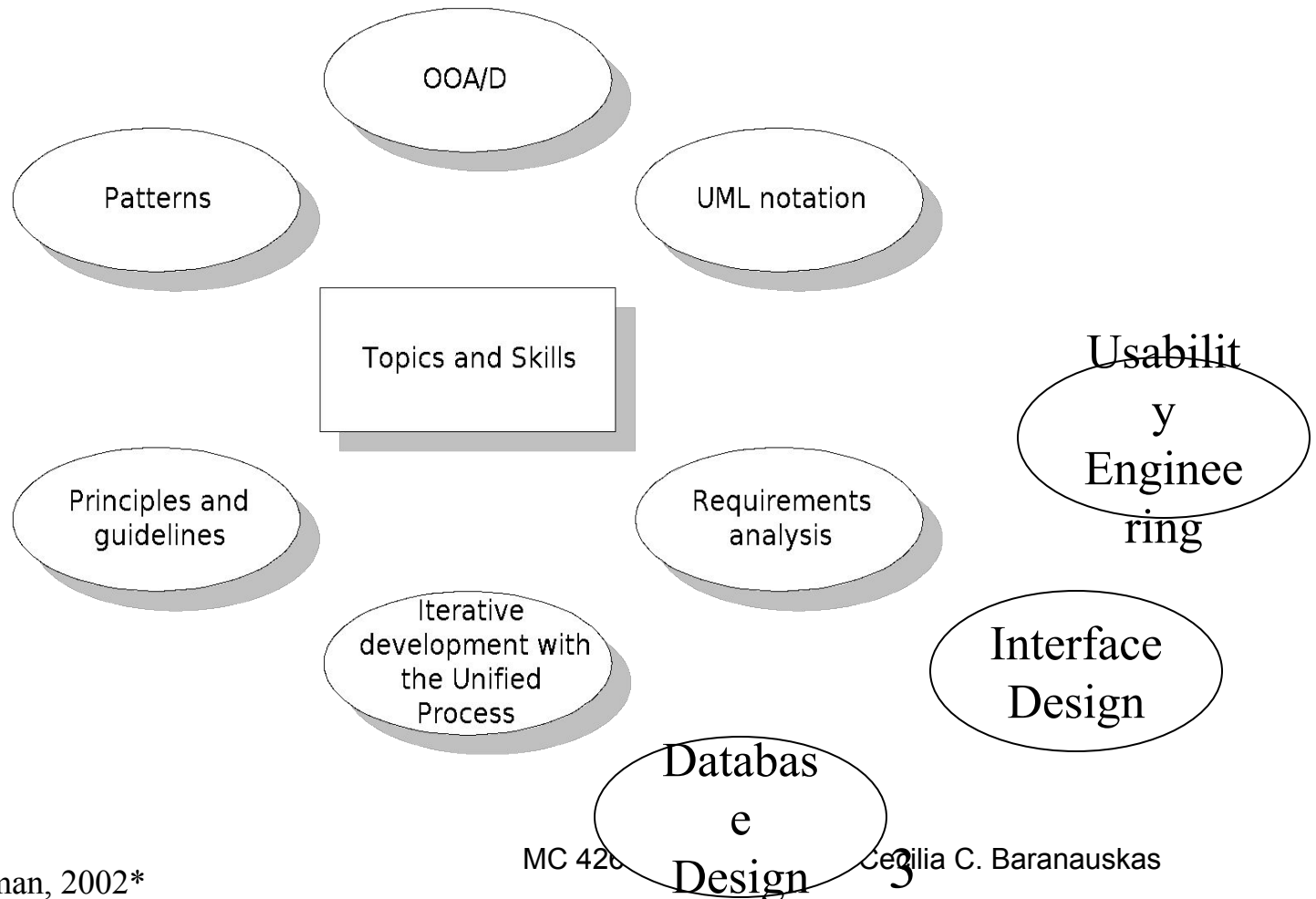

Desenvolvimento Iterativo e o Processo Unificado

Análise e Design Orientados a Objeto OOA/D

- Aplicando
 - Notação UML, a *Unified Modelling Language*
 - *Design Patterns*,
 - Princípios das melhores-práticas (best-practice), heurísticas
 - O Processo Unificado (PU)
 - Processo de desenvolvimento iterativo
- Como pensar em objetos, como projetar sistemas orientados a objeto
 - Como responsabilidades podem ser alocadas a classes de objetos? Como os objetos devem interagir? Que classes devem fazer o quê?

Tópicos envolvidos



O que é Análise e O que é Design?

- **Análise**: uma investigação do problema e requisitos (não sua solução)
 - Análise de Requisitos: uma investigação dos requisitos
 - Análise de Objetos: uma investigação dos objetos do domínio
- **Design**: uma solução conceitual que dá conta dos requisitos (não sua implementação)

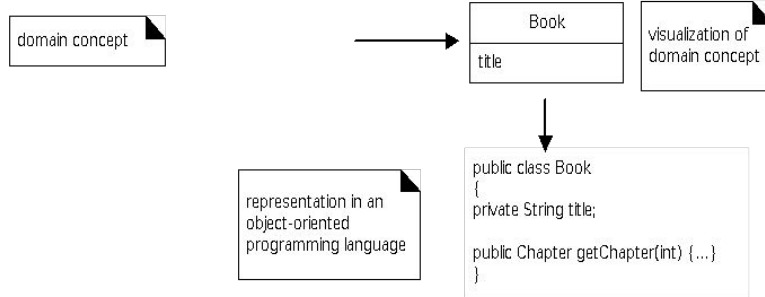
Do the right thing (analysis),

Do the thing right (design)

O que é Análise e Design Orientados a Objeto?

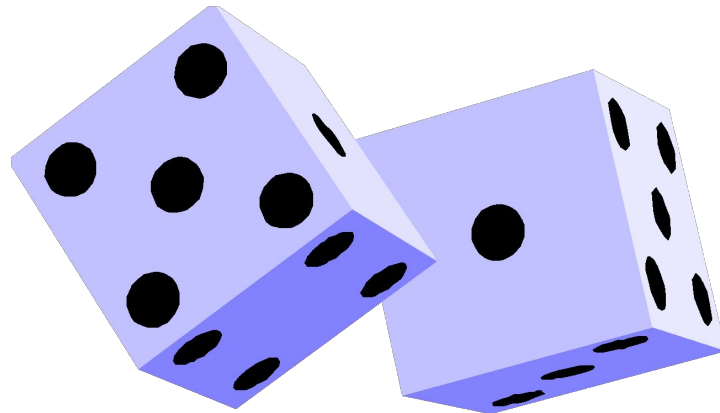
- **Análise Orientada a Objetos (AOO):**
 - Encontrar e descrever os objetos – ou conceitos
 - no domínio do problema
- **Design Orientado a Objetos (DOO):**
 - Definir objetos de software e como eles colaboram para satisfazer os requisitos
- **Programação Orientada a Objetos (POO):**
 - Os objetos projetados (*designed*) são implementados

