O Processo Unificado (PU)

Análise e Projeto Orientados a Objetos

Profa. Dra. Rosana T. Vaccare Braga

2º semestre de 2016

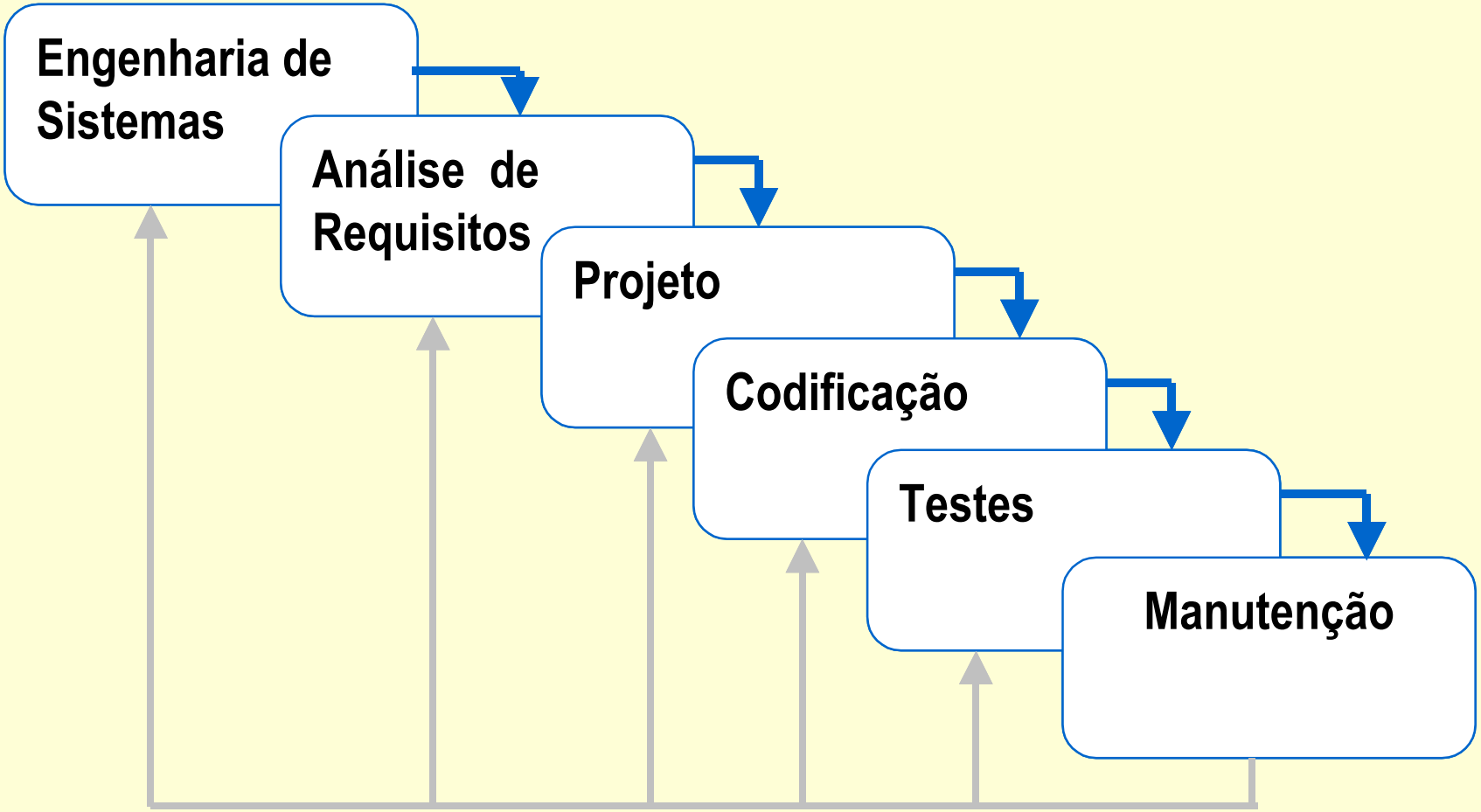
Modelos de Processo de Software

- Desenvolver software é geralmente uma tarefa complexa e sujeita a erros
- Sucesso ou fracasso dependem de inúmeros fatores que ocorrem durante todo o processo
- Necessidade de estabelecer processos sistemáticos para desenvolvimento → Modelos de processo de Software

Exemplos de Modelos de Processo de Software

- Modelo em Cascata
- Modelo de Prototipagem
- Modelo Evolucionário
- Desenvolvimento Baseado em Componentes
- Modelo de Métodos formais
- Extremme Programming
- Processo Unificado

