

Introdução ao Processo Unificado (PU)

Prof. Anderson Cavalcanti

UFRN-CT-DCA

Processo de Desenvolvimento

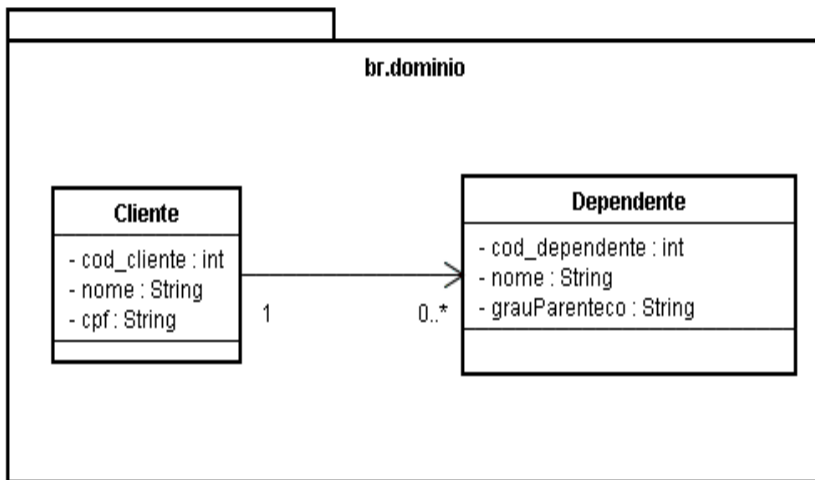
- O conjunto de **atividades de desenvolvimento**, sua **ordem temporal** e a **atribuição de responsabilidades** (papéis de desenvolvedores) definem um processo de desenvolvimento de software;
- Um processo de software é a especificação do processo de transformar necessidades em software;
- Ciclo de Vida de um Processo:
 - Determina as fases do processo;
 - Define atividades importantes e opcionais para cada fase.

Processo Unificado (PU)

- O processo unificado (*Unified Process* UP, ou em português, PU) é um processo de desenvolvimento de software;
- Existem outros processos além do PU?
 - IBM/Rational (Baseado no PU)
 - RUP Rational Unified Process
 - Scott W. Ambler (Baseados no PU)
 - AUP Agile Unified Process
 - EUP Enterprise Unified Process

Onde entra a UML?

- O PU usa a UML como linguagem de modelagem;
- Qual a razão de utilizar linguagens gráficas? Veja:



```
package br.dominio;
public class Cliente {
    private int cod_cliente;
    private String nome;
    private String cpf;
    private Dependente[] dependente;
}
```

```
package br.dominio;
public class Dependente {
    private int cod_dependente;
    private String nome;
    private String grauParenteco;
}
```

Onde entra a UML?

- A UML é usada para facilitar o entendimento de aspectos complexos inerentes ao sistema computacionais;
- A UML é uma família de notações gráficas que ajuda na descrição e no projeto de sistemas de software.

Características do PU

- Dirigido por casos de uso:
 - Ter os casos de uso como entrada (fonte) para a maioria das atividades do processo;
- Centrado na arquitetura:
 - Motivado a desenvolver o produto com base em uma arquitetura de software;
- Iterativo e incremental:
 - Dividir o projeto em partes gerenciáveis, de forma a incrementar as funcionalidades continuamente até o final da construção do produto.

Entendendo o PU: Casos de Uso?

- O que são casos de uso?
 - Seqüência de ações que são executadas por um ou mais atores e pelo próprio sistema;
 - Produz um ou mais resultados de valor para um ou mais atores.
- Como o PU utiliza os casos de uso?
 - Os casos de uso servem de força condutora do desenvolvimento;
 - A expressão dirigido refere-se a utilizar os casos de uso como fonte de todo o trabalho no processo de desenvolvimento:
 - Desde a captação dos requisitos dos usuários aos testes de aceitação.

Entendendo o PU: Casos de Uso?

- Por que casos de uso?
 - São expressos sob a perspectiva dos usuários do sistema;
 - São expressos em língua natural, intuitivamente óbvios para o leitor;
 - Oferecem uma habilidade consideravelmente maior para a compreensão dos reais requisitos do que documentos típicos;

Entendendo o PU: Casos de Uso?

- Por que casos de uso?
 - Oferecem uma habilidade para atingir um alto grau de rastreamento de requisitos dos outros artefatos que são construídos;
 - Oferecem um meio simples de decompor os requisitos dos usuários em pedaços menores que permitam alocação de trabalho de sub-equipes.

Entendendo o PU: Arquitetura?

