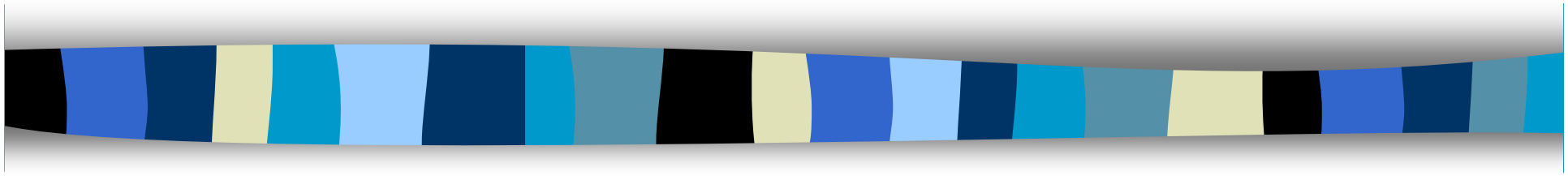
Processo Unificado: Visão Geral



Karin Becker

Instituto de Informática - UFRGS

Processo Unificado

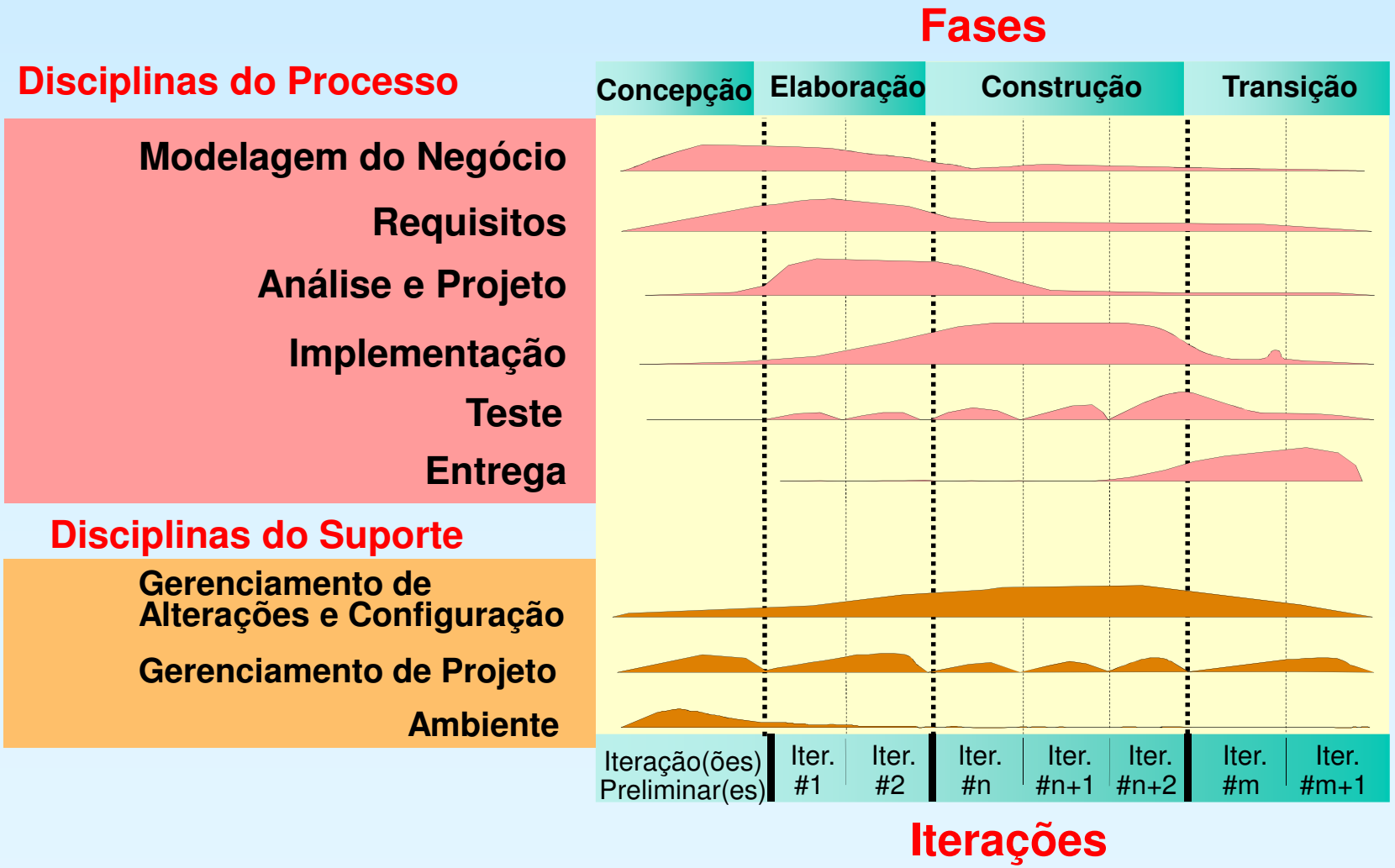
- Modelo prescritivo
  - Conjunto de Atividades bem definido
  - responsáveis
  - artefatos de entrada e saída
    - Documentos UML
  - dependência e ordem de execução entre atividades
  - ciclo de vida completo
  - Configurável
- Ferramentas
- RUP: Versão do Processo Unificado da Rational/IBM
- Origem: Objectory (Ivar Jacobson)