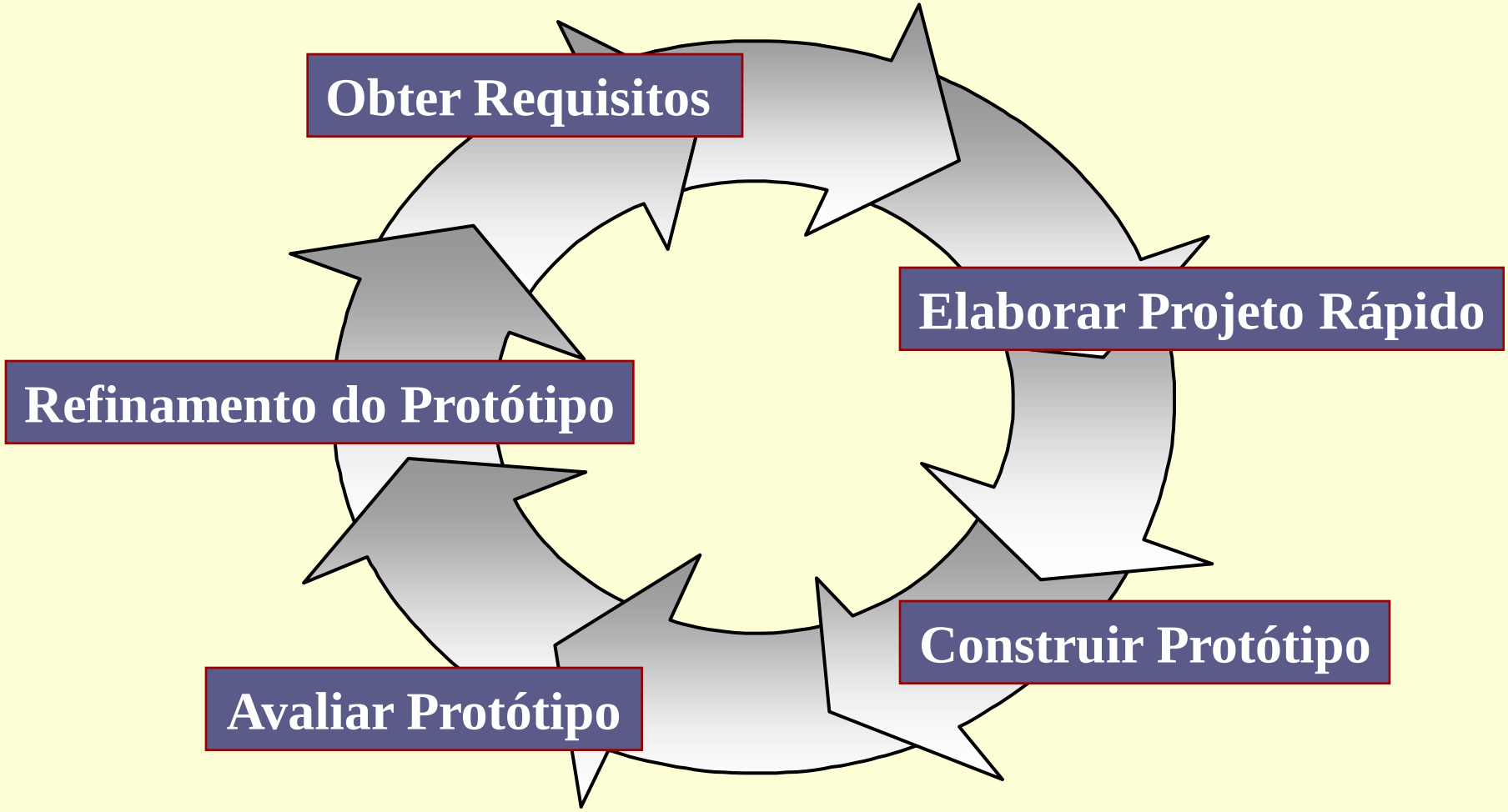
Modelo em Cascata



Problemas com o Modelo Cascata

- Projetos reais raramente seguem o fluxo seqüencial que o modelo propõe
- Logo no início é difícil estabelecer explicitamente todos os requisitos. No começo dos projetos sempre existe uma incerteza natural
- O cliente deve ter paciência. Uma versão executável do software só fica disponível numa etapa avançada do desenvolvimento

O Paradigma de Prototipação
para obtenção dos requisitos



Problemas com a Prototipação

- cliente não sabe que o software que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenibilidade a longo prazo
- desenvolvedor freqüentemente faz uma implementação comprometida (utilizando o que está disponível) com o objetivo de produzir rapidamente um protótipo

