

Processo de Desenvolvimento de Software

Unidade VI – Processo Unificado

Luiz Leão – luizleao@gmail.com

<http://www.luizleao.com>



Estácio

Conteúdo Programático desta aula

- Fases do Processo.
- Ciclo de vida do processo.



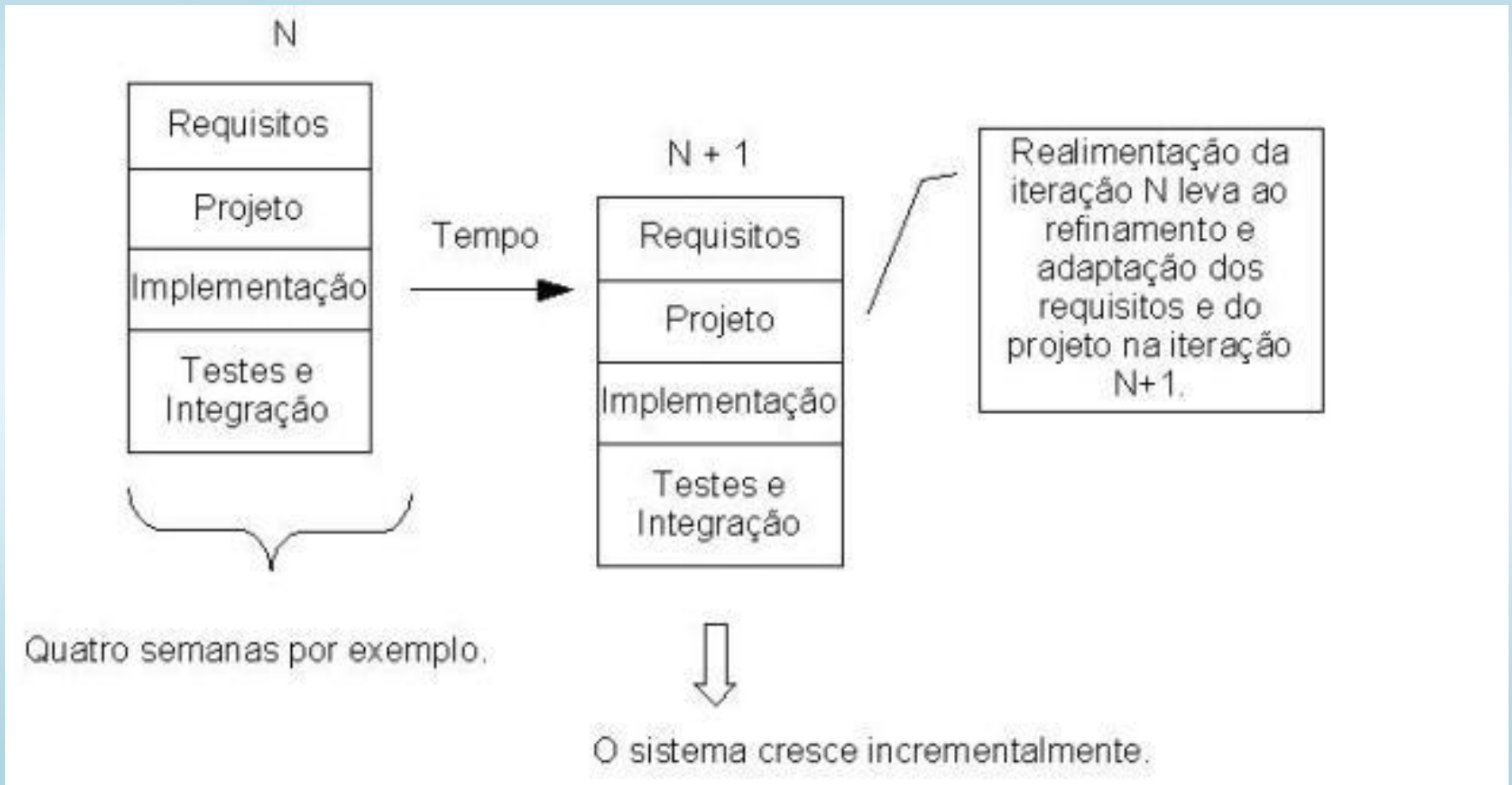
Processo Unificado

- Orientado por Casos de Uso, surgiu para realizar o desenvolvimento de software visando a construção de sistemas orientados a objetos.
- Este modelo de desenvolvimento de software é iterativo e adaptativo, desta forma consegue produzir um sistema de grande porte como se fossem vários pequenos sistemas, o que diminui o risco do projeto.

Processo Unificado

- O PU utiliza um paradigma evolucionário para o desenvolvimento de softwares. O ciclo de vida iterativo é baseado em refinamentos e incrementos sucessivos a fim de convergir para um sistema adequado.
- Em cada iteração incrementa-se um pouco mais o produto, baseando-se na experiência obtida nas iterações anteriores e no *feedback* do usuário. Cada iteração pode ser considerada um miniprojeto de duração fixa, sendo que cada um destes inclui suas próprias atividades de análise de requisitos, projeto, implementação e testes.

Processo Unificado



Processo Unificado

- Segundo Ivar Jacobson, Grady Booch e James Rumbaugh (1999):