Análise e Design OO



Ex. Jogo de Dados

Um jogador joga 2 dados. Se o total for 7 ele/a ganha; caso contrário, ele/a perde



Definir Casos de Uso

- Uma descrição dos processos do domínio, escritos como casos de uso:

***Jogar o Jogo dos Dados:** Um jogador toma os dados e os joga. Se o valor das faces somar 7 ele/a ganha; caso contrário ele/a perde.*

- Casos de Uso não são artefatos OO
 - Ferramenta popular na análise de requisitos e parte importante no PU (Processo Unificado)

Definir um Modelo do Domínio

- Identificação dos conceitos, atributos e associações [ooa]
- Um Modelo de Domínio **não é** uma descrição de objetos de software
- É uma visualização de conceitos do domínio no mundo real



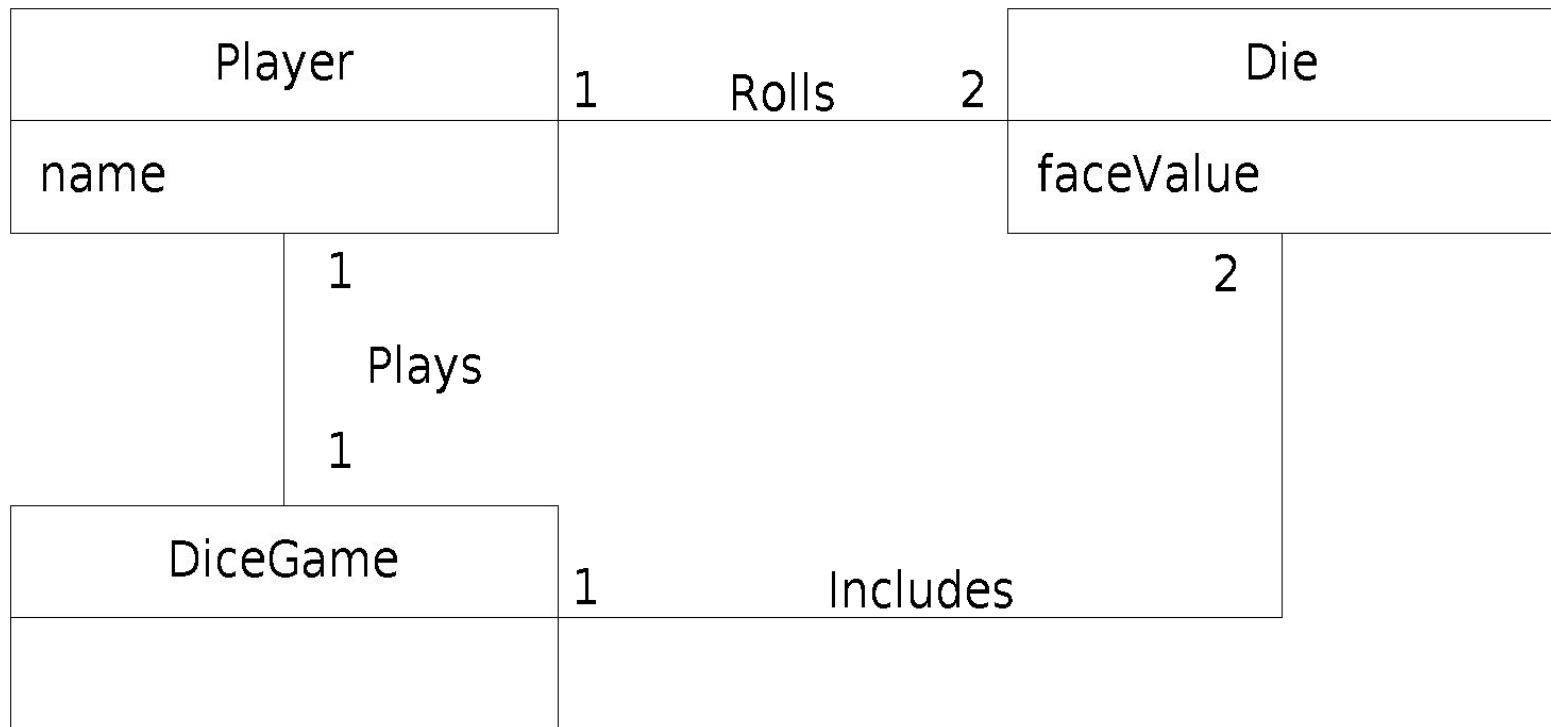
Define use cases

Define domain model

Define interaction diagrams

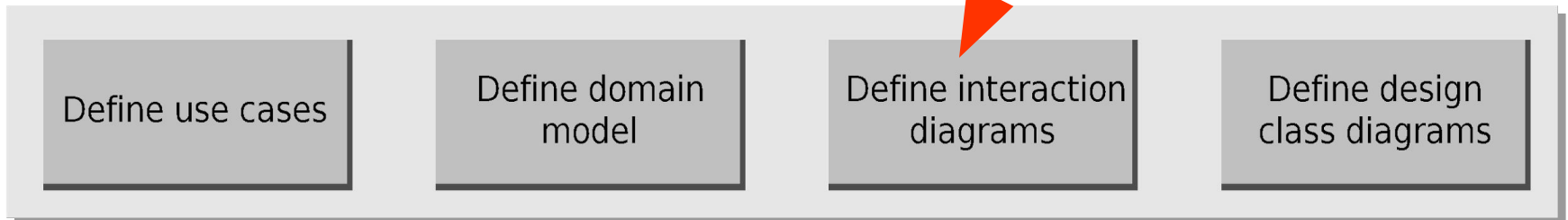
Define design class diagrams

Modelo de Domínio Parcial para o Jogo dos Dados

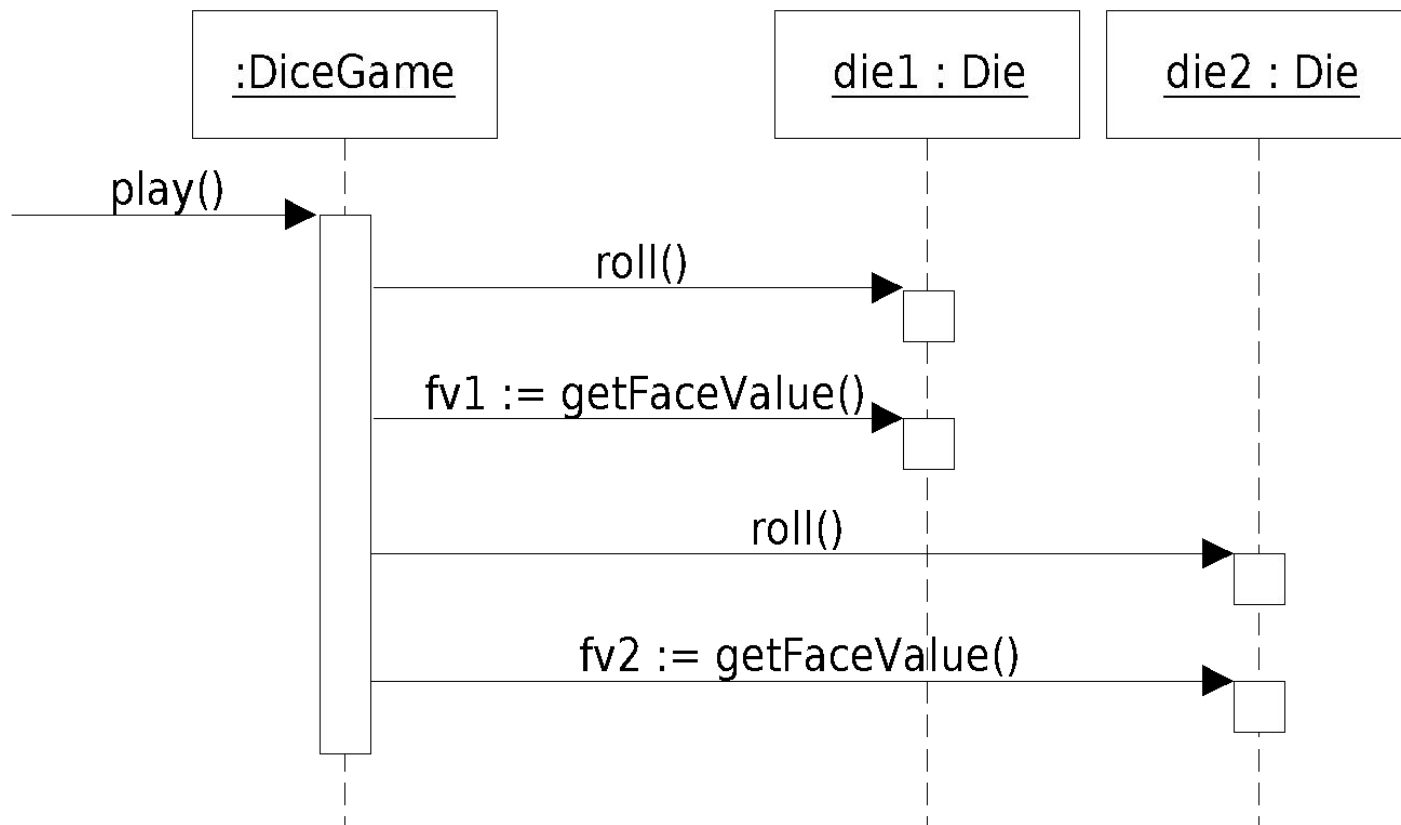


Definir Diagramas de Interação