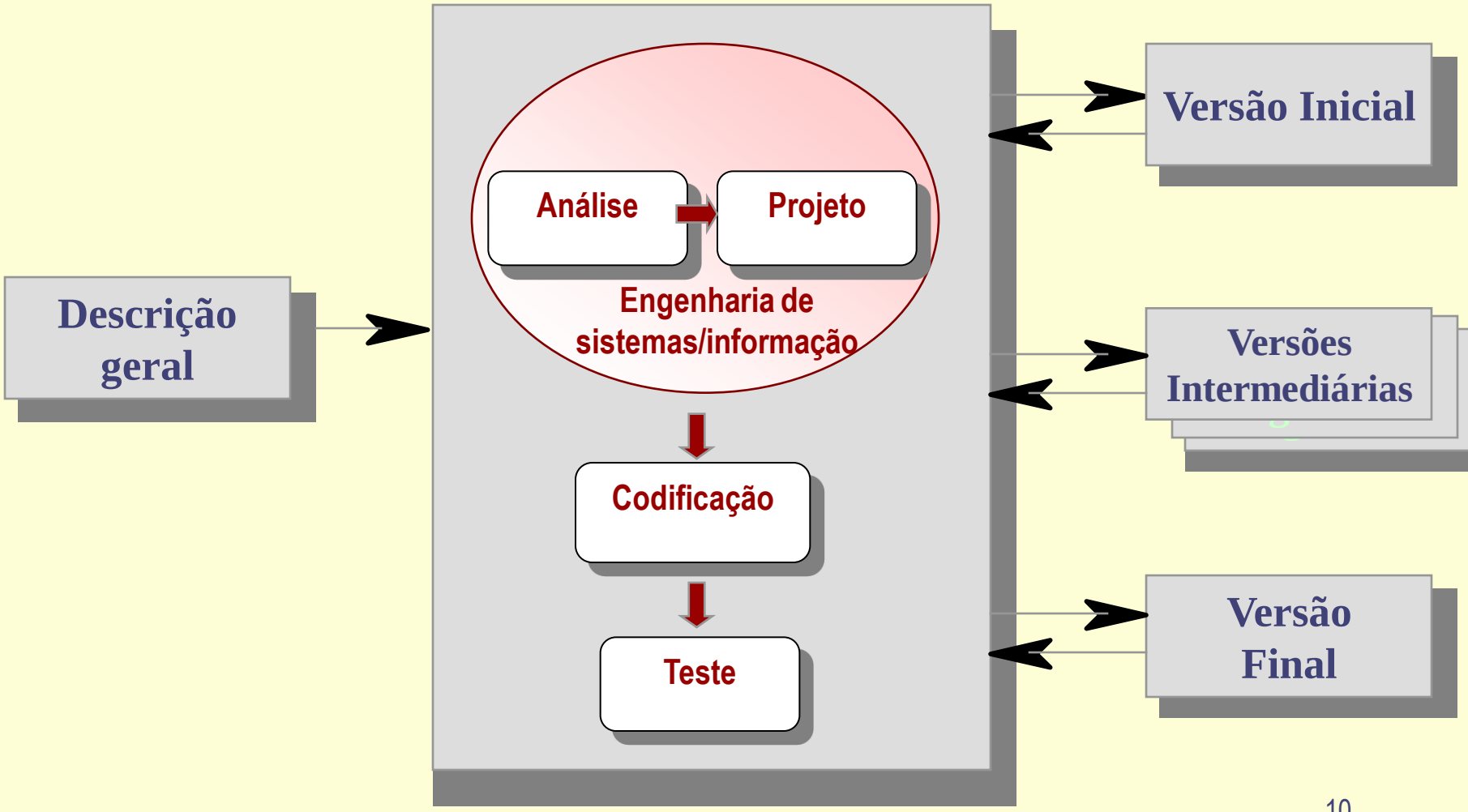
Comentários sobre o Paradigma de Prototipação

- ainda que possam ocorrer problemas, a prototipação é um ciclo de vida eficiente.
- a chave é definir as regras do jogo logo no começo.
- o cliente e o desenvolvedor devem ambos concordar que o protótipo será construído para servir como um mecanismo a fim de definir os requisitos

O Modelo Incremental

- o modelo incremental combina elementos do modelo cascata (aplicado repetidamente) com a filosofia iterativa da prototipação
- o objetivo é trabalhar junto com o usuário para descobrir seus requisitos, de maneira incremental, até que o produto final seja obtido.

O Modelo Incremental



O Modelo Incremental

- a versão inicial é frequentemente o núcleo do produto (a parte mais importante)
  - a evolução acontece quando novas características são adicionadas à medida que são sugeridas pelo usuário
- Este modelo é importante quando é difícil estabelecer *a priori* uma especificação detalhada dos requisitos

O Modelo Incremental

- o modelo incremental é mais apropriado para sistemas pequenos
- As novas versões podem ser planejadas de modo que os riscos técnicos possam ser administrados (Ex. disponibilidade de determinado hardware)

O que é o Processo Unificado (PU)?

- É um modelo de processo de software baseado no modelo incremental, visando a construção de software orientado a objetos
- Usa como notação de apoio a UML (*Unified Modeling Language*)

PU

- História:
  - Raízes no trabalho de Jacobson na Ericsson no final da década de 1960.
  - 1987 – Jacobson iniciou uma companhia chamada de Objectory AB – desenvolvimento de um processo chamado Objectory
  - 1995 – a Rational comprou a Objectory AB, aperfeiçoou o Objectory e foi criado o Processo Objectory da Rational (ROP) (Jacobson, Rumbaugh e Booch)
  - Paralelamente, desenvolviam a UML

PU

- História (cont..):
  - Progresso do ROP e a aquisição e desenvolvimento de ferramentas de desenvolvimento agregaram valor ao ROP
  - 1998 – Rational mudou o nome do ROP para Processo Unificado da Rational (RUP- *Rational Unified Process*)
  - O RUP é uma especialização, com refinamento detalhado, do PU

O que é o PU?

- é um processo de Software: *conjunto de atividades executadas para transformar um conjunto de requisitos do cliente em um sistema de software.*
- é um framework que pode ser personalizado de acordo com as necessidades específicas e recursos disponíveis para cada projeto.