- O que é uma arquitetura?
 - Organização fundamental do sistema como um todo;
 - Entre aspectos importantes de uma arquitetura estão incluídos elementos estáticos e dinâmicos;
 - Arquitetura também descreve questões de desempenho, escalabilidade, reuso, restrições tecnológicas (os requisitos não funcionais);

Entendendo o PU: Arquitetura?

- A arquitetura do sistema em construção é o alicerce fundamental sobre o qual construirá o produto;
- A arquitetura deve ser vista e compreendida por todos da equipe;
- Arquitetura é o mecanismo para chegar a um produto robusto, flexível, expansível e de custo viável.

Entendendo o PU: Arquitetura?

- Por que centrar o desenvolvimento na arquitetura?
 - Entender a visão global, simplificando o entendimento de sistemas complexos;
 - Organizar o esforço de desenvolvimento, dividindo o software em porções discretas;
 - Facilitar as possibilidades de reuso, facilitando o reuso de componentes dentro das porções discretas;

Entendendo o PU: Arquitetura?

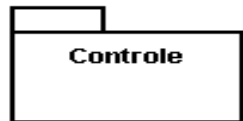
- Por que centrar o desenvolvimento na arquitetura?
 - Facilitar a evolução do sistema, permitindo mudanças mais fáceis dentro das porções principalmente quando não muda as responsabilidades;
 - Dirigir os casos de uso, fornecendo condições de sempre adicionar mais casos de uso.

Exemplo de Arquitetura

- Padrão Arquitetural: Camadas
 - Organizar a estrutura lógica de larga escala de um sistema em camadas distintas de responsabilidades precisas e relacionadas.



Janelas
Relatórios
XHTML, Javascript, entre outros



Trata as requisições da camada de visão



Trata as requisições da camada de controle
Implementa as regras de domínio



Trata as requisições da camada de domínio
Persiste os objetos da camada de domínio (em um BD, por exemplo)

Observe que, neste padrão de arquitetura, cada camada é representada por um pacote, em linguagem UML.

Entendendo o PU: Iterativo e Incremental

- Iteração é um mini-projeto que resulta em uma versão do sistema liberada internamente ou externamente;
- É pressuposto que a cada mini-projeto posterior **incrementa** em funcionalidade o mini-projeto anterior.

Ciclo de Vida

- O que é ciclo de vida?
 - Apresenta um conjunto de períodos, desde nascimento até a sua morte;
 - Cada período possui um conjunto de **fases**;
 - A transição entre as fases é marcada por algum evento.

Fases do PU

- Concepção ou iniciação (*Inception*)
 - Visão aproximada, casos de negócio, escopo e estimativas vagas;
- Elaboração (*Elaboration*)
 - Visão refinada, implementação iterativa da arquitetura central, resolução dos altos riscos, identificação da maioria dos requisitos e estimativas mais realistas;
- Construção (*Construction*)
 - Implementação iterativa dos elementos restantes de menor risco e mais fáceis e preparação para a implementação;
- Transição (*Transition*)
 - Testes beta e implantação.

Fluxo de Trabalho

- O que é fluxo de trabalho?
 - Cada fluxo indica um conjunto de atividades e vários tipos de membros que as executam;
 - No processo unificado existem 5 fluxos de trabalho: Requisito, análise, projeto, implementação e teste;
 - Os fluxos permeiam as 4 fases do processo unificado.
- Fluxos de trabalho é sinônimo de disciplinas (oriundo do RUP).

Fluxos de Trabalho do PU

- **Requisitos** - Qual o objetivo?
 - Visa construir o modelo de casos de uso que captura os requisitos funcionais do sistema.
- **Análise** - Qual o objetivo?
 - Visa construir o modelo de análise, que ajuda os desenvolvedores a refinar e estruturar os requisitos funcionais