Processo Unificado

- descrito a partir de três perspectivas:
  - dinâmica : mostra as fases, iterações e marços ao longo do tempo
  - estática : como o processo é descrito em termos de componentes, disciplinas, atividades, fluxos de trabalho, artefatos e papéis do processo
  - prática : sugere boas práticas para execução das atividades
- Para um guia completo de RUP

  http://www.wthreex.com/rup/portugues/index.htm

O Processo Unificado



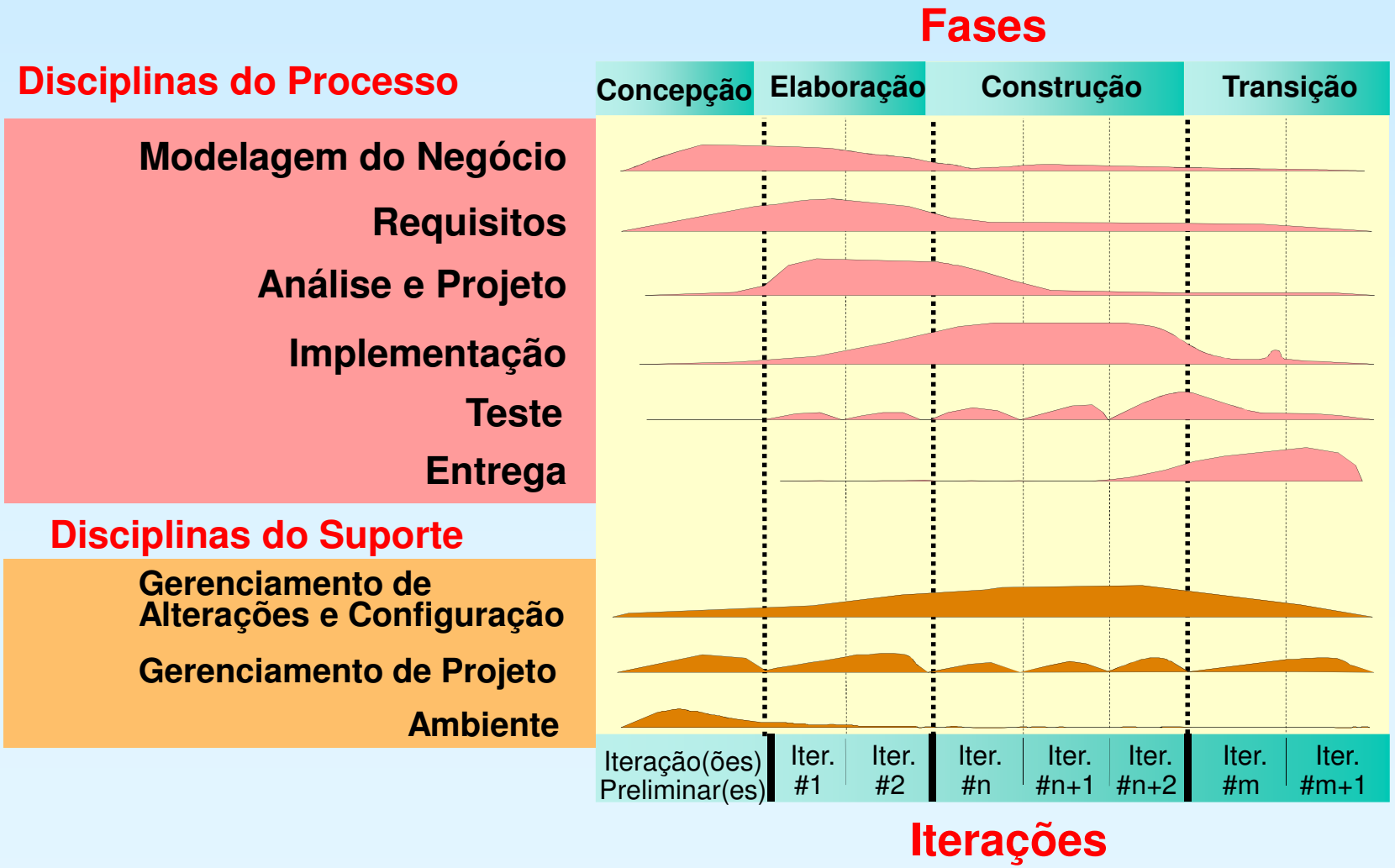
Perspectiva Dinâmica : Fases

- **Concepção**: estabelece o caso de negócio para o sistema e delimita o escopo do projeto.
- **Elaboração**: envolve o planejamento das atividades e respectivos recursos, análise do domínio do problema e a especificação da arquitetura do sistema.
- **Construção**: envolve a elaboração do software a partir de arquitetura em um produto completo para utilização pelos usuários.
- **Transição**: viabiliza que o software possa ser utilizado pelos usuários.

Perspectiva Dinâmica

- Fases
  - Diferentes momentos do projeto, com diferentes níveis de especificação, risco, planejamento e tipo de disciplina predominante
- Iterações
  - Cada iteração gera release(s) de vários artefatos que, juntos, constituem a “baseline”
  - Baseline é o conjunto de artefatos revisados e aprovados
    - Fornecem uma base consolidada para a continuação
    - somente pode ser alterada através de um procedimento formal
- Incremental
  - Software (código e outros tipos de artefatos)