Elementos do PU

- **Um processo descreve**
  - **quem** (papel) está fazendo
  - **o quê** (artefato),
  - **como** (atividade) e
  - **quando** (disciplina).

Papel

- Um **trabalhador** é alguém que desempenha um papel e é responsável pela realização de atividades para produzir ou modificar um artefato.

Artefato

- Porção significativa de informação interna ou a ser fornecida a interessados externos que desempenhe um papel no desenvolvimento do sistema.
- Um **artefato** é algum documento, relatório, modelo ou código que é produzido, manipulado ou consumido.
  - Exemplos: modelo de caso de uso, modelo do projeto, um caso de uso, um subsistema, um caso de negócio, um documento de arquitetura de software, código fonte, executáveis, etc.

Atividade

- É uma tarefa que um trabalhador executa a fim de produzir ou modificar um artefato.

Disciplina

- Descreve as sequências das atividades que produzem algum resultado significativo e mostra as interações entre os participantes
- São realizadas a qualquer momento durante o ciclo de desenvolvimento (Fases do PU)
- Requisitos, Análise, Projeto, Implementação e Teste

Princípios básicos do PU

- Desenvolvimento iterativo
- Baseado em casos de uso
- Centrado na arquitetura

Desenvolvimento Iterativo

- O desenvolvimento de um software é dividido em vários ciclos de iteração, cada qual produzindo um sistema testado, integrado e executável.
- Em cada ciclo ocorrem as atividades de análise de requisitos, projeto, implementação e teste, bem como a integração dos artefatos produzidos com os artefatos já existentes.

Desenvolvimento Iterativo

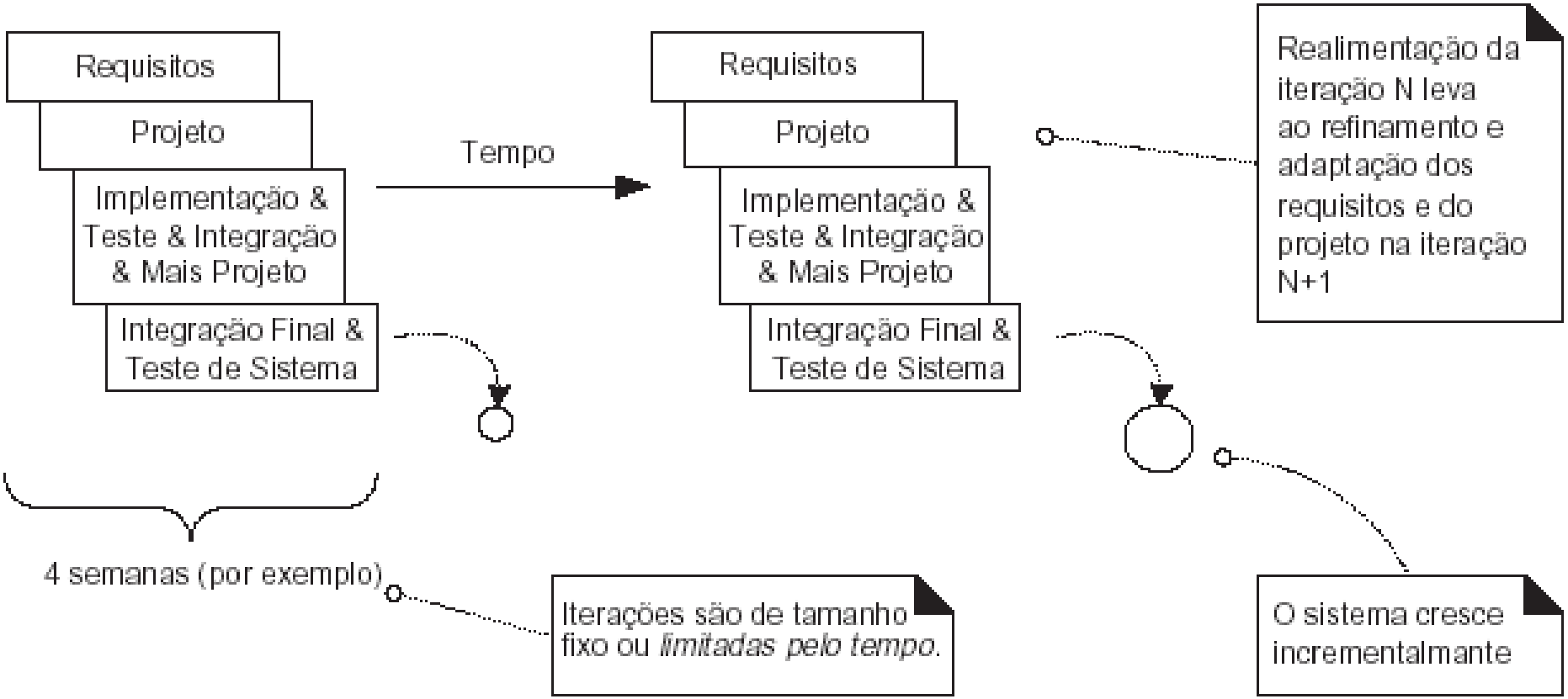


Figura extraída de Larman, 2004

Desenvolvimento Iterativo

- planejar quantos ciclos de desenvolvimento serão necessários para alcançar os objetivos do sistema
- as partes mais importantes devem ser priorizadas e alocadas nos primeiros ciclos
  - a primeira iteração estabeleça os principais riscos e o escopo inicial do projeto, de acordo com a funcionalidade principal do sistema.
  - partes mais complexas do sistema devem ser atacadas já no primeiro ciclo, pois são elas que apresentam maior risco de inviabilizar o projeto.