 - Hoje em dia, a tendência do software é no sentido de sistemas maiores e mais complexos. Isso deve, em parte, ao fato de que os computadores tornam-se mais potentes a cada ano, levando aos usuários a ter uma expectativa maior em relação a eles (...) Queremos software que seja mais e mais adaptado as nossas necessidades, mas isso, por sua vez, simplesmente torna o software mais complexo.

Características

- Baseado em componentes que realizam interfaces
- Usa UML
- Aspectos:
 - Dirigido por Casos de Uso
 - Centrado em arquitetura
 - Iterativo e incremental
 - Focado no Risco
- Composto pelos 4 P's: Pessoal, Projeto, Produto e Processo

P4 = Pessoa, Projeto, Produto, Processo

- **PESSOAS:** Financiam, escolhem, desenvolvem, gerenciam, testam, usam e são beneficiadas por produtos
- **PROJETOS:** Sofrem alterações. Determinam os tipos de *pessoas* que irão trabalhar no projeto e os *artefatos* que serão usados



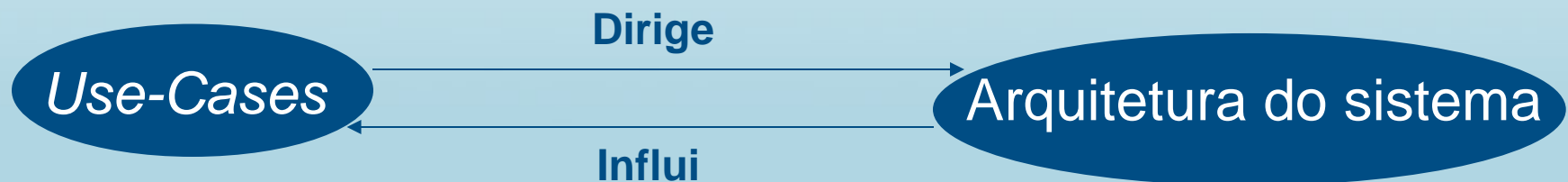
P4 = Pessoa, Projeto, Produto, Processo

- **PRODUTO** código fonte, código de máquina, subsistemas, classes, diagramas: interação, de estados e outros **artefatos**
 - **ARTEFATO** é qualquer tipo de informação criada por uma pessoa (diagramas UML, textos, modelos de interfaces)
- **PROCESSO** define **quem faz o que, quando e como**
- **PU** é um processo. Considera fatores *organizacionais*, do *domínio*, *ciclo de vida* e *técnicos*

Dirigido a *Use-Cases*

Porque USE-CASES??