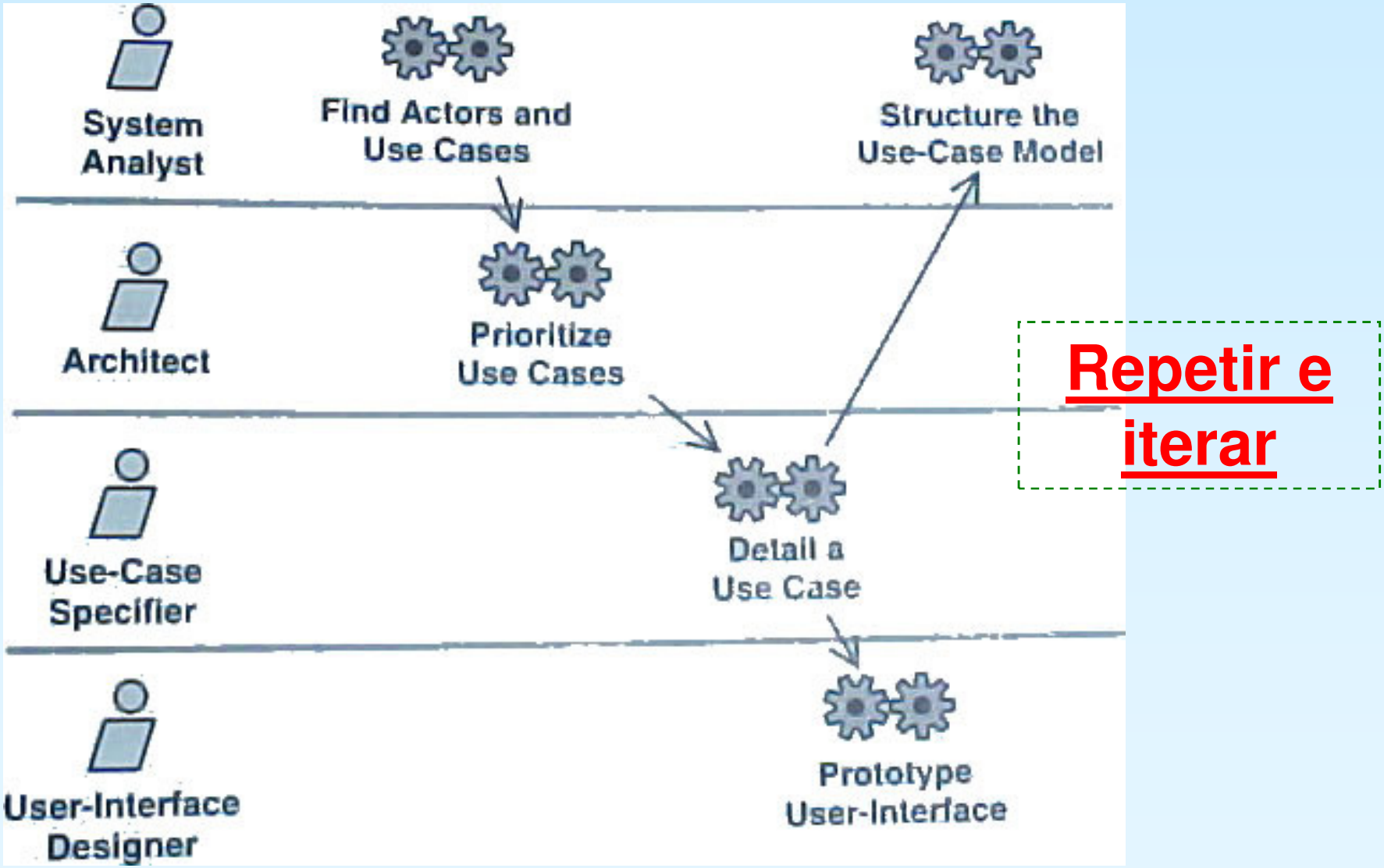
O Processo Unificado



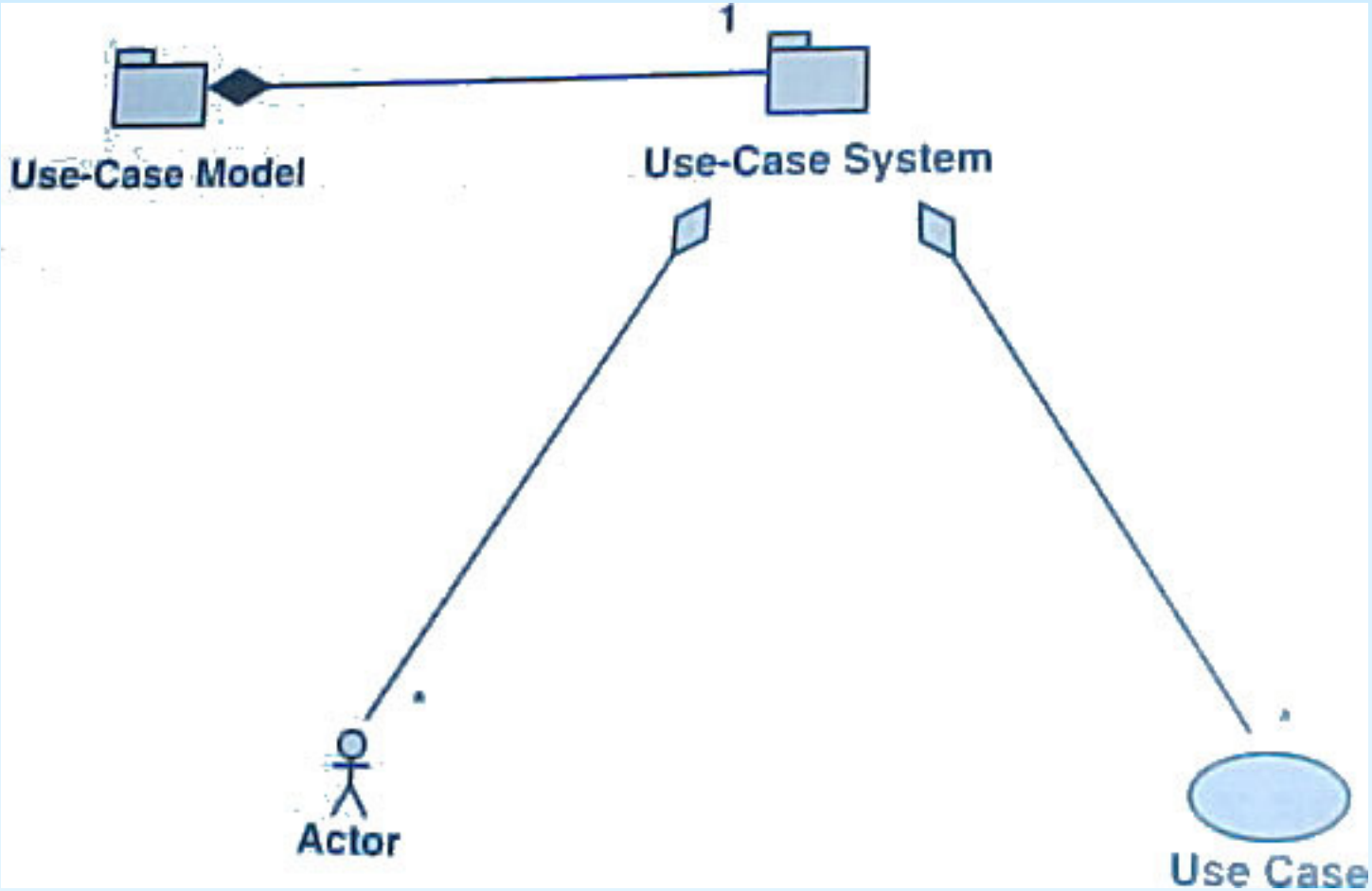
Perspectiva Estática

- Disciplinas
  - Pessoas (papéis): quem?
  - Atividades: como?
  - Artefatos: o quê?
  - Workflow: coordenação dos elementos precedentes no tempo.

