

# DIRETRIZES PARA A CONSTRUÇÃO DE **CASOS DE USO EFICAZES**

Aline Martins dos Santos, Edilson Modesto Radael, Rosemeire Fátima Pinhata, Sandreliz Alves Rossi e Sandro Mendes

Faculdade de Filosofia, Ciências e Letras - Centro Universitário Fundação Santo André {line.martins,oedilsao,rosepinhata,sandreliz,osandrom@}gmail.com

### **RESUMO**

Este artigo tem o propósito de apresentar diretrizes para a construção de casos de uso eficientes, visando o processo de desenvolvimento de software. Conceitua a UML, descrevendo cada um de seus diagramas, especificamente o Diagrama de Casos de Uso. Foram utilizados como exemplo, situações do dia-a-dia, mostrando as principais técnicas, passo a passo, para a construção de um bom caso de uso.

Palavras-chave: casos de uso, UML, engenharia de software.

### **ABSTRACT**

This article intends to present lines of direction for the construction of efficient Use Cases viewing the process of software development. It considers UML describing each one of their diagrams, specifically the Use Cases Diagram. They were used as example, situations of the day by day, showing the principals techniques, step by step, for the construction of a good use case.

**Keywords:** use cases, UML, software engineering



## I. INTRODUÇÃO

A UML (Unified Modeling Language) foi criada para auxiliar na resolução de problemas relacionados a projetos de desenvolvimento de software, oferecendo ferramentas essenciais, que admitem uma resposta profissional a desafios enfrentados nesse segmento.

Fazem parte da UML nove diagramas-padrão, dentre eles o Diagrama de Casos de Uso, que representa o sistema com suas funcionalidades e relacionamentos. Porém, o Diagrama de Casos de Uso é apenas a representação visual do caso de uso, um documento narrativo que descreve a seqüência de eventos de um ator (um agente externo) que usa um sistema para completar um processo.

O objetivo do trabalho desenvolvido é, além de introduzir e conceituar aspectos sobre casos de uso, indicar diretrizes que facilitarão a construção de casos de uso eficientes.

#### 2. UML

Durante anos, em projetos na área de tecnologia, as mudanças fazem parte da rotina dos envolvidos, é uma constante que faz com que as pessoas sejam jogadas para várias direções, existe um leque de diversidade, de possibilidades onde o profissional da área pode seguir, porém, mesmo com tais mudanças, existem alguns pontos que sempre geram polêmica e questionamentos, como:

- Requisitos mal definidos;
- Requisitos que mudam constantemente;
- Dificuldades entre os membros das equipes de TI, comercial e clientes;
- Estimativas em que não se pode confiar porque, na realidade, são suposições individualizadas, já que ninguém manteve qualquer medida do desenvolvimento (Pender, 2004).

A Unified Modeling Language (linguagem de modelagem unificada) foi criada para auxiliar na resolução destes pontos, oferecendo ferramentas essenciais, que admitem uma resposta profissional a esses desafios.

A finalidade da UML é proporcionar um pa-

drão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais, como processos de negócios e funções de sistema, além de itens concretos, como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de software reutilizáveis (BOOCH; RUMBAUGH; JACOBSON, 1999).

A UML oferece:

- Uma forma de comunicação coerente, que funciona igualmente bem na análise e no projeto;
- Uma apresentação visual que funciona igualmente bem para membros das equipes técnica e não técnica:
- Um padrão formal, porém flexível, para garantir a coerência e a clareza;
- Uma linguagem extensível, que pode ser ajustada a qualquer setor ou tipo, de aplicação;
- Uma maneira de especificar o software, independente da linguagem de programação;
- Um meio rigoroso de especificar a estrutura e o comportamento do software, a fim de gerar código ou até mesmo gerar aplicações completas (Pender, 2004).