- Criam uma visão dinâmica de objetos em colaboração
 - definindo **objetos de software** e sua colaboração [ood]
- Mostram o fluxo de mensagens entre **objetos de software** e a invocação de métodos



Passos Essenciais no Jogo dos Dados



Definindo Diagramas de Interação

- No mundo real, o jogador joga os dados
 - No design do sftw o **objeto *Dicegame*** “rolls” os dados
 - i.e. envia **mensagens** para os **objetos *Die***
- Objetos de software têm inspiração no mundo real, mas **não são** modelos diretos do mundo real.

Definir Diagramas de Classe de Design

- Criam uma visão estática das definições das classes
- Ilustram os atributos e métodos das classes



Define use cases

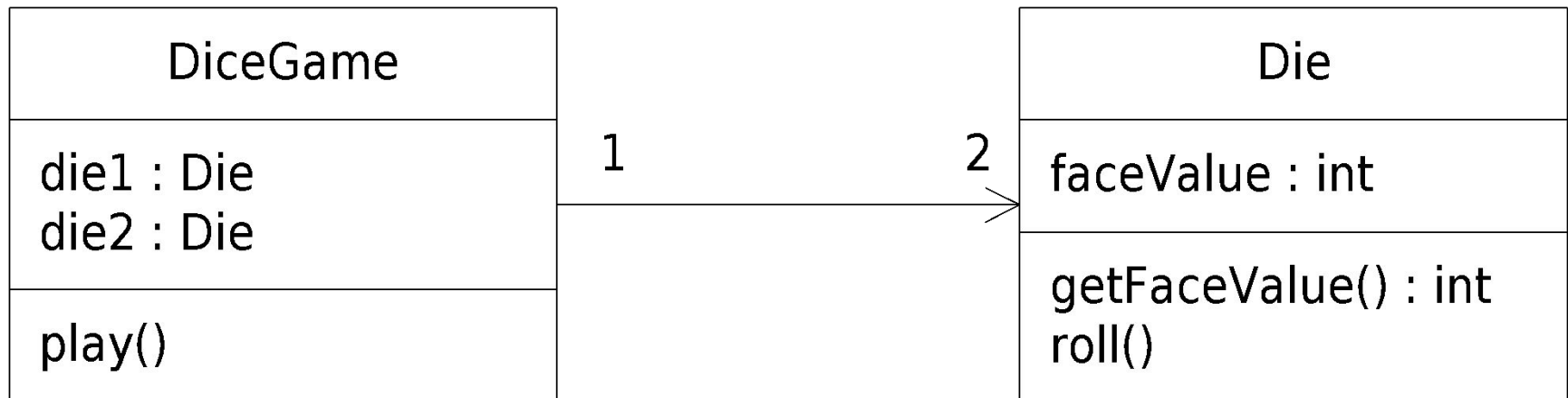
Define domain
model

Define interaction
diagrams

Define design
class diagrams

Diagrama de Classes Parcial para o JD

- Uma vez que uma mensagem “play” é enviada ao objeto “dicegame”, a classe “dicegame” requer um método “play”
- A classe “die” requer os métodos “roll” e “getfacevalue”