Exemplo: Workflow Requisitos (parcial)

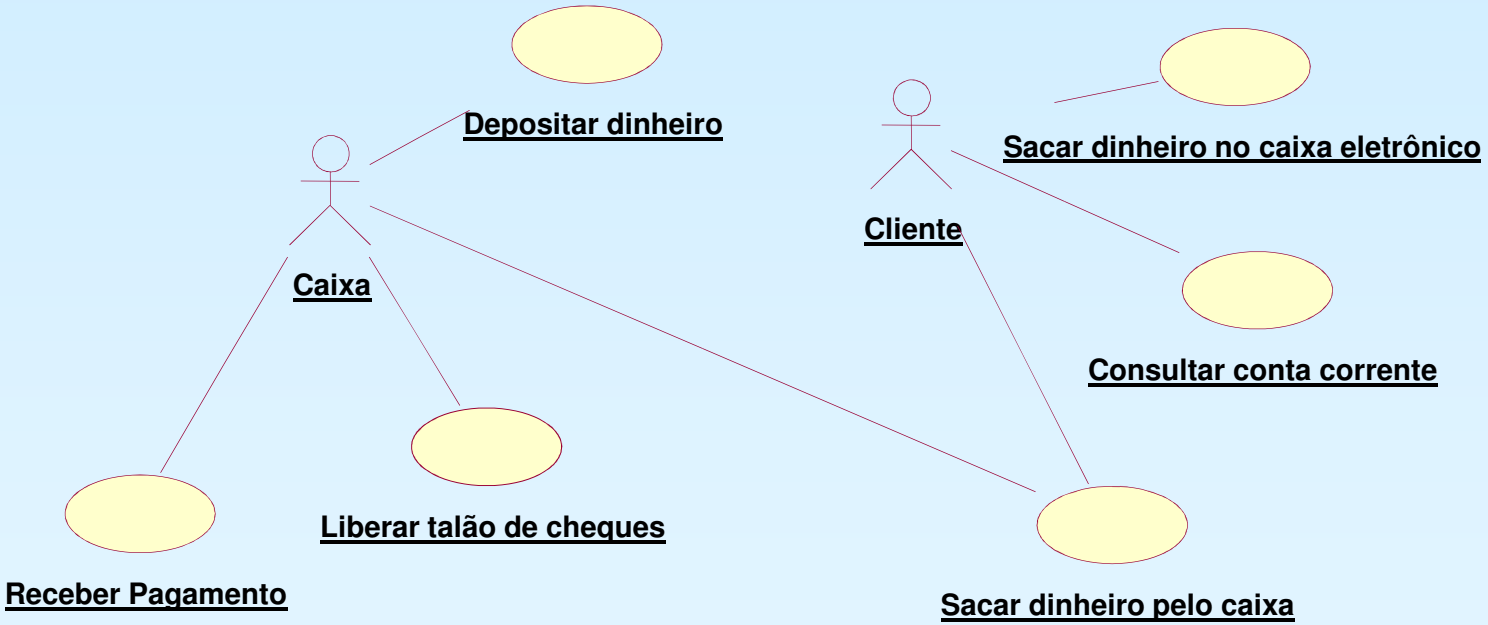


Workflow Requisitos: Artefatos



Exemplo: Modelo de Caso de Uso

- Sistema Bancário (incompleto)



Perspectiva Estática (Disciplina)

Tabela 4.1 Workflows estáticos no Rational Unified Process.

Workflow	Descrição
Modelagem de negócios	Os processos de negócios são modelados usando casos de uso de negócios.
Requisitos	Os agentes que interagem com o sistema são identificados e os casos de uso são desenvolvidos para modelar os requisitos de sistema.
Análise e projeto	Um modelo de projeto é criado e documentado usando modelos de arquitetura, modelos de componente, modelos de objeto e modelos de seqüência.
Implementação	Os componentes de sistema são implementados e estruturados em subsistemas de implementação. A geração automática de código com base nos modelos de projeto ajuda a acelerar esse processo.
Teste	O teste é um processo iterativo realizado em conjunto com a implementação. O teste de sistema segue o término da implementação.
Implantação	Uma versão do produto é criada, distribuída aos usuários e instalada no local de trabalho.
Gerenciamento de configuração e mudanças	Este workflow de apoio gerencia as mudanças do sistema (veja o Capítulo 29).
Gerenciamento de projetos	Este workflow de apoio gerencia o desenvolvimento do sistema (veja o Capítulo 5).
Ambiente	Este workflow está relacionado à disponibilização de ferramentas apropriadas de software para a equipe de desenvolvimento.

Sommervile

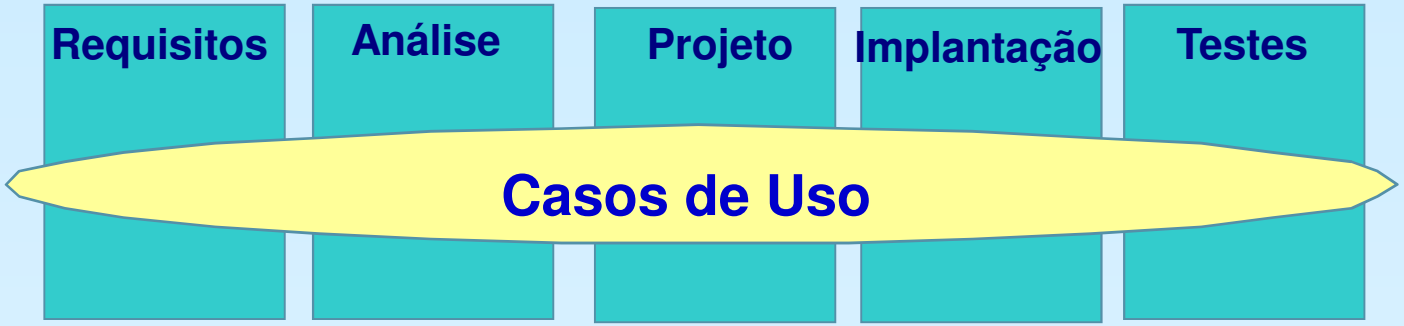
Para saber mais

- http://www.wthreex.com/rup/portugues/index.htm
- O jogo do RUP
  - http://www.ruppers.com.br/overview/

