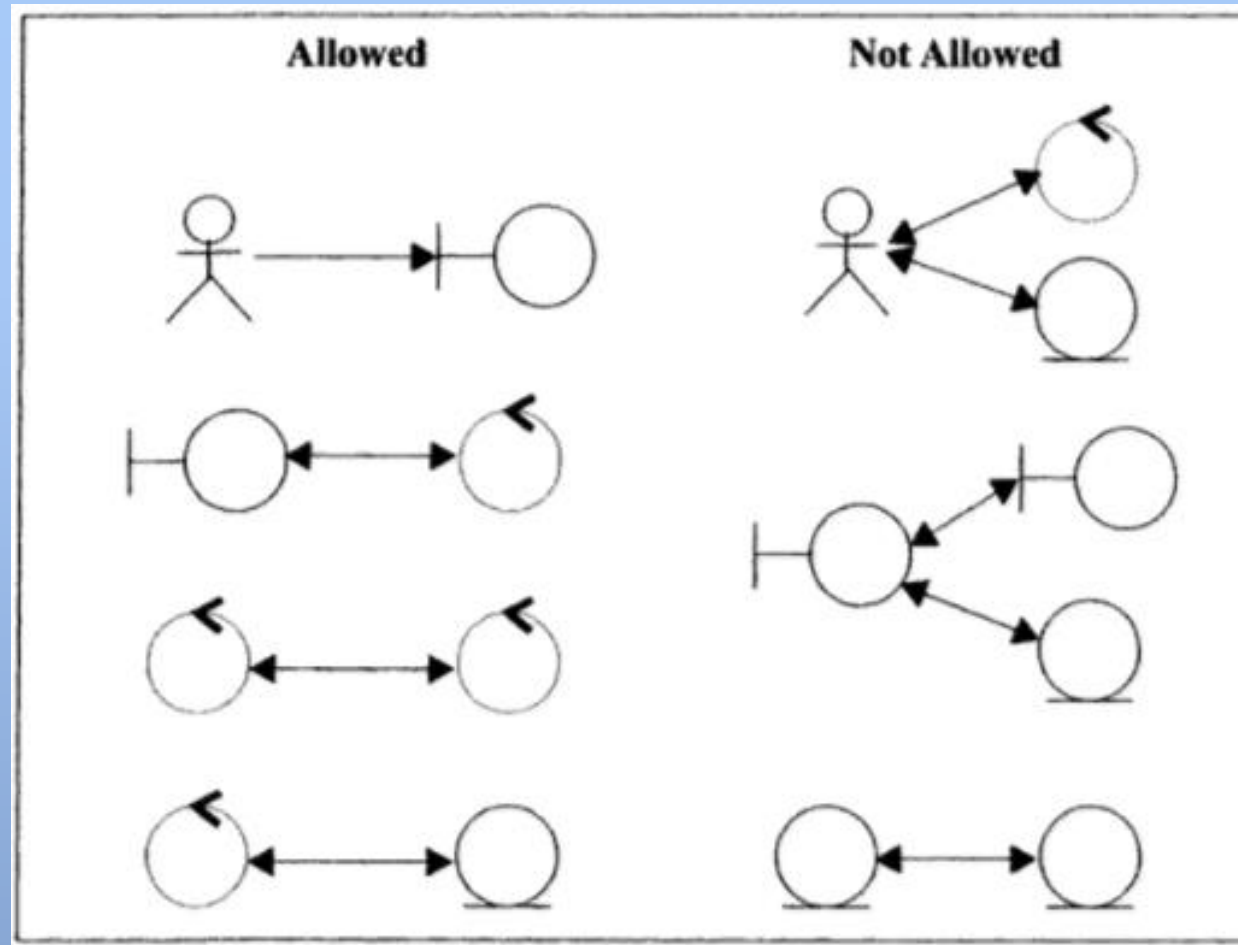