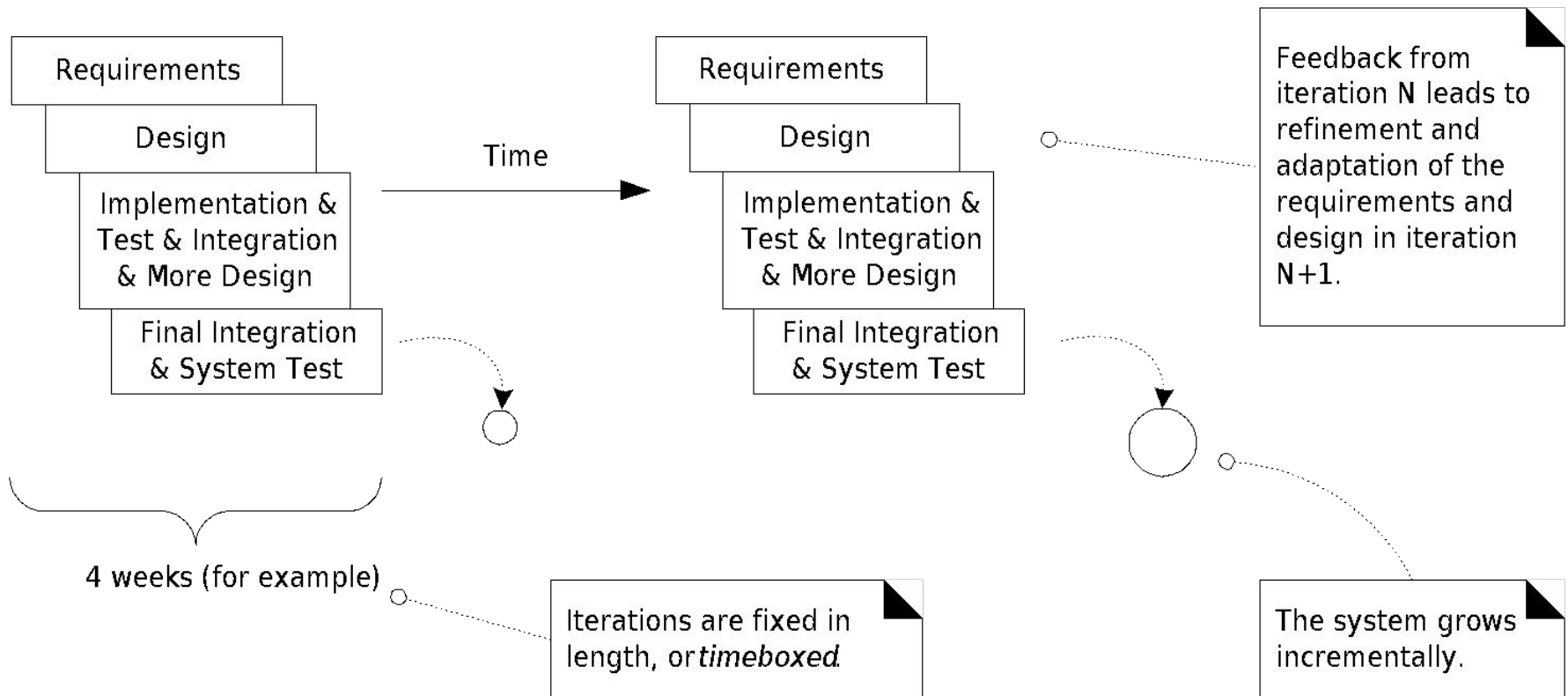


A idéia mais importante no PU:

- **Desenvolvimento Iterativo**

- Desenvolvimento é organizado em uma série de iterações curtas de comprimento fixo [ex. 4 semanas]
- O resultado de cada iteração é um sistema testado, integrado e executável, porém incompleto e não pronto para entrega à produção
- O resultado de uma iteração **não é** um protótipo descartável
 - Desenvolvimento iterativo **não é** prototipação
- Não há pressa para codificação, nem um longo caminho no design/projeto que tenta ter todos os detalhes com perfeição
 - 10 or 15 iterations will be necessary

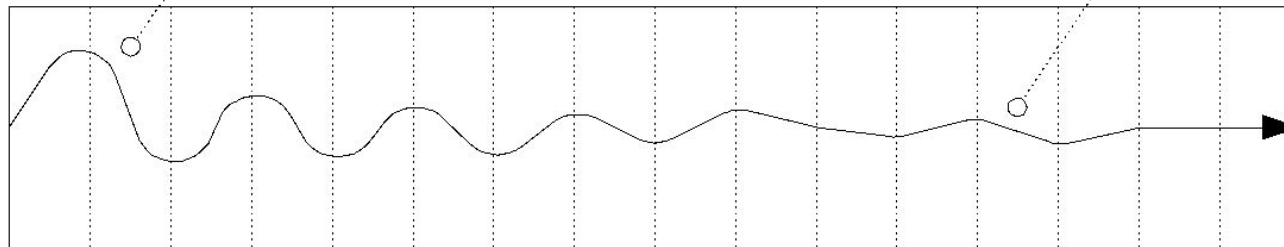
Desenvolvimento Iterativo e Incremental



A instabilidade nos requisitos cai com o tempo

Early iterations are farther from the "true path" of the system. Via feedback and adaptation, the system converges towards the most appropriate requirements and design.

In late iterations, a significant change in requirements is rare, but can occur. Such late changes may give an organization a competitive business advantage.