#### 3. DIAGRAMA DE CASOS DE USO

Para uma representação visual, temos o Diagrama de Casos de Uso, que é o diagrama mais geral e informal da UML, usado normalmente nas fases de Levantamento e Análise de Requisitos do sistema, onde são determinadas as necessidades do usuário, embora venha a ser consultado durante todo o processo de modelagem e sirva de base para todos os outros diagramas. Apresenta uma linguagem simples e de fácil compreensão para que os usuários possam ter uma idéia geral de como o sistema irá se comportar. Procura identificar os atores (usuários, outros sistemas e até mesmo algum hardware especial) que serão utilizados, de alguma forma, no software, bem como os serviços, ou seja, as opções que o sistema disponibilizará aos atores, conhecidas neste diagrama como caso de uso (GUEDES, 2005).



Diagramas de casos de uso são valiosos porque:

- Identificam as expectativas dos clientes para o sistema:
  - Identificam recursos específicos do sistema;
- Identificam o comportamento compartilhado entre os recursos do sistema:

· Oferecem um modo simples e fácil de entender, para os clientes verem seus requisitos.

Segundo Pender (2004), seis elementos de modelagem compõem o Diagrama de Casos de Uso: atores, caso de uso, associações, relacionamentos <<include>> e <<extend>>, e generalização. Esses elementos estão ilustrados na Figura I.

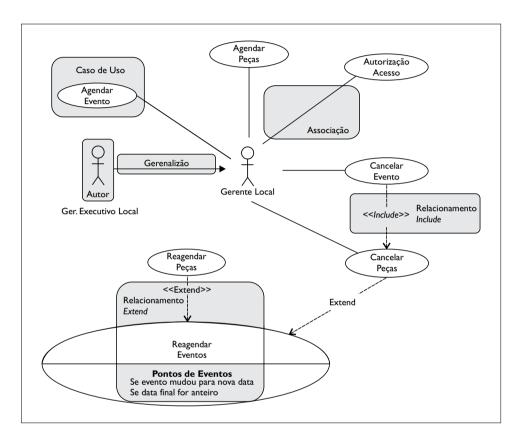


Figura I – Elementos do Diagrama de Caso de Uso. Fonte: Pender, 2004. p. 314.

#### 4. CASOS DE USO

Os casos de uso foram propostos inicialmente por Ivar Jacobson, em sua metodologia de desenvolvimento de sistemas orientados a objetos OOSE. Posteriormente, foi incorporada à UML, tornando seu uso uma prática frequente na identificação de requisitos de um sistema. Nesse contexto, um caso de uso descreve um comportamento que o software a ser desenvolvido apresentará quando estiver pronto (MEDEIROS, 2005).

Um caso de uso é um documento narrativo que descreve a sequência de eventos de um ator (um

agente externo), que usa um sistema para completar um processo (Jacobson, 1992). Eles são histórias ou casos de utilização de um sistema. Casos de uso não são exatamente especificação de requisitos ou especificação funcional, mas ilustram e implicam requisitos na história que eles contam (LARMAN, 2000).

Casos de uso são fundamentalmente uma forma contextual, embora possam ser escritos usando fluxogramas, diagramas de seqüência ou linguagens de programação.

Casos de uso podem ser utilizado em diferen-

tes situações, incluindo as seguintes:

- Para descrever um processo de trabalho de um negócio;
- Para focar discussão sobre futuros requisitos de software, mas não para ser a descrição dos requisitos;
- Para ser os requisitos funcionais de um sistema;
  - Para documentar o projeto do sistema;
- Para ser escrito em um grupo pequeno e restrito, ou em um grupo grande ou distribuído.

Os casos de uso têm vários objetivos importantes, como:

- Serem compreensíveis para usuários que, provavelmente, não entendem de informática;
- •Incentivarem a análise do sistema, especificando as funcionalidades necessárias:
  - Delimitarem o sistema;
  - Servirem de base para os casos de teste.

Casos de uso precisam ser compreensíveis pelos usuários, pois somente eles sabem o que o sistema precisa fazer. Os casos de uso permitem verificar se o desenvolvedor e o usuário concordam sobre o que o sistema deve fazer. Isso é um problema importante no desenvolvimento de software (WIKPÉDIA, 2005).

# 5. DIRETRIZES PARA ELABORAÇÃO DE UM BOM CASO DE USO

Sendo o Caso de Uso um elemento muito importante em um projeto de desenvolvimento de software, deve-se tomar muito cuidado ao escrevêlo, a fim de que se obtenha um bom entendimento do mesmo. Para tanto, seguem algumas diretrizes, que ajudam a elaborar bons Casos de Uso.