- **Capturar os requisitos:** o *Diagrama Use-Case* mostra quais **atores** usam quais **use-cases**
- **Dirigir o processo:** para realizar os *use-cases* são definidos *classificadores* (classes, subsistemas, interfaces) e relacionamentos (colaborações) entre estes
- **Elaborar a arquitetura, determinar iterações, determinação dos manuais de usuário**



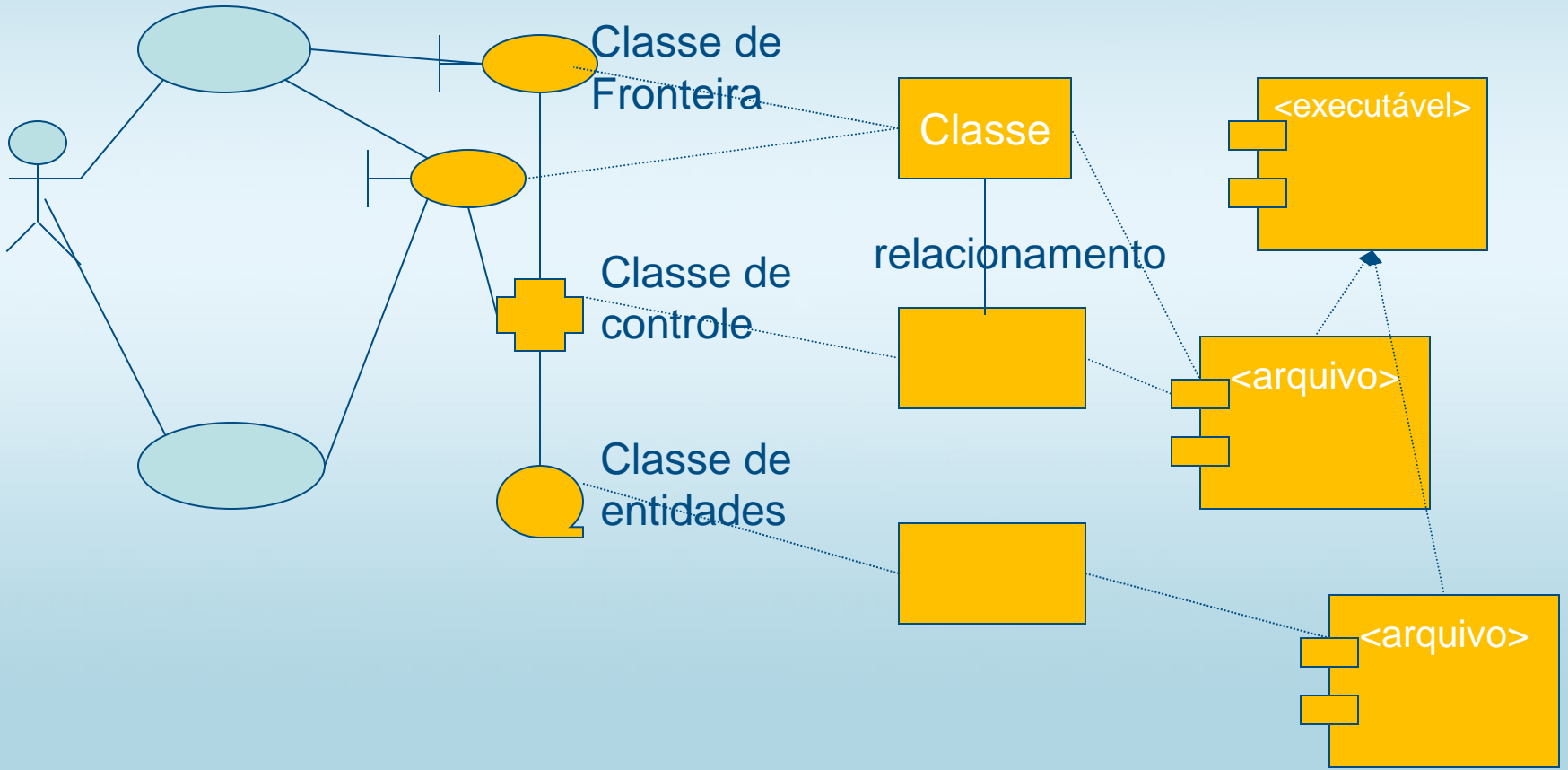
Dirigido a Use-Cases

Modelo *USE-CASE*

Modelo ANÁLISE

Modelo PROJETO

Modelo IMPLEMENT.