Processo Unificado : Características

- Utiliza a UML (Unified Modeling Language)
- Dirigido por casos de uso
- Centrado na arquitetura
- Baseado em componentes
  - Componentes de software interconectados por uma interface bem definida

Orientado a Casos de Uso

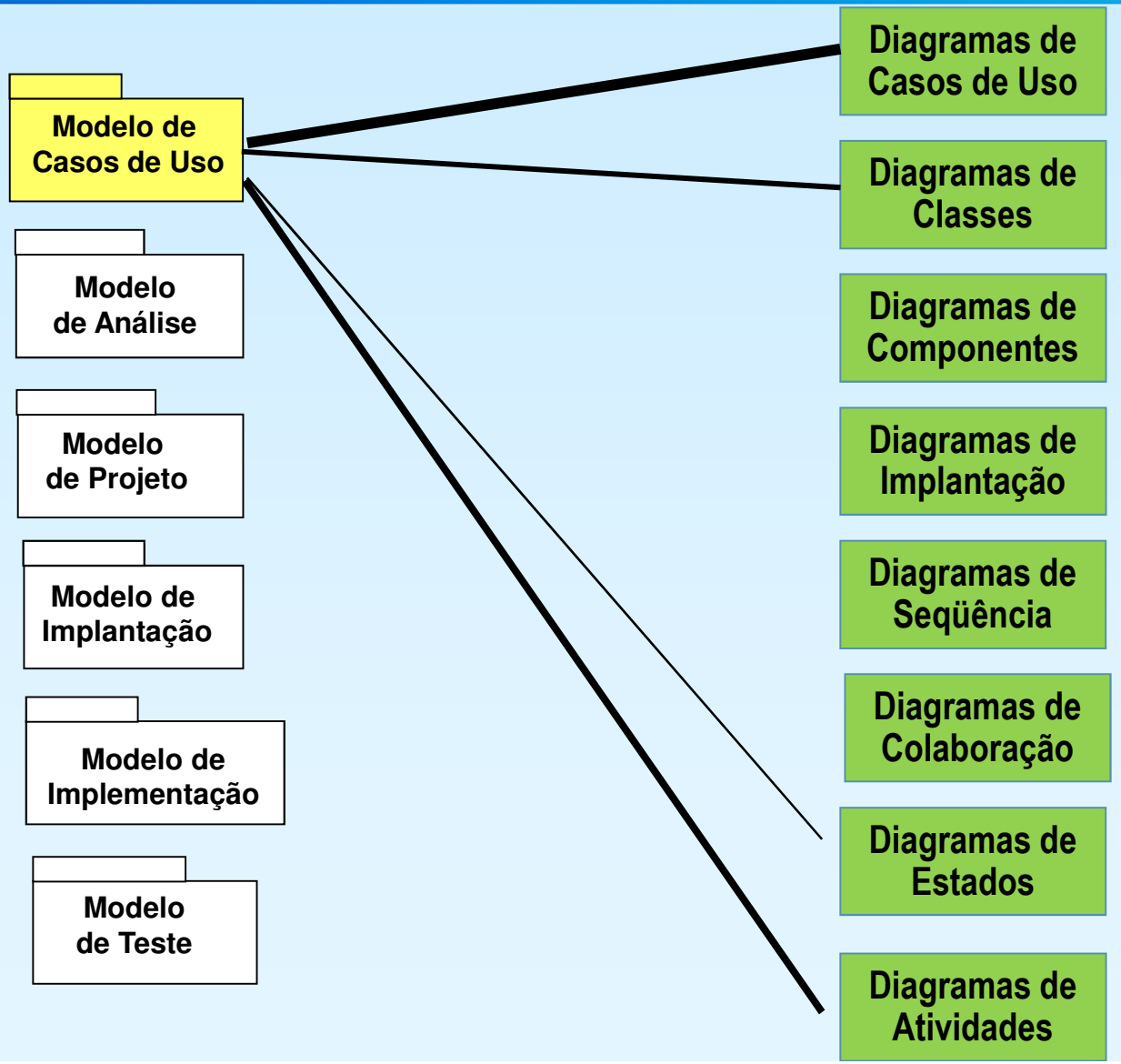


- Caso de uso: funcionalidade completa como externamente percebida pelo usuário
- Orienta uma série de atividades de desenvolvimento:
  - Criação e validação da arquitetura do sistema
  - Definição dos casos de teste e procedimentos
  - Planejamento das iterações
  - Criação da documentação do usuário
  - Implantação do sistema

PU : Orientado a Modelos usando UML

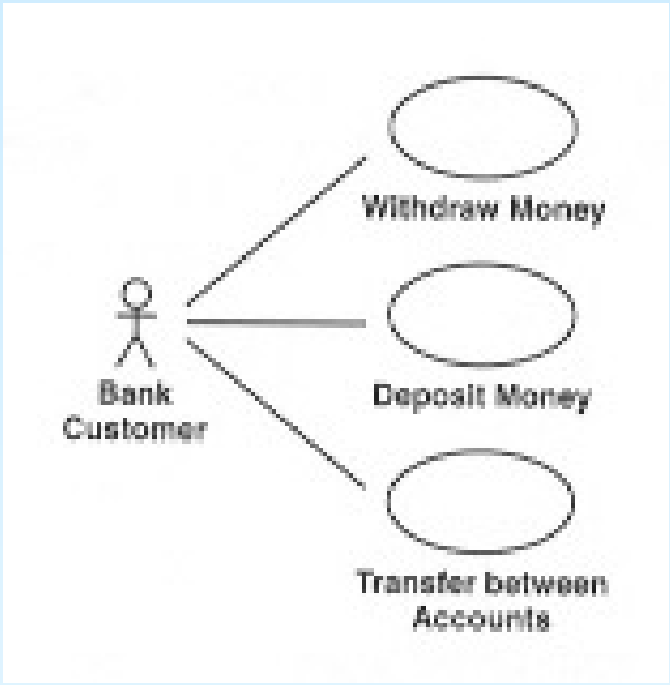
- Modelo: Uma abstração do sistema segundo um certo ponto de vista e nível de abstração
  - Ponto de vista:
    - Modelo de projeto vs. Modelo de Casos de Uso
    - Destinados a diferentes pessoas, com diferentes missões
  - Semanticamente autocontido
    - “usuário” do modelo não necessita de informação complementar para interpretá-lo
- Manter relacionamentos entre modelos
- Um modelo é descrito através de um conjunto de diagramas
  - UML, no caso do processo unificado
- Modelo ≠ Diagrama
  - Modelo é composto de um ou mais diagramas, possivelmente de tipos diferentes