one iteration of design,
implement, integrate, and test

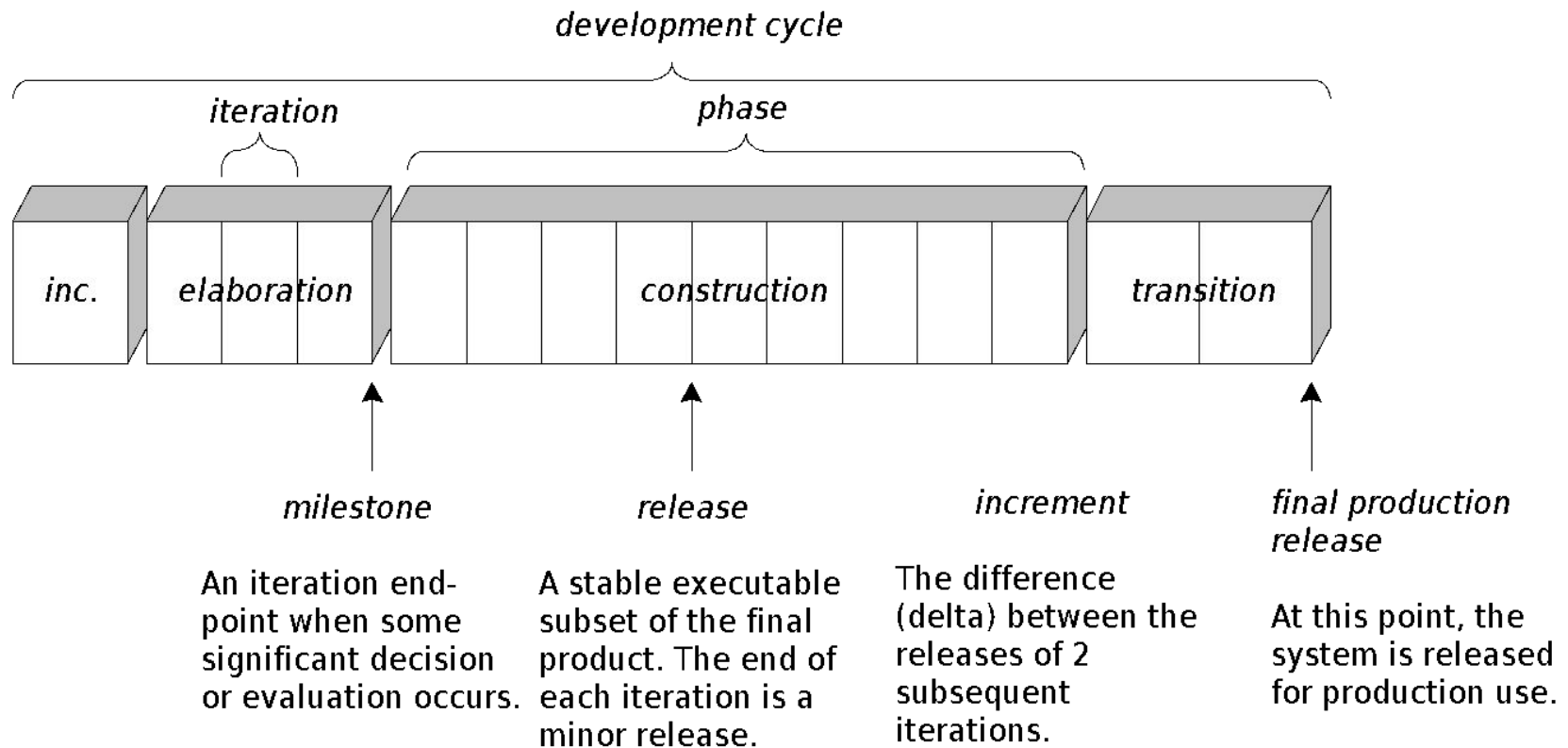
Boas práticas do PU e conceitos chave

- Trate aspectos de alto risco e valor já nas primeiras iterações
- Envolver usuários continuamente para avaliação, feedback e requisitos
- Construa um núcleo coeso de arquitetura já nas primeiras iterações
- Verifique qualidade continuamente, teste cedo e realisticamente
- **Aplique Casos de Uso**
- **Modele o sftw visualmente com UML**
- **Gerencie requisitos cuidadosamente**

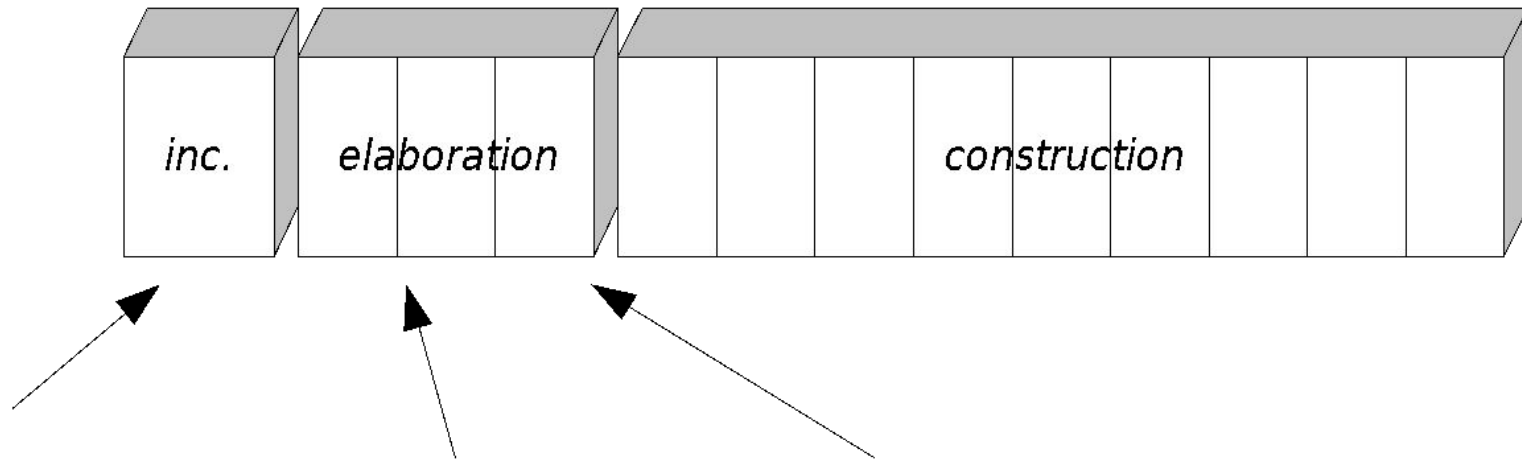
Fases de um Projeto PU

- **Inception:**
 - Visão aproximada, caso de negócio, escopo, estimativas vagas
- **Elaboration:**
 - Visão refinada, implementação iterativa da arquitetura (*core*), resolução dos alto riscos, identificação de mais requisitos e escopo, estimativa mais realista
- **Construction:**
 - Implementação iterativa dos demais elementos de baixo risco e preparação para entrega
- **Transition:**
 - Testes Beta, entrega

Cronograma no PU



Estimativas no PU

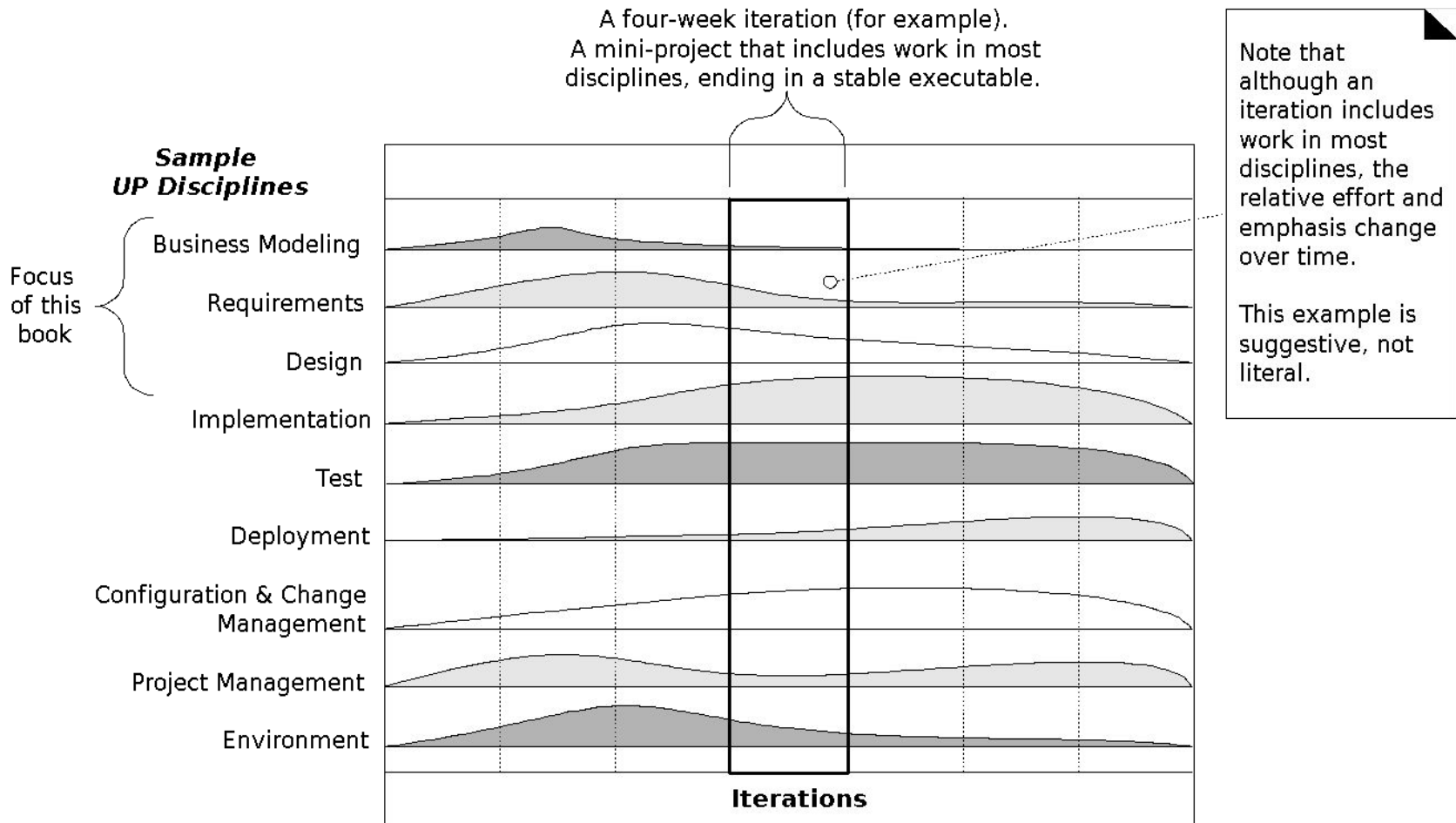


Estimates during inception are *not* used to commit to project duration and effort. Rather, they provide guidance to decide if it is worth continuing on to elaboration and doing some realistic investigation.