Desenvolvimento Iterativo

- O tamanho de cada ciclo pode variar de uma empresa para outra e conforme o tamanho do sistema.
  - Por exemplo, uma empresa pode desejar ciclos de 4 semanas, outra pode preferir 3 meses
- Produtos entregues em um ciclo podem ser colocados imediatamente em operação, mas podem vir a ser substituídos por outros produtos mais completos em ciclos posteriores.

Baseado em Casos de Uso

- Um caso de uso é uma seqüência de ações, executadas por um ou mais atores e pelo próprio sistema, que produz um ou mais resultados de valor para um ou mais atores.
- O PU é dirigido por casos de uso, pois os utiliza para dirigir todo o trabalho de desenvolvimento, desde a captação inicial e negociação dos requisitos até a aceitação do código (testes).

Baseado em Casos de Uso

- Os casos de uso são centrais ao PU e outros métodos iterativos, pois:
  - Os requisitos funcionais são registrados preferencialmente por meio deles
  - Eles ajudam a planejar as iterações
  - Eles podem conduzir o projeto
  - O teste é baseado neles

Centrado na Arquitetura

- Arquitetura é a organização fundamental do sistema como um todo. Inclui elementos estáticos, dinâmicos, o modo como trabalham juntos e o estilo arquitetônico total que guia a organização do sistema.
- A arquitetura também se refere a questões como desempenho, escalabilidade, reúso e restrições econômicas e tecnológicas.

Centrado na Arquitetura

- No PU, a arquitetura do sistema em construção é o alicerce fundamental sobre o qual ele se erguerá
- Deve ser uma das preocupações da equipe de projeto
- A arquitetura, juntamente com os casos de uso, deve orientar a exploração de todos os aspectos do sistema

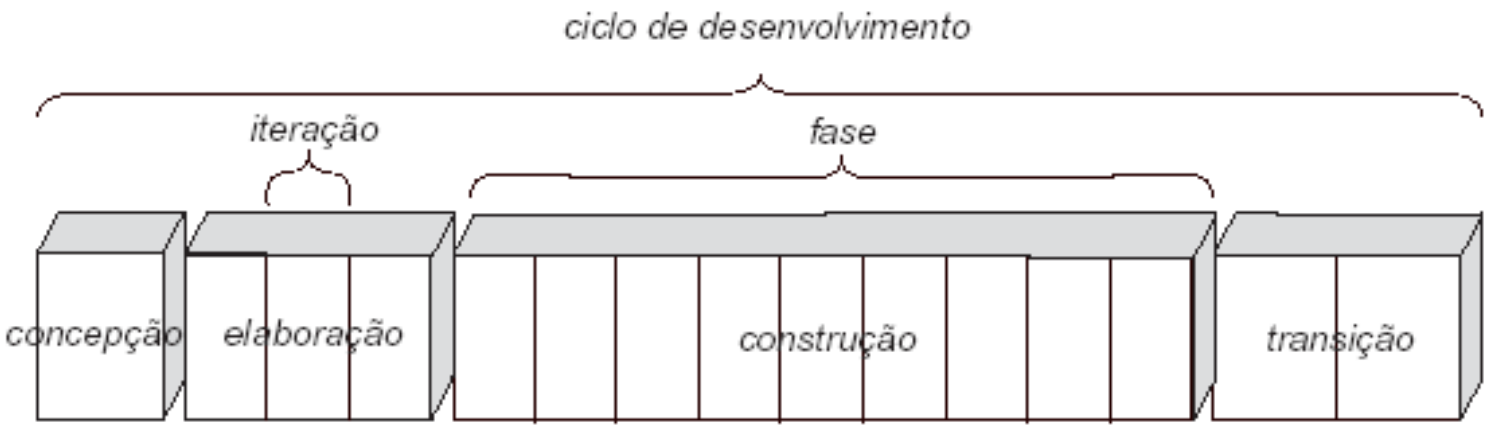
Centrado na Arquitetura

- A arquitetura é importante porque:
  - Ajuda a entender a visão global
  - Ajuda a organizar o esforço de desenvolvimento
  - Facilita as possibilidades de reúso
  - Facilita a evolução do sistema
  - Guia a seleção e exploração dos casos de uso

As Fases do PU

- O PU é dividida em quatro fases:
  - Concepção
  - Elaboração
  - Construção
  - Transição

As Fases do PU



Um ponto de término de uma iteração no qual ocorre alguma decisão ou avaliação significativa.

Um subconjunto estável executável do produto final. O fim de cada iteração é uma entrega secundária.

É a diferença (delta) entre as entregas de duas iterações subseqüentes.

Nesse ponto, o sistema é entregue para uso na produção.