Diagrama vs. Modelo



Breve tour pelo PU

Requisitos: o começo de tudo !!



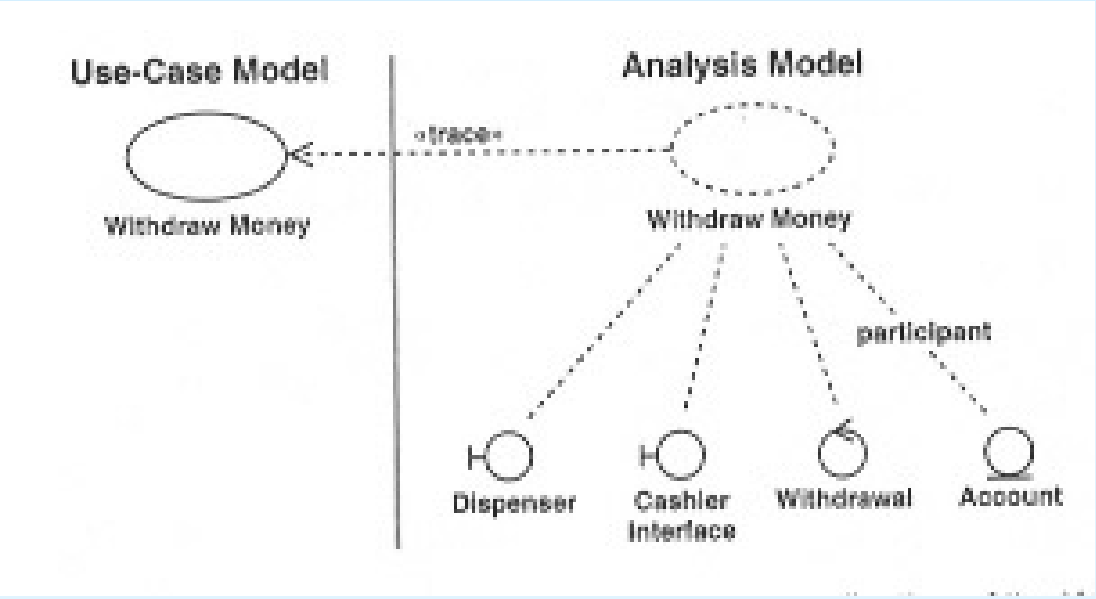
Modelo de Casos de Uso

Breve tour pelo PU

Iniciando a Solução...

"Temos que identificar em nossos requisitos, quais são os elementos essenciais para satisfazê-los..."

Modelo de Análise

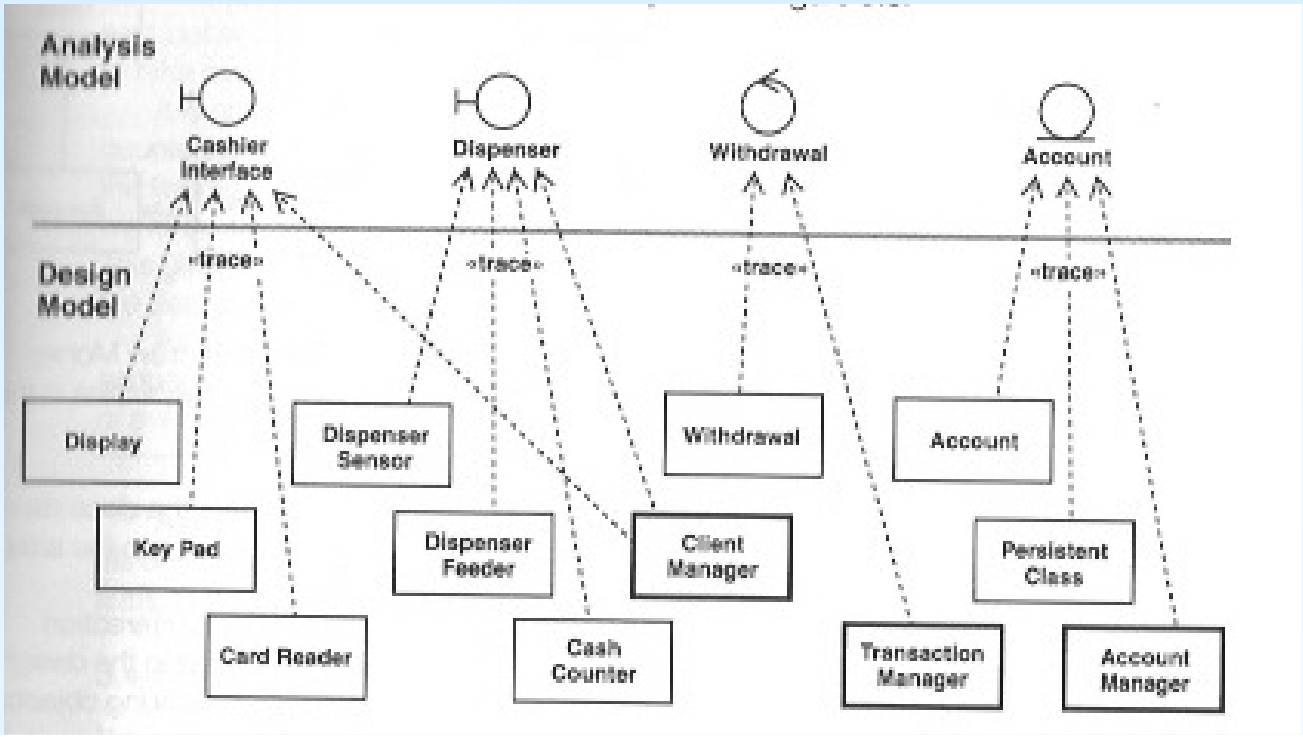


Breve tour pelo PU

Sedimentando a Solução...