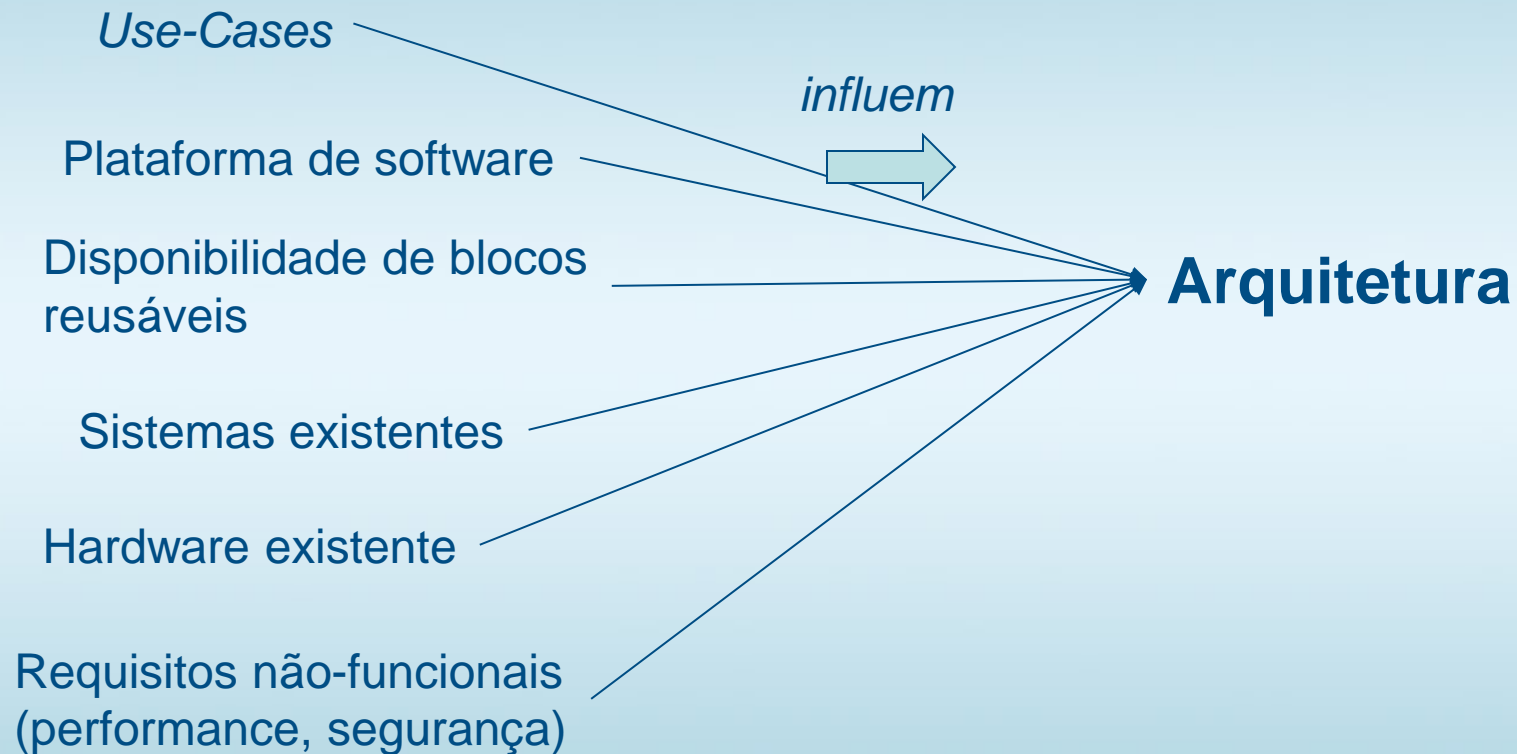


Centrado na arquitetura

Decisões Sobre:

- A organização do sistema como um todo
- Os elementos estruturais, interfaces e seu comportamento
- Composição de elementos estruturais e comportamentais em subsistemas
- A ARQUITETURA descreve as **partes essenciais** do sistema, importantes para todos desenvolvedores
 - Menos de 10% das classes são relevantes para a arquitetura
- Descrição de REQUISITOS ADICIONAIS: segurança, distribuição, concorrência, plataformas, etc.