At the end of elaboration iteration 1, a believable estimate *starts* to emerge.

After two elaboration iterations, and more so by the end, there has been enough realistic investigation to generate and commit to overall project effort and duration estimates.

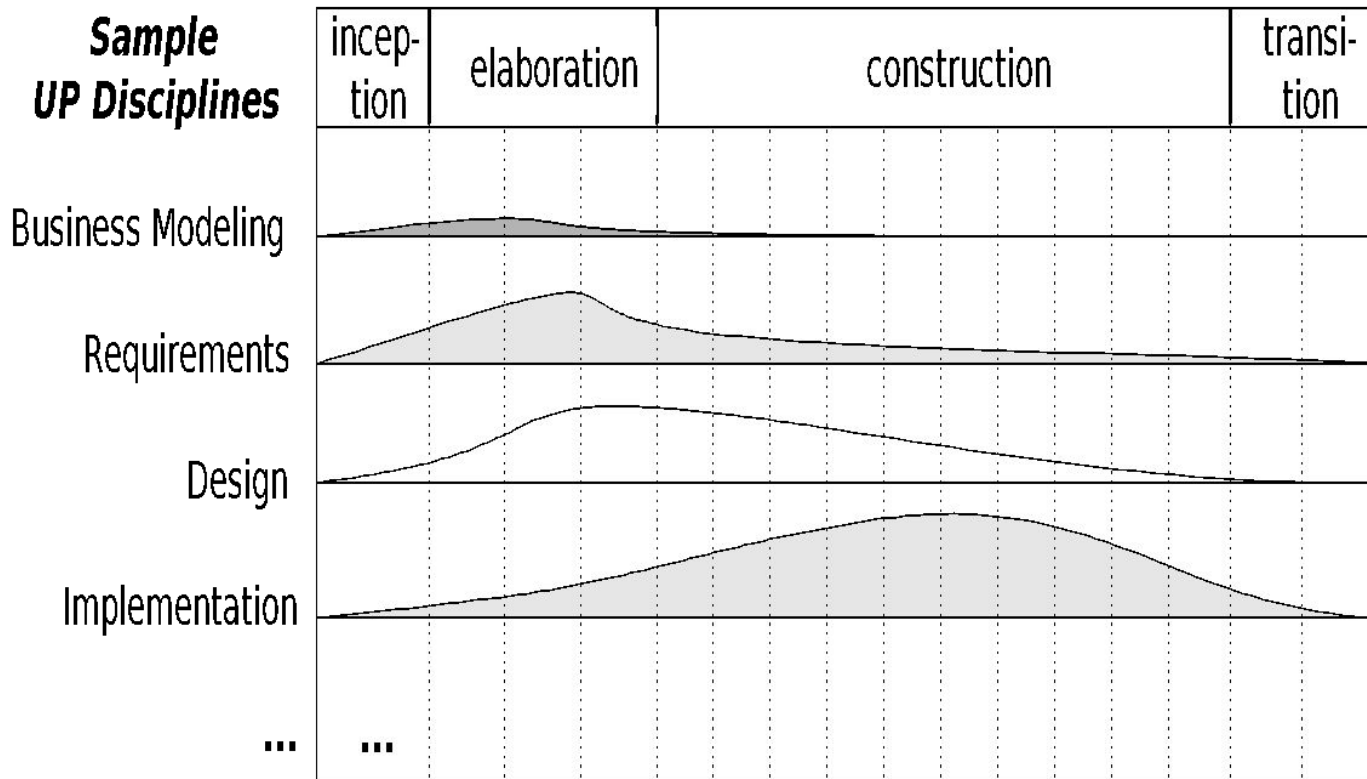
Disciplinas no PU



Disciplinas no PU

- **Business Modeling**
 - Modelagem de objetos do domínio
 - Modelagem dinâmica dos processos de negócio em larga escala
- **Requirements**
 - Escrita de Casos de Uso
- **Design**
 - Arquitetura, objetos, bases de dados, etc. gerais
- **Implementation**
- ...
- **Environment**
 - Customização do processo para o Projeto