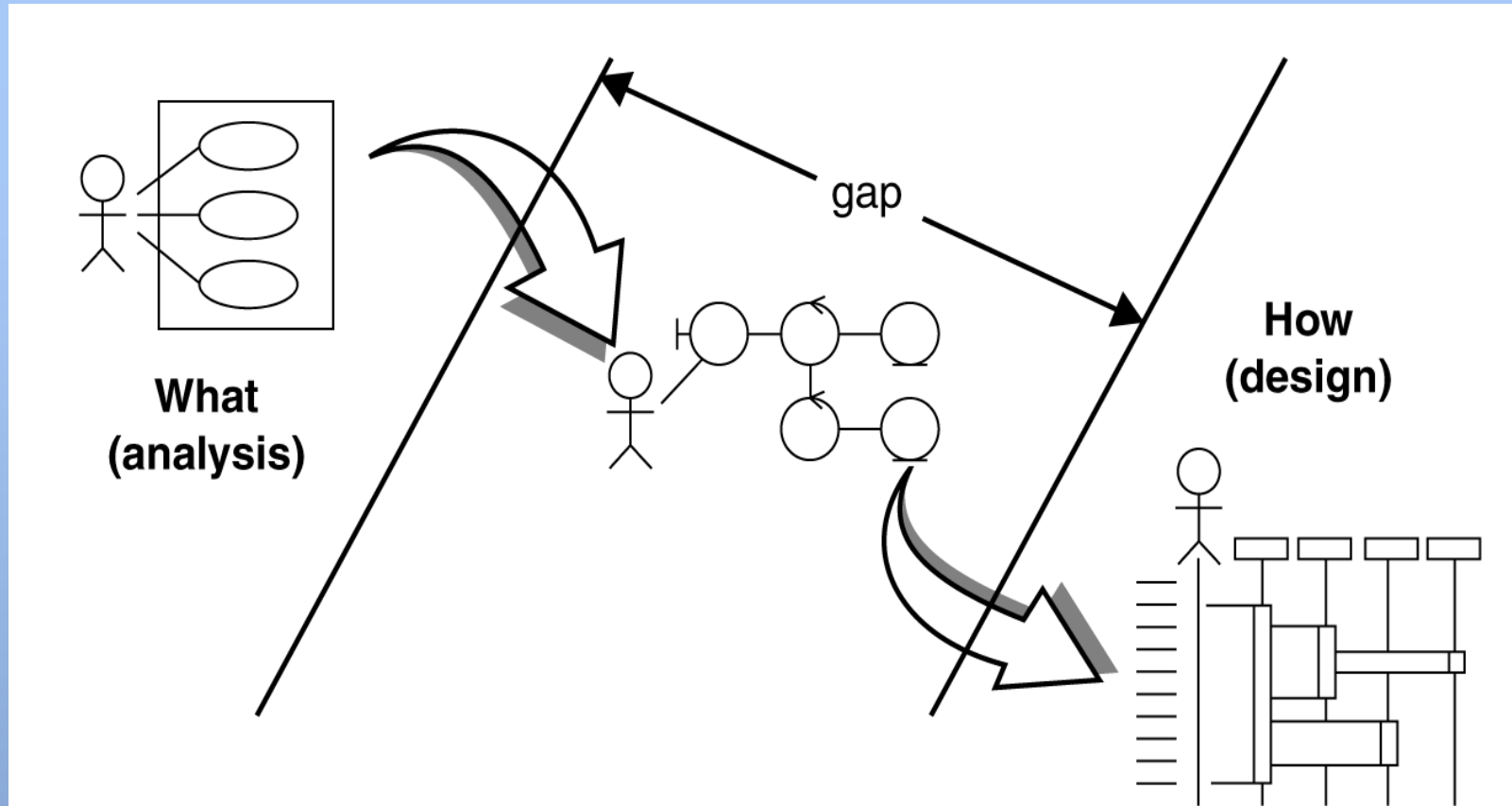