Figura extraída de Larman, 2004

Fases do PU: Concepção

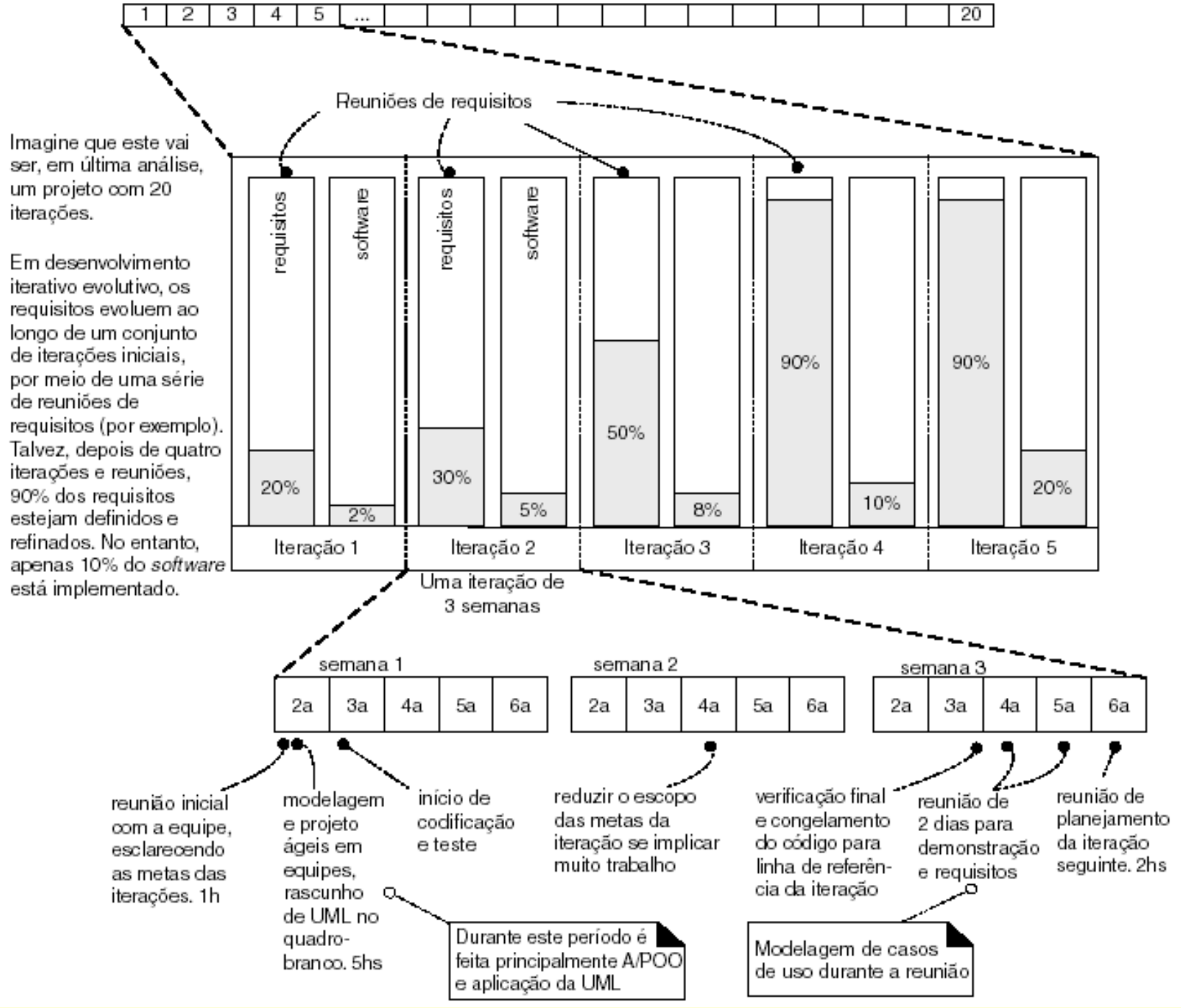
- Estabelece-se a viabilidade de implantação do sistema.
- Definição do escopo do sistema
- Estimativas de custos e cronograma
- Identificação dos potenciais riscos que devem ser gerenciados ao longo do projeto
- Esboço da arquitetura do sistema, que servirá como alicerce para a sua construção.

Fases do PU: Elaboração

- Visão refinada do sistema, com a definição dos requisitos funcionais, detalhamento da arquitetura criada na fase anterior e gerenciamento contínuo dos riscos envolvidos.
- Estimativas realistas feitas nesta fase permitem preparar um plano para orientar a construção do sistema.

Fases do PU: Construção

- O sistema é efetivamente desenvolvido e, em geral, tem condições de ser operado, mesmo que em ambiente de teste, pelos clientes.