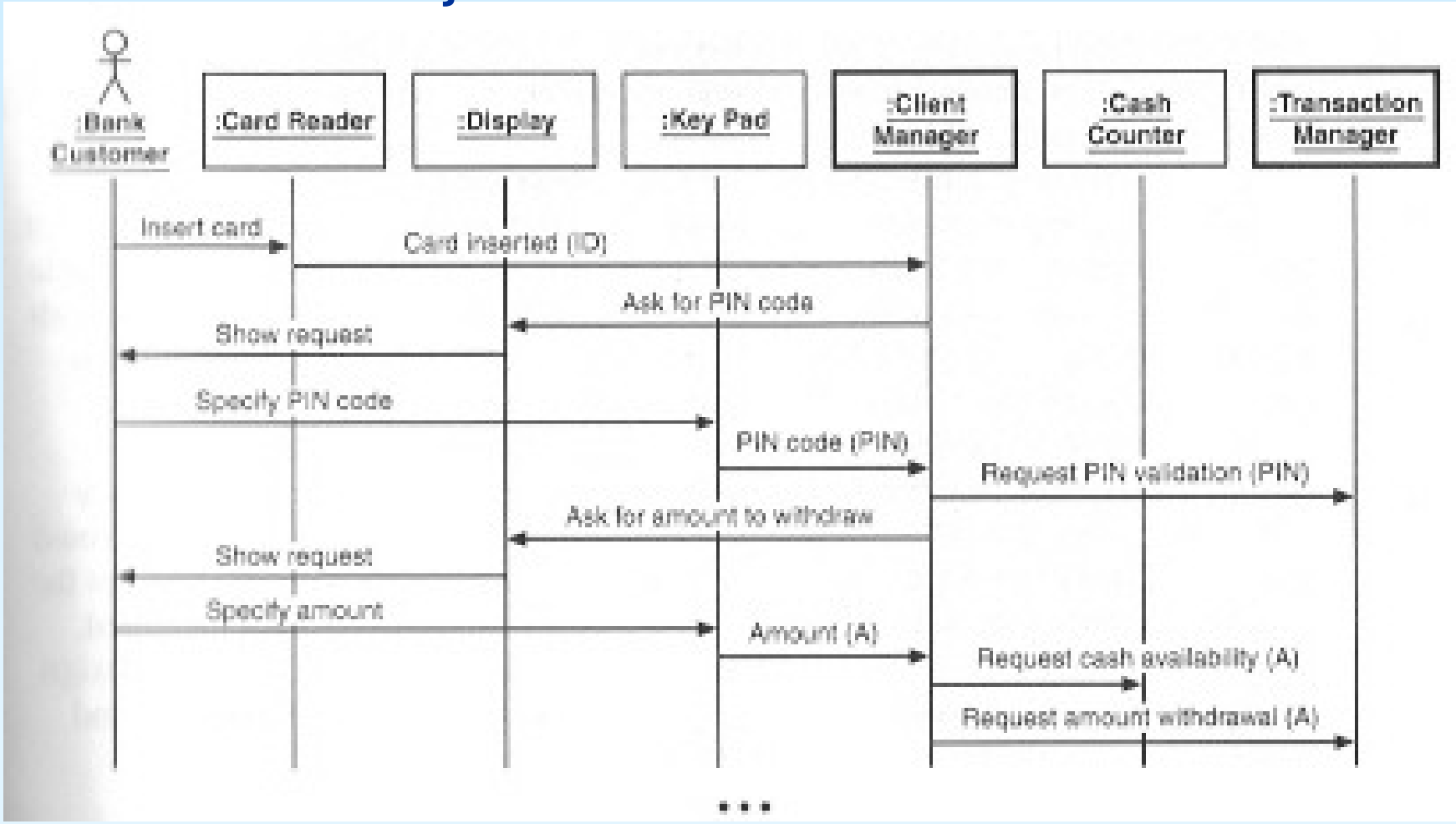
"A partir dos elementos essenciais, precisamos definir estratégias para satisfazê-los incluindo suas restrições..."

Modelo de Projeto



Breve tour pelo PU

Modelo de Projeto



Breve tour pelo PU

Modelo de Projeto: Garantindo a qualidade ...

```
public class Account {
  private int balance;
  /*@ invariant balance>=0 @*/
  ...
  void debit(int amount) {
   /*@ requires amount <= balance @*/
   /*@ ensures balance = \old(balance) - amount @*/
  }
...
}
```

Breve tour pelo PU

"Com a solução definida, o próximo passo é operacionalizá-la"