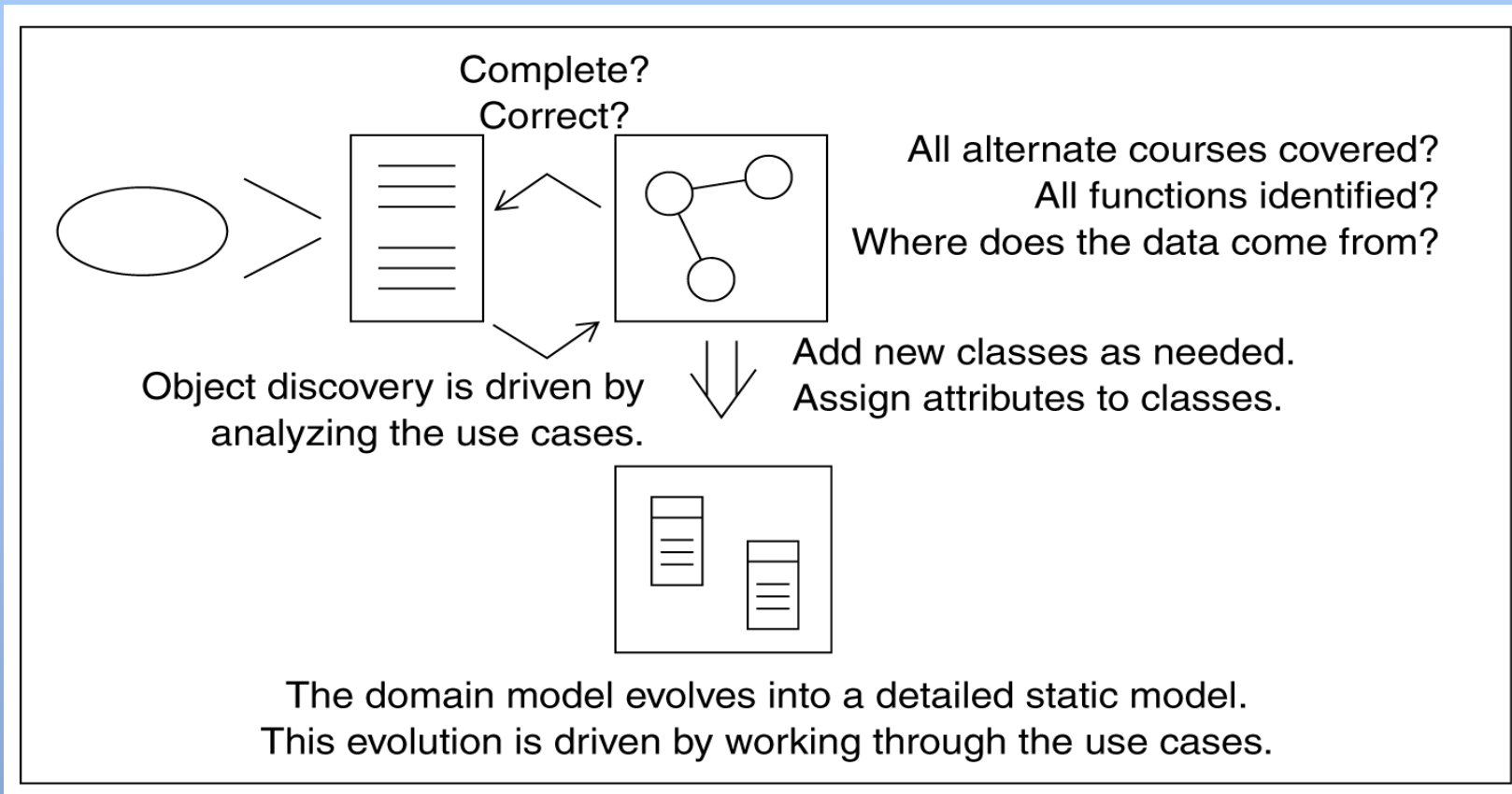
Diagrama de Robustez - Regras



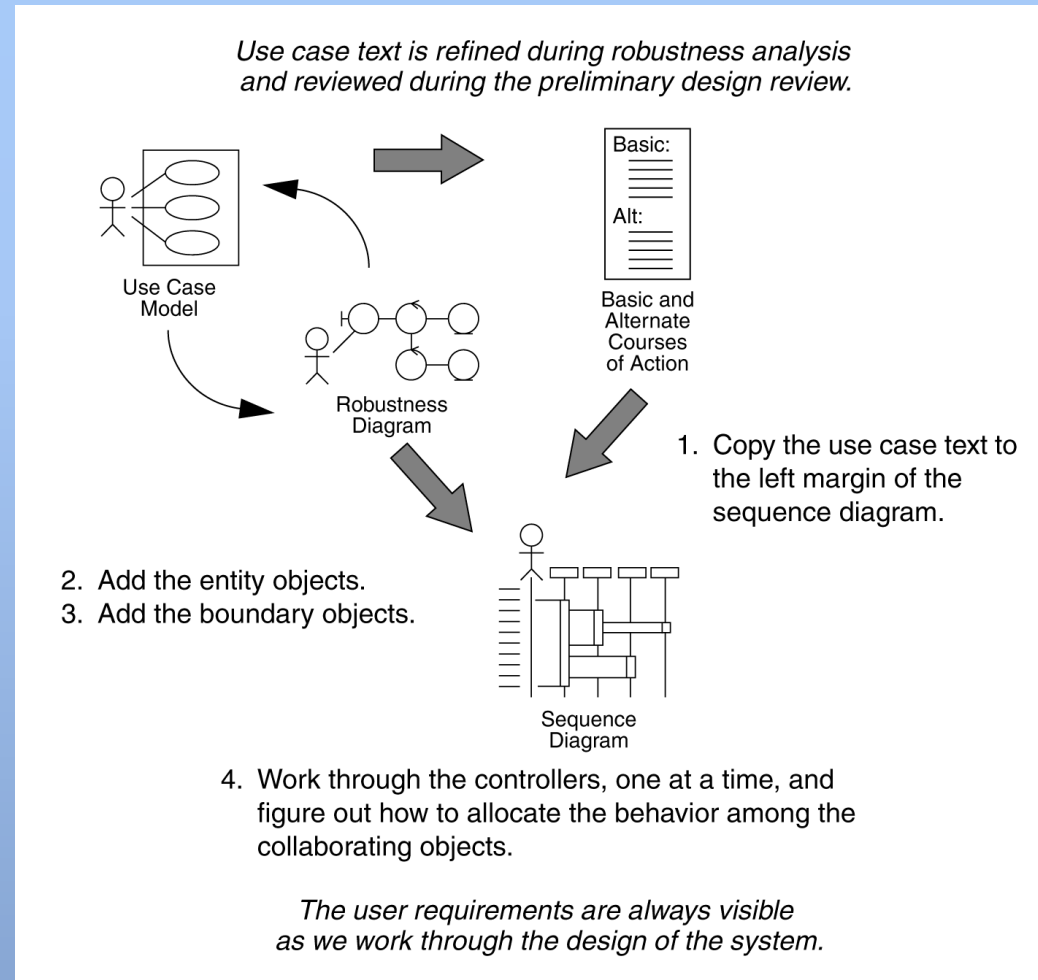
Análise e Projeto Preliminar



Análise de Robustez



Análise de Robustez



Projeto

- Especificar o comportamento, refinar o diagrama de classes
- Necessário efetuar uma especificação do comportamento do sistema dirigido pelos casos de uso

Projeto

- Especificar o comportamento, refinar o diagrama de classes
- Necessário efetuar uma especificação do comportamento do sistema dirigido pelos casos de uso
- Finalizar o modelo estático, adicionando a informação encontrada, o mais detalhadamente possível, sobre o projeto, relativamente à visibilidade de cada uma das classes e os respectivos padrões de projeto

Implementação (Codificação)

- Implementação propriamente dita a partir dos projeto detalhado
- Elaboração do testes
- Etc.

Considerações

- O método se concentra apenas nas atividades de requisitos (casos de uso) e análise/projeto OO
- Descreve a dinâmicas entre essas atividades durante o desenvolvimento do software
- Não apresenta aspectos de gerenciamento do projeto
- Extremamente simplificado → ausência de práticas importantes (ex: CCM, GP, Testes, Arquitetura e etc)