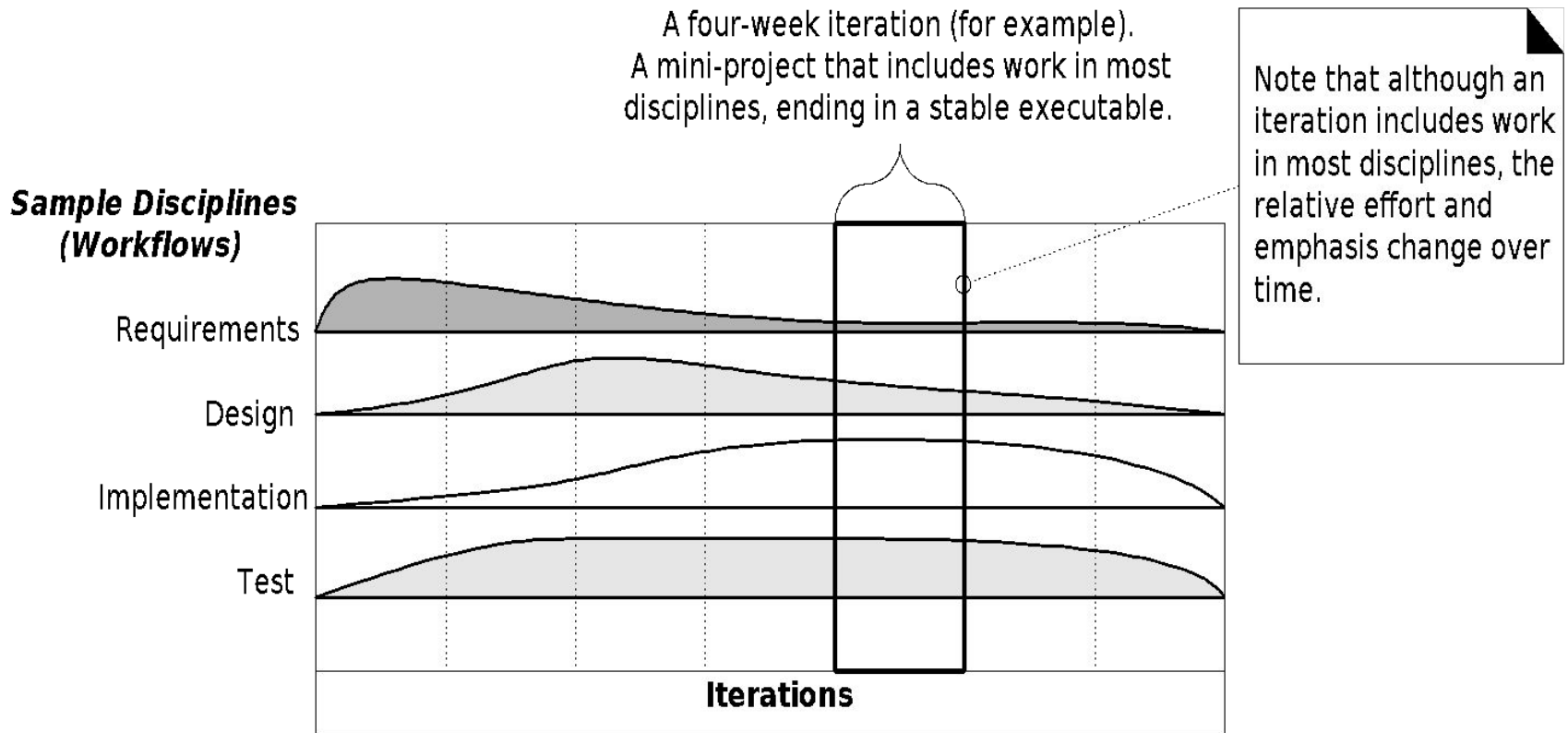
Disciplinas e Fases



The relative effort in disciplines shifts across the phases.

This example is suggestive, not literal.

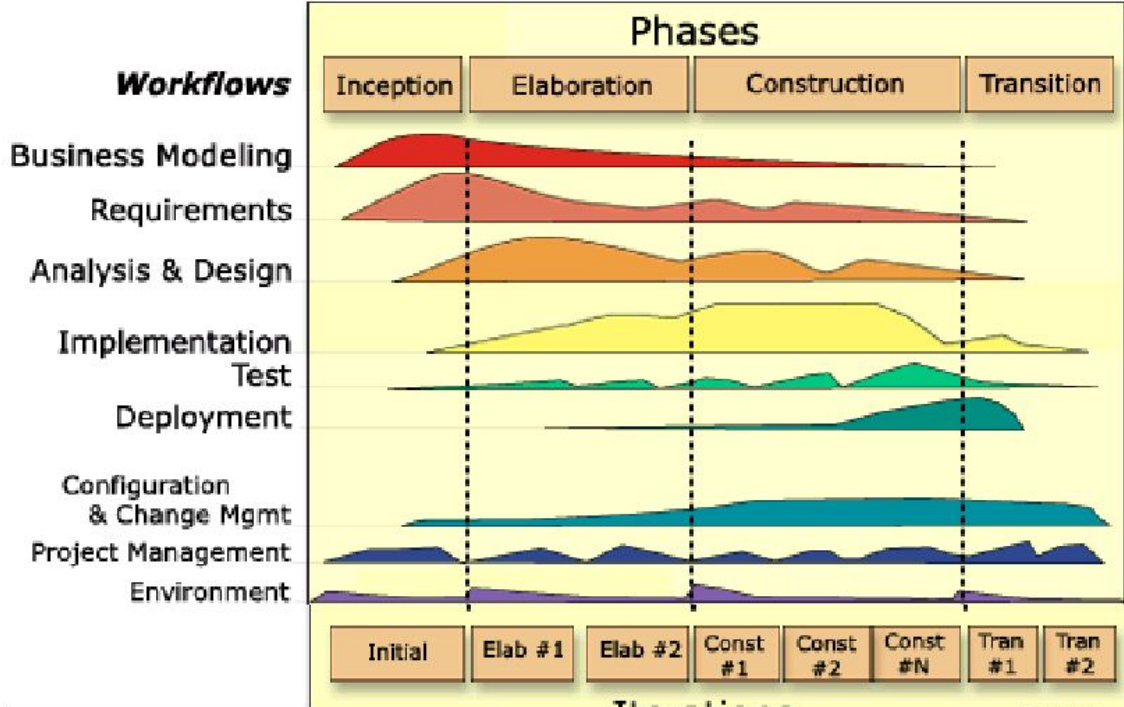
Disciplinas do PU



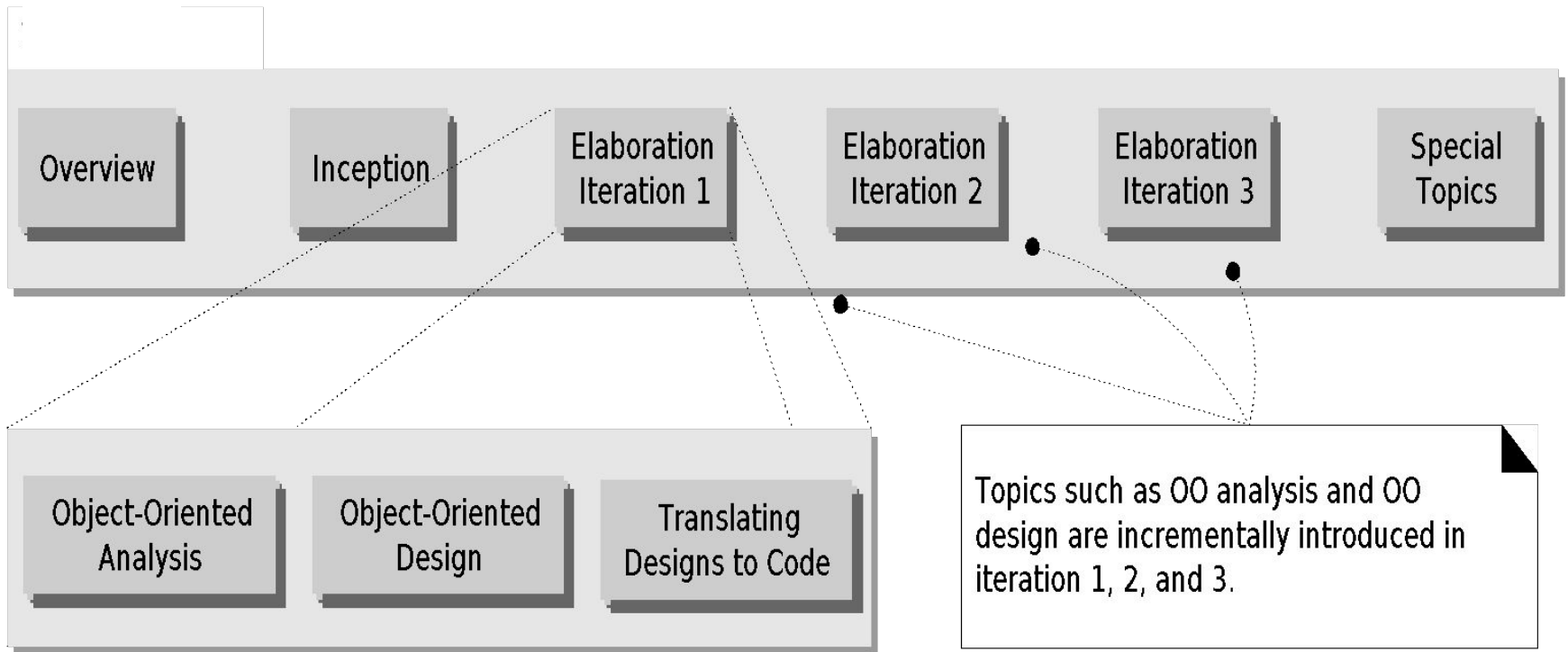
- Overview
- Site Map
- Roadmaps
- Navigation Tools
- Key Concepts
- Glossary
- References
- Best Practices
- What's New
- Phases
 - Inception
 - Elaboration
 - Construction
 - Transition
- Core Workflows
- Workers and Activities
- Artifacts
- Tool Mentors
- Templates
- White Papers
- Work Guidelines
- Web Resource Center
- About the Unified Process