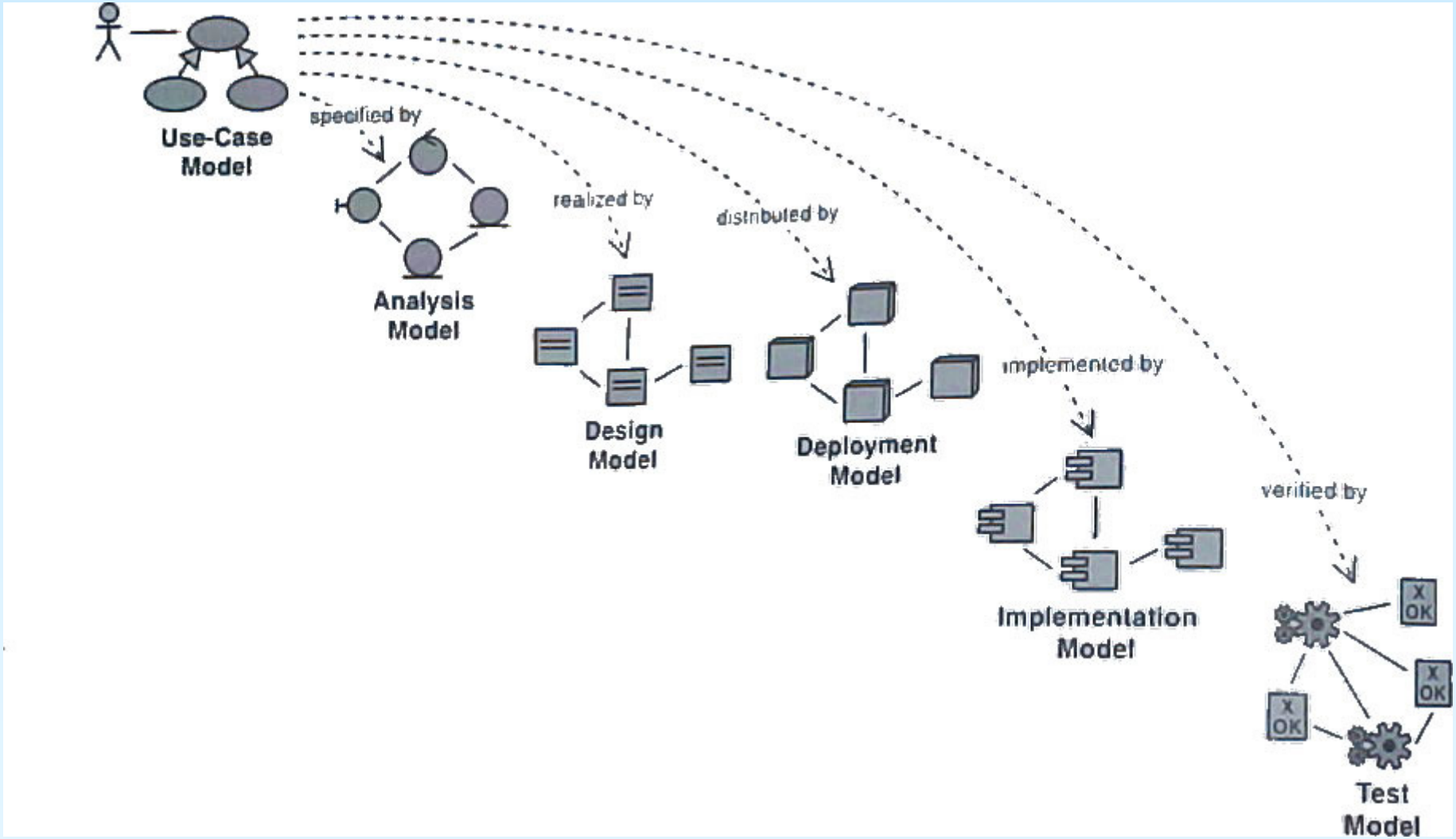
```
public class Account {
  private int balance;
  ...
  void debit(int amount) {
    if(amount<=balance)
       balance = balance – amount;
    else throw new AccountException("...");
  }
...
}
```

Breve tour pelo PU

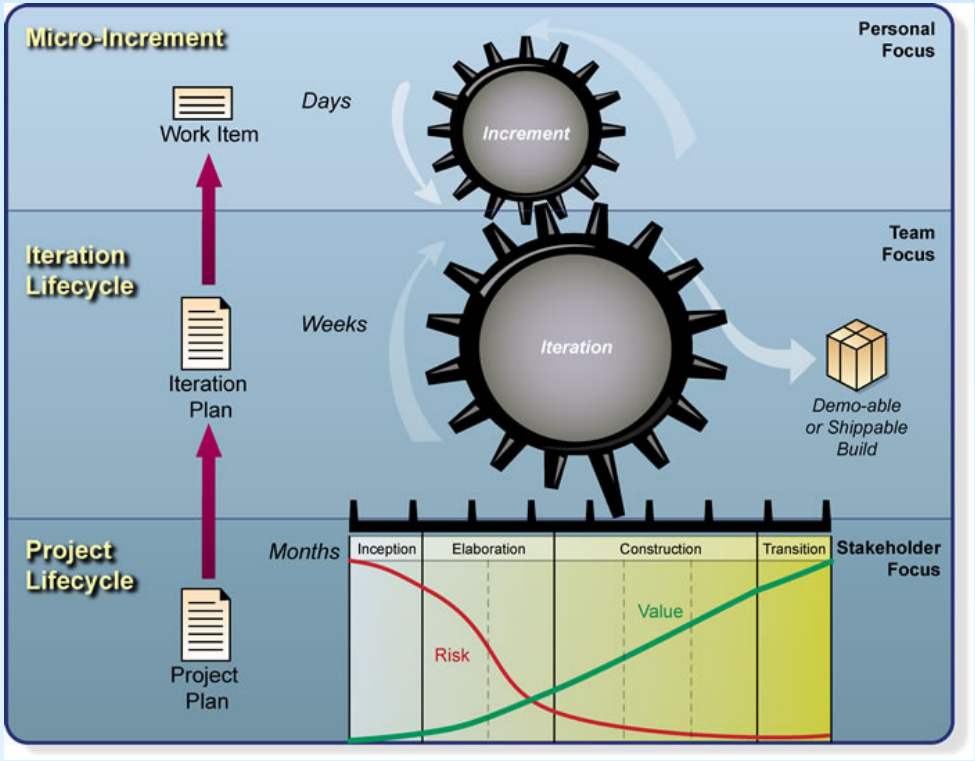
"Funciona?? Com a implementação feita, podemos então executar os testes !!!"

```
public class AccountTest extends TestCase {

  void testDebit() {
    Account acc = new Account(10);
    acc.debit(10);
    assertEquals(0, acc.getBalance());
  }

}
```

PU: Relacionamentos entre Modelos



Rup ficando “ágil” : Open UP



- Parte do Eclipse Process Framework
- Fundamentada por valores ágeis
  - iterativa
  - incremental (software)
  - colaboração entre desenvolvedores e clientes
  - requisitos gerenciados de forma a agregar valor ao cliente
  - centrada na arquitetura como forma de aumentar qualidade e compreensão técnica

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

http://www.ibm.com/developerworks/br/rational/local/open_up/index.html