Fluxos de Trabalho do PU

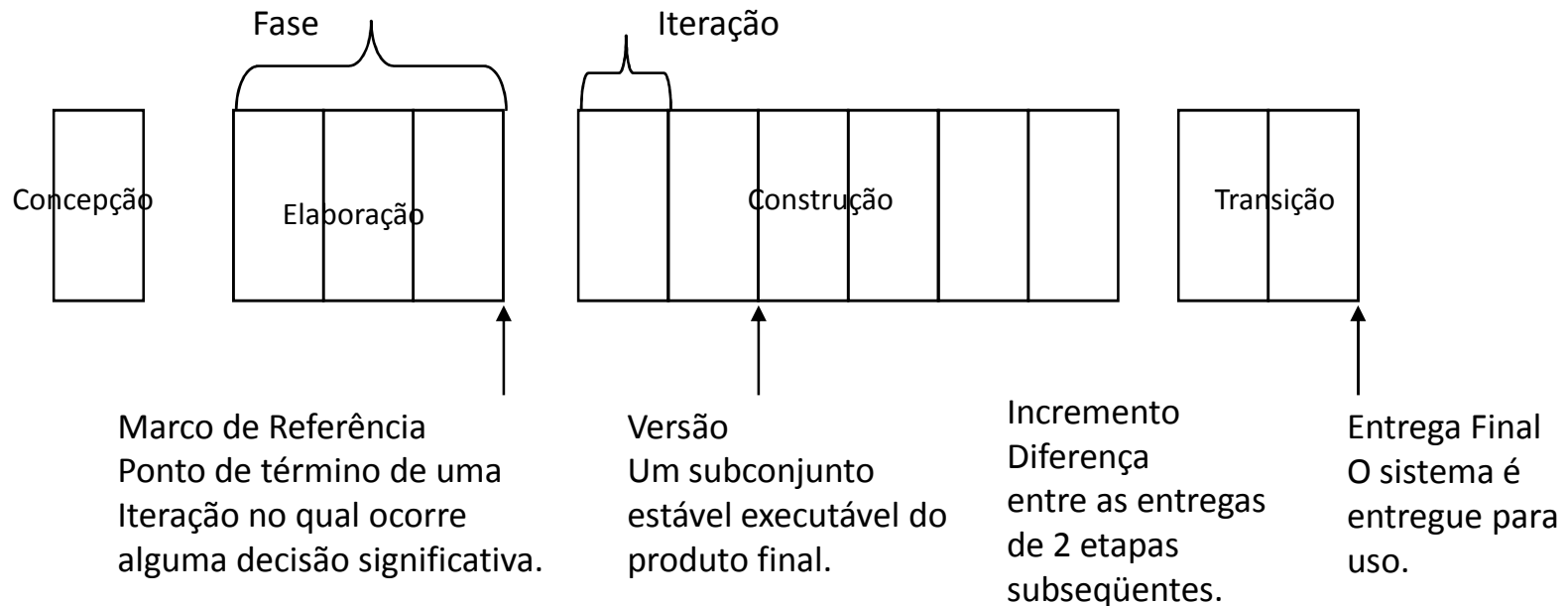
- **Projeto** - Qual o objetivo?
 - Visa a construir o modelo de projeto, o qual descreve as realizações físicas dos casos de uso - abstração de implementação;
 - Visa também o modelo de instalação, que define a organização física do sistema.

Fluxos de Trabalho do PU

- **Implementação** - Qual o objetivo?
 - Visa a construir o modelo de implementação, que descreve como os elementos do modelo de projeto são empacotados em componentes de software.
- **Teste** - Qual o objetivo?
 - Visa a construção do modelo de teste que descreve como os testes de integração e de sistema exercitarão componentes executáveis a partir do modelo de implementação.

Fases x Iterações

- Qual a relação entre fases e iterações?
 - As fases contém iterações seqüenciais;
 - O marco principal (fases) é conseguido a partir de mini-projeto (iterações).



Iterações x Fluxos de Trabalho

- Qual a relação entre iterações e fluxos de trabalho?
 - As iterações **podem** conter os 5 fluxos de trabalho;
 - As iterações do início do processo normalmente ficam nos fluxos de requisitos e análise;
 - Estas iterações podem chegar ao fluxo de projeto, mas raramente ao fluxo de implementação e teste;
 - As iterações do final do processo de desenvolvimento normalmente usualmente executam os fluxos de implementação e teste.

Iterações e incrementos

- Cada fase é dividida em iterações
 - Relembrando, iterações são mini-projetos
 - Uma iteração típica, usa os 5 fluxos de trabalho
 - O resultado da execução dos fluxos é um incremento
- O que é um incremento?
 - Incremento é uma versão do sistema que contém funcionalidade adicionada ou melhorada em comparação com a anterior

Atividades de cada iteração

- Planejar a iteração;
- Executar as disciplinas (fluxos de trabalho);
- Fazer análise ao término da iteração;
- Descartar os riscos que o incremento tratou;
- Revisar o plano do projeto;
- Ir para a próxima iteração, se existir.

Artefatos

- O que são artefatos?
- Artefato é qualquer porção significativa de informação interna ou externa que desempenhe um papel no desenvolvimento do sistema;
- Existem artefatos técnicos e genéricos
 - Exemplo de artefatos genéricos: análise econômica, plano de projeto, riscos.