Overview

- [Artifacts](#)
[Examples](#)
[Workers](#)
[Roadmaps](#)
[Site Map](#)



Visão Geral



| Discipline | Artifact | Incep. I1 | Elab. E1..En | Const. C1..Cn | Trans. T1..T2 |
|--------------------------|--|----------------------|-------------------------|--------------------------|--------------------------|
| Business Modeling | Domain Model | | Start | | |
| Requirements | Use-Case Model Vision Supplementary Specif Glossary | Start S S S | Refine R R R | | |
| Design | Design Model SW Architecture Doc Data Model | | S S S | R R | |
| Implementation | Implementation Model | | S | R | R |
| Project Manag. | SW Development Plan | S | R | R | R |
| Testing | Test Model | | S | R | |
| Environment | Development Case | S | R | | |

Estudo de Caso: o Sistema NextGen POS

**Point Of Sale
[POS] system**



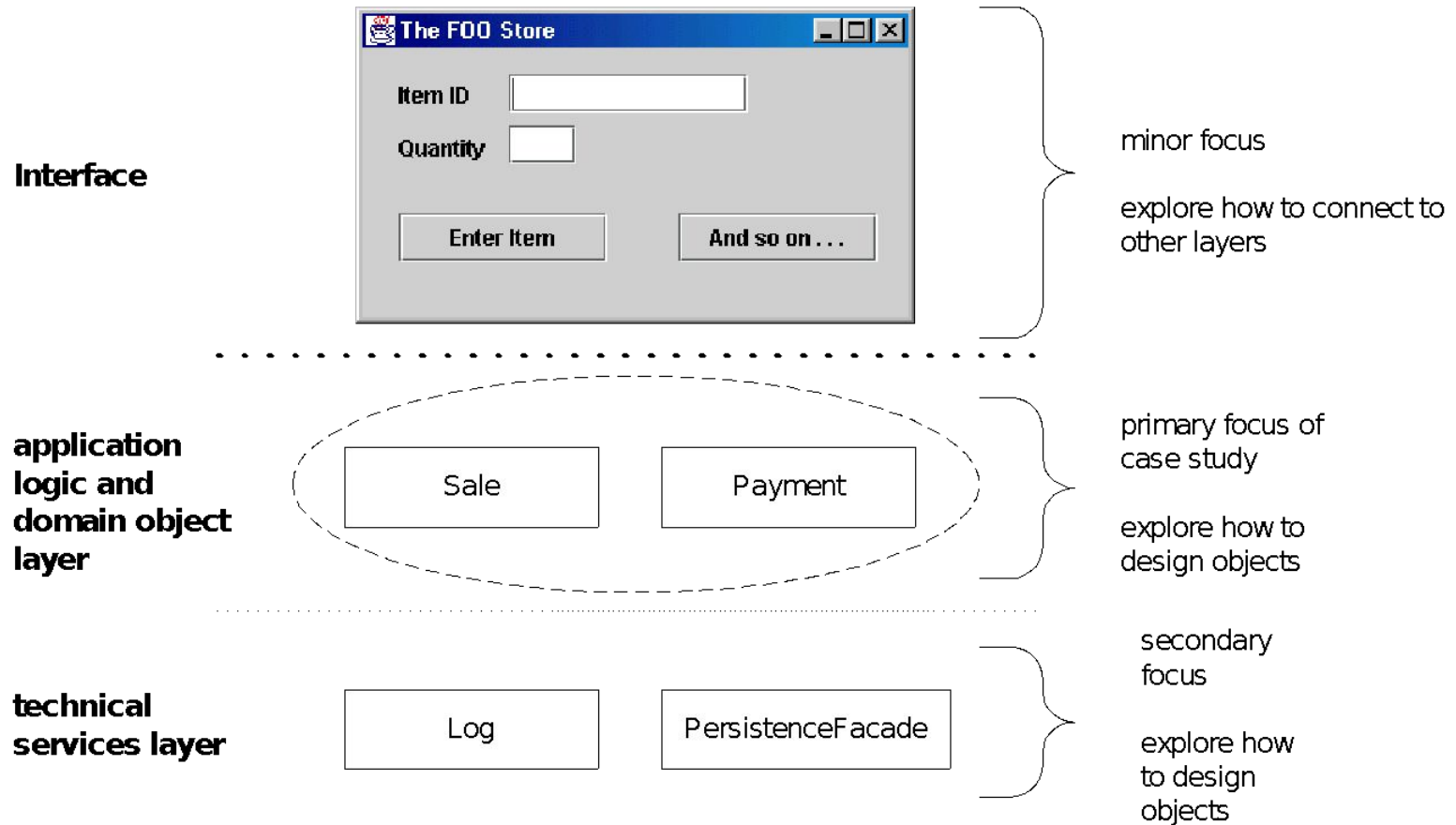
O Sistema NextGen POS

“A POS system is a computerized application used (in part) to record sales and handle payments; it is typically used in a retail store. It includes hardware components such as a computer and a bar code scanner, and a sftw to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault-tolerant; that is, even if remote services are temporarily unavailable (such as the inventory system), they must still be capable of capturing sales and handling at least cash payments (so that the busines is not crippled)”

Camadas de Arquitetura

- **Interface de Usuário**
 - Interface Gráfica – windows
- **Lógica da Aplicação e Objetos do Domínio**
 - Objetos de Sftw representando conceitos do domínio (ex. a classe *Sale*)
- **Serviços Técnicos**
 - Objetos de propósito geral e subsistemas que provêm suporte para serviços técnicos (ex. interface com database, registro de erros etc)

Camadas de Arquitetura



Como será trabalhado

Iteration 1

Introduces just those analysis and design skills related to iteration one.

Iteration 2

Additional analysis and design skills introduced.

Iteration 3

Likewise.

Referências

- Larman, C. (2004) *Applying UML and Patterns – An Introduction to Object Oriented Analysis and Design and the Unified Process*, Prentice-Hall Inc.