Centrado na arquitetura



Centrado na arquitetura

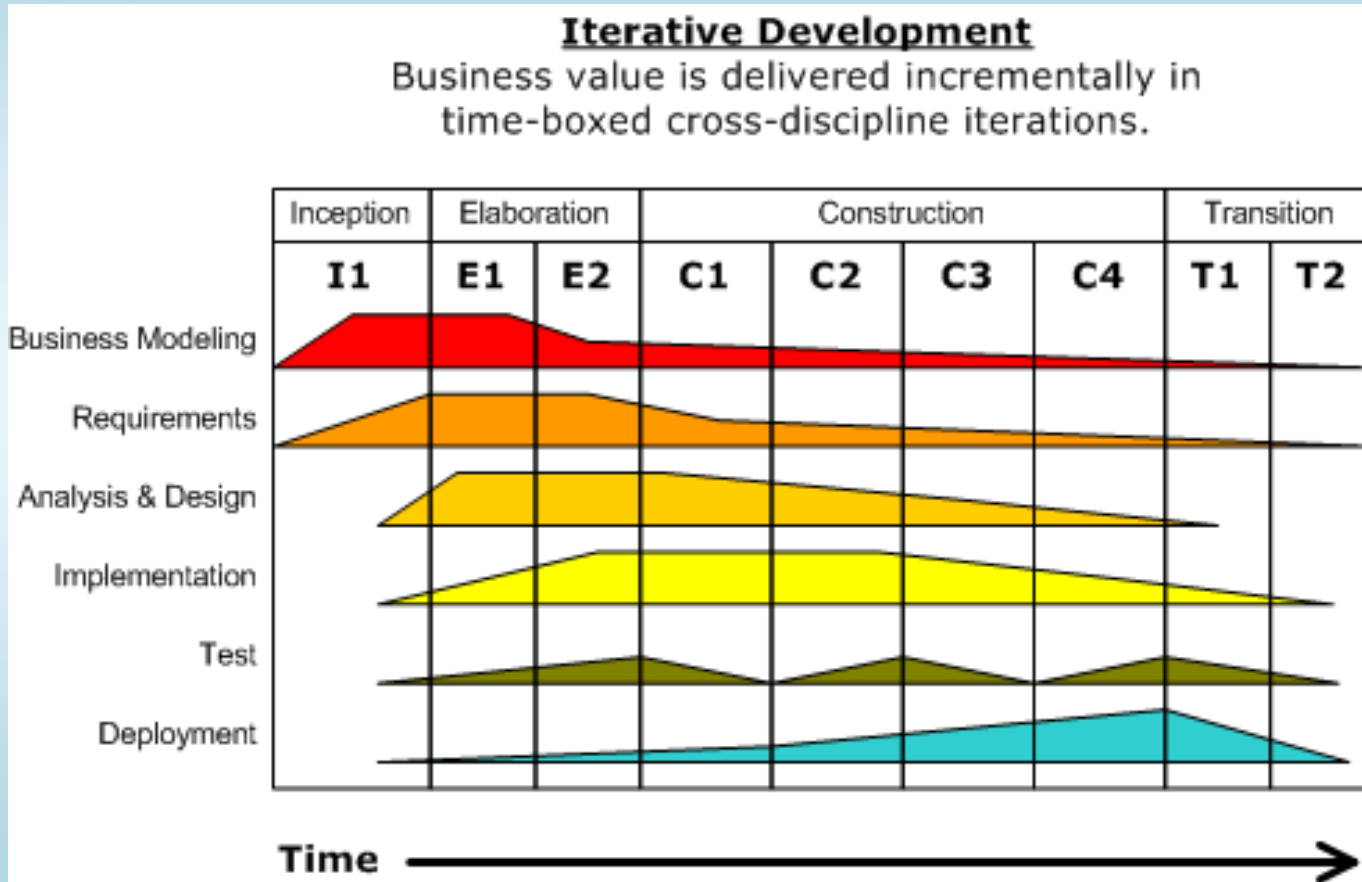


- Sequência para o **arquiteto**:
- Criar uma visão preliminar da arquitetura
- Analisar os **use-cases chave** (5-10%) e especificar um **subsistema** para cada um
- Pela especificação dos subsistemas aparecem mais detalhes da arquitetura e novos *use-cases*
- Repetir o passo acima, até terminar o sistema

Benefícios

- Mitigação precoce, ao invés de tardia, minimizando os riscos do projeto;
- Progresso visível desde o início
- Realimentação, envolvimento do usuário e adaptação imediatos, levando a um sistema refinado que atenda, de forma mais adequada, às reais necessidades dos interessados;
- A complexidade é administrada
 - A equipe não é sobrecarregada pela “paralisa da análise” ou por passos muito longos e complexos;
- O aprendizado obtido em uma iteração pode ser usado para melhorar o próprio processo de desenvolvimento.

Fases do Processo Unificado



Fases do Processo Unificado

- Concepção
- Elaboração
- Construção
- Transição
- Produção

Fases do Processo Unificado

Concepção

- Estabelece o *business case* (prioridade de negócio)
- Envolve tanto a atividade de comunicação com o cliente como a de planejamento
- Delimita o **escopo do sistema**
- Determina **arquitetura** candidata (elementos novos, arriscados)