1 2 3 4 5 ... 20
Reuniões de requisitos
Imagine que este vai ser, em última análise, um projeto com 20 iterações.
Em desenvolvimento iterativo evolutivo, os requisitos evoluem ao longo de um conjunto de iterações iniciais, por meio de uma série de reuniões de requisitos (por exemplo). Talvez, depois de quatro iterações e reuniões, 90% dos requisitos estejam definidos e refinados. No entanto, apenas 10% do *software* está implementado.
requisitos
software
requisitos
software
20%
2%
30%
5%
50%
8%
90%
10%
90%
20%
Iteração 1
Iteração 2
Iteração 3
Iteração 4
Iteração 5
Uma iteração de 3 semanas
semana 1
semana 2
semana 3
2a 3a 4a 5a 6a
2a 3a 4a 5a 6a
2a 3a 4a 5a 6a
reunião inicial com a equipe, esclarecendo as metas das iterações. 1h
modelagem e projeto ágeis em equipes, rascunho de UML no quadro-branco. 5hs
início de codificação e teste
reduzir o escopo das metas da iteração se implicar muito trabalho
verificação final e congelamento do código para linha de referência da iteração
reunião de 2 dias para demonstração e requisitos
reunião de planejamento da iteração seguinte. 2hs
Durante este período é feita principalmente A/POO e aplicação da UML
Modelagem de casos de uso durante a reunião

Fases do PU: Transição

- O sistema é entregue ao cliente para uso em produção.
- Testes são realizados e um ou mais incrementos do sistema são implantados.
- Defeitos são corrigidos, se necessário.

As Disciplinas do PU

- Se analisarmos as fases do PU, podemos ter a impressão de que cada ciclo de iteração comporta-se como o modelo em Cascata.
- Mas isso não é verdade: paralelamente às fases do PU, atividades de trabalho, denominadas **disciplinas do PU**, são realizadas a qualquer momento durante o ciclo de desenvolvimento
- As disciplinas entrecortam todas as fases do PU, podendo ter maior ênfase durante certas fases e menor ênfase em outras, mas podendo ocorrer em qualquer uma delas

As Disciplinas do PU

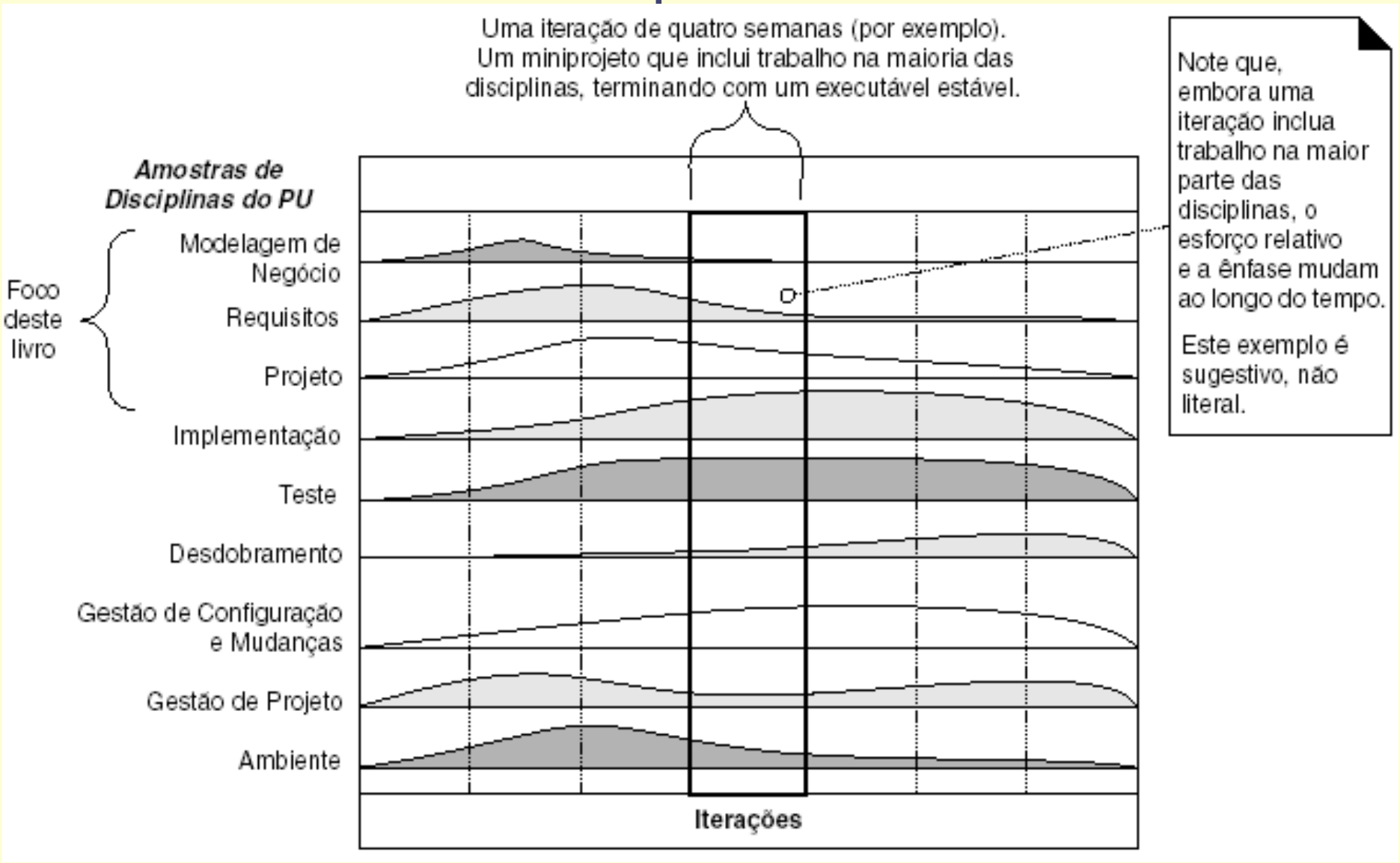
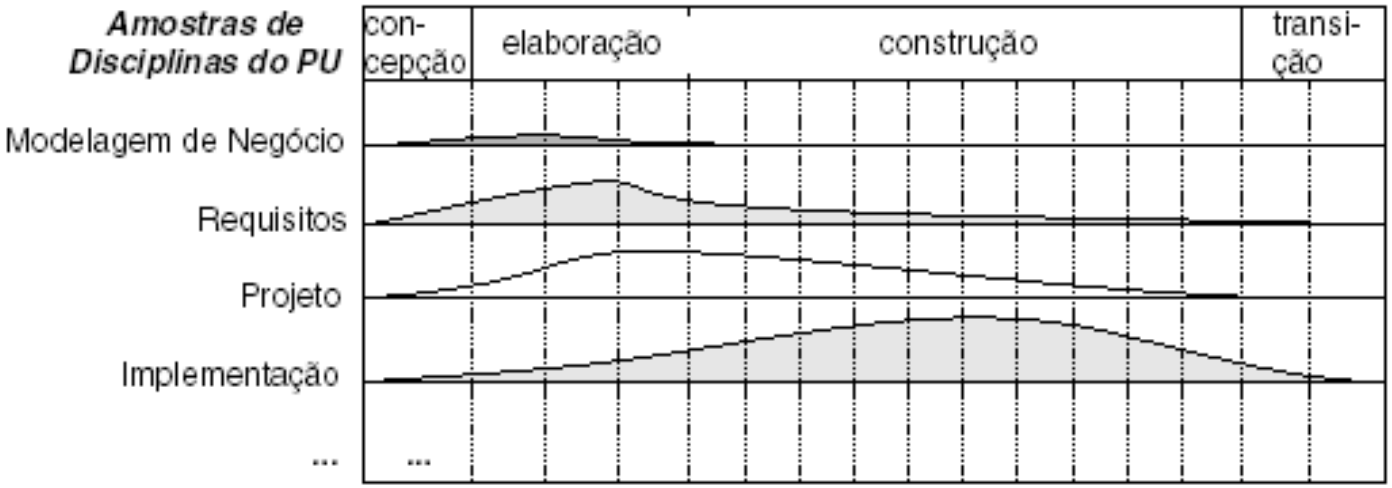