Trabalhadores

- O que são trabalhadores?
 - Trabalhador é um papel que um indivíduo pode desempenhar no projeto em um dado momento;
- Diferença entre trabalhador e Ator
 - Atores utilizam o sistema e podem participar do desenvolvimento, normalmente fornecendo informações sobre o cliente;
 - Trabalhadores somente participam do desenvolvimento do sistema.

Atividades

- O que são atividades?
 - Atividades são tarefas que um trabalhador executa a fim de produzir ou modificar um artefato;
 - Para executar uma determinada atividade o trabalhador faz uso de artefatos.

Disciplinas

- O que são disciplinas?
 - Determina um conjunto coerente de atividades a serem executadas para cumprir objetivos;
 - A disciplina é especificada por um fluxo de atividades, determinando os artefatos de entrada e saída de cada atividade deste fluxo bem como os respectivos níveis de detalhe de cada artefato.

PU – Fase de Concepção

PU: Concepção

- Deve ser curta
- Devem ser exploradas as seguintes questões:
- Qual a **visão e o caso de negócio para o projeto?**
 - Ele é viável?
 - Devemos comprar ou construir?
 - Estimativa aproximada de custo: é da ordem de 10 mil, 100 mil ou de milhões?
 - Devemos continuar ou parar?

PU: Concepção

- Devem ser explorados alguns requisitos do sistema para responder algumas dessas questões;
- Concepção em uma frase:
 - *“Conceber o escopo do produto, a visão e o caso de negócio”*

Artefatos a Serem Iniciados

<i>Artefato</i>	<i>Comentário</i>
Visão e Caso de Negócio	Descrevem os objetivos e as restrições de alto nível, o caso de negócio, além de um resumo para executivos
Modelo de Casos de Uso	Descreve os requisitos funcionais e requisitos não funcionais
Especificações Suplementares	Descrevem outros requisitos
Glossário	Contém a terminologia-chave do domínio
Lista de Riscos e Plano de Gerenciamento de Riscos	Descrevem os riscos de negócio, técnicos, de recursos e de cronograma, as idéias para a sua minimização ou solução

Outros artefatos podem ser construídos, no entanto, dentro do escopo desta disciplina, apenas a Visão e Caso de Negócio e o Modelo de Caso de Uso serão utilizados.

Artefatos a Serem Iniciados

- Com exceção do Modelo de Casos de Uso, nenhum desses artefatos necessita ser implementado, a não ser que contribuam de fato com o desenvolvimento;
- Como estamos na concepção, o conteúdo da investigação e dos artefatos deve ser leve;
- Modelos de Documento estão disponíveis em <http://www.dca.ufrn.br/~anderson/>

Compreensão dos Requisitos

- **Requisitos = são as capacidades e** condições às quais o sistema – em termos mais amplos, o projeto – deve atender
- Desafio: encontrar, comunicar e lembrar o que é realmente necessário → Gerência de Requisitos

Tipos de requisitos (FURPS+)

- **Funcionais**
 - Características, capacidade e segurança
- **Usabilidade**
 - Fatores humanos, recursos de ajuda e documentação
- **Confiabilidade**
 - Frequência de falhas, capacidade de recuperação e previsibilidade

Tipos de requisitos (FURPS+)

- **Desempenho**

- Tempos de resposta, fluxo de vazão (throughput), precisão, disponibilidade e uso de recursos

- **Facilidade de Suporte**

- Facilidade de adaptação e de manutenção, internacionalização e configurabilidade

Tipos de requisitos (FURPS+)

- **Implementação**
 - Limitações de recursos, linguagens e ferramentas, hardware, etc.
- **Interface**
 - Restrições impostas pelas interfaces com sistemas externos

Tipos de requisitos (FURPS+)

- **Operações**
 - Gerenciamento do sistema no ambiente operacional
- **Empacotamento**
- **Questões Legais**
 - Licenças de uso, etc.

Exploração dos Requisitos

- Caracterizados de forma macro como funcionais (comportamentais) e não funcionais (todos os outros)
- Requisitos funcionais são explorados:
 - No Modelo de Casos de Uso
 - Na lista de características do sistema do artefato de Visão
- Os outros requisitos podem ser registrados:
 - Nos Casos de Uso com os quais eles se relacionam
 - No artefato de Especificações Suplementares

Referências

- JACOBSON, I.; BOOCH, G. and RUMBAUGH, J. *The Unified Software Development Process*. Reading, MA.: Addison-Wesley, 1999,
- ALLEIXO, F. *Notas de aula da disciplina de Análise e Projeto Orientado a Objeto*, CEFET/RN, 2007.
- SCOTT, K. *O Processo Unificado Explicado*. Ed. Bookman, 2003;
- MINORA, L. *Notas de Aula de Eng. de Software I*, CEFET/RN, 2006.