- Identifica **riscos críticos**
- Identifica potenciais **usuários** ou **clientes** do sistema

Fases do Processo Unificado

Elaboração

- **Determina uma arquitetura estável**
- Identificar e reduzir **riscos** de construção
- Especificar maioria dos Casos de Uso
- Fixar a arquitetura em proporções executáveis
- Preparar o **plano de projeto** (para a próxima fase)
- Estimar e justificar o orçamento
- Finalizar o *business case*

Fases do Processo Unificado

Construção

- **Determina capacidades operacionais iniciais**
- Estender o modelo de **Casos de Uso** para toda a aplicação
- Finalizar a **análise, projeto, implementação e testes**
- Checar **integridade da arquitetura** (com possíveis alterações)
- Monitorar **riscos críticos**

Fases do Processo Unificado

Transição

- Transforma versão *beta* em um sistema em produção
- Preparar atividades de transição
- Avisar clientes sobre mudanças no ambiente (hardware, software, distribuição, ..)
- Preparar **documentação** final
- Carregar o novo sistema
- Corrigir possíveis **defeitos** detectados no beta-teste

Fases do Processo Unificado

Produção (Segundo Pressman [2010])

- Durante essa fase, monitora-se o uso contínuo do software
- Disponibiliza-se suporte para o ambiente operacional (infraestrutura)
- Realiza-se e avalia-se relatórios de defeitos e solicitações de mudanças