Figura extraída de Larman, 2006

As Disciplinas do PU



O esforço relativo nas disciplinas muda ao longo das fases.

Este exemplo é uma sugestão e não deve ser tomado "ao pé da letra".

Figura extraída de Larman, 2006

Os Artefatos do PU

- Cada uma das disciplinas do PU pode gerar um ou mais artefatos, que devem ser controlados e administrados corretamente durante o desenvolvimento do sistema
- Artefatos são quaisquer dos documentos produzidos durante o desenvolvimento, tais como modelos, diagramas, documento de especificação de requisitos, código fonte ou executável, planos de teste, etc.
- Muitos dos artefatos são opcionais, produzidos de acordo com as necessidades específicas de cada projeto

Os Artefatos do PU

Disciplina	Artefato *Iteração* →	Concepção C_1	Elaboração $E_1...E_n$	Construção $C_1...C_n$	Transição $T_1...T_n$
Modelagem de Negócio	Modelo Conceitual		p		
Requisitos	Diagrama de Casos de Uso	p	r		
	Casos de Uso Textuais	p	r		
	Diagrama de Seqüência do Sistema	p	r		
	Contratos para operações	p	r		
	Glossário	p	r		
Projeto	Diagrama de Classes		p	r	
	Diagrama de Colaboração		p	r	
	Diagrama de Pacotes		p	r	
	Documento de Arquitetura do Software		p		
Implementação	Código fonte			p	r

PU Ágil

- Processos pesados X leves e preditivos X adaptativos
  - Muitos artefatos criados
  - Rigidez e controle
  - planejamento detalhado, elaborado e de longo prazo
  - preditivo, em vez de adaptativo
- Processos ágeis implicam em processos leves e adaptativos

PU Ágil

- O grande número de atividades e artefatos leva a uma impressão errada sobre o PU
- Como fazer:
  - Opte por um pequeno conjunto de atividades e artefatos PU
  - PU é iterativo:
    - requisitos e projetos não são completados antes da implementação
    - Não há um plano detalhado para todo o projeto

Referências

- LARMAN, CRAIG – Utilizando UML e Padrões, 3a edição, Bookman, 2007.
- WASLAWICK, RAUL – Análise e Projeto de sistemas de Informação Orientados a Objetos, Campus, 2004.
- Kendall Scott – O Processo Unificado Explicado, Bookman, 2003.
- Ernani Medeiros – Desenvolvendo Software com UML 2, Makron Books, 2004.