# 5.1. Diretriz 1: use gramática simples

A estrutura da sentença deve ser simples:

Sujeito... verbo... objeto direto... frase preposicional...

Por exemplo:

# O sistema... deduz... a quantia... do saldo da conta.

É importante que se mencione esta diretriz porque muitas pessoas, acidentalmente, deixam de fora o sujeito, não deixando claro quem está controlando a ação. Se a sentença é mal formada, torna-se difícil acompanhar a história (COCKBURN, 2005).

# 5.2. Diretriz 2: mostre claramente que ator executa qual ação

A cada passo de um cenário, um ator executa uma determinada ação. É muito importante que se deixe bem claro qual ator está tomando qual ação. Mais uma vez, um bom exemplo é um jogo de futebol: pessoa I passa a bola para a pessoa 2; pessoa 2 dribla e passa a bola para pessoa 3. Se a bola ficar suja de lama, um dos jogadores irá limpá-la (COCKBURN, 2005).

### 5.3. diretriz 3: escreva com uma visão geral

Segundo Cockburn (2005), autores de casos de uso iniciantes, particularmente programadores que estão escrevendo o documento de caso de uso, freqüentemente escrevem o cenário como visto pela visão de mundo do sistema e conversando consigo mesmo. Suas sentenças têm a aparência "Pegar o cartão do caixa eletrônico e a senha" ou "Deduzir quantia do saldo da conta". Em vez disso, escreva o caso de uso com uma visão geral:

O cliente insere no caixa eletrônico o cartão e a senha. O sistema deduz a quantia do saldo da conta.

Existe também um outro estilo de descrição de cenário que têm a característica de uma peça, descrevendo atores executando suas partes:

Cliente: insere no caixa eletrônico o cartão e a senha. Sistema: deduz a quantia do saldo da conta.

Observe que a informação é a mesma em ambos os estilos.

### 5.4. Diretriz 4: mostre a intenção dos atores, não os movimentos

Descrever os movimentos do usuário operando a interface de usuário do sistema é um dos erros mais comuns e graves na escrita dos casos de uso e está relacionado a escrever os objetivos em um nível muito baixo. Esta ação pode ser chamada de uma descrição de detalhes da interface. Descrições de detalhe da interface tornam o documento de requisitos pior em três aspectos: maior, frágil e muito restrito.

### 5.5. Diretriz 5: "valide", não "verifique se"

Um dos três tipos de passos de ação é a verificação do sistema de que alguma regra de negócio é satisfeita. Freqüentemente pessoas escrevem que o sistema verifica a condição. Esse não é um bom verbo de ação. Ele não leva distintamente o processo adiante, não é realmente o objetivo e deixa em aberto qual é o resultado da verificação.

É recomendado que se utilize a técnica de perguntar o porquê, para encontrar uma frase melhor. Por que o sistema está verificando a condição? Resposta: ele está determinando ou validando ou assegurando algo. Esses são bons verbos de ação para realização de objetivo. Substitua "O sistema verifica se a senha está correta" por: "O sistema verifica que a senha está correta" (COCKBURN, 2005).

# 5.6. Diretriz 6: mencione o tempo opcionalmente

A maioria dos passos segue diretamente do anterior. Ocasionalmente, escrever algo como "A qualquer hora, entre os passos 3 e 5, o usuário irá..." ou "Assim que o usuário tiver... o sistema irá", se fará necessário.

Não há problema em colocar o tempo, contanto que seja realmente necessário. Geralmente, o tempo é óbvio e não precisa ser mencionado (COCKBURN, 2005).

### 5.7. Diretriz 7: "usuário tem: sistema A chama sistema B"

Ocasionalmente, pode-se querer um sistema em desenvolvimento (A) para recuperar informação de um sistema (B) ou para executar uma interação com ele. Isso só deveria ocorrer quando o ator primário indica que o tempo está correto. Não podemos escrever Usuário pressiona botão Recuperar, no momento em que o sistema recupera os dados do sistema B, pois estaríamos descrevendo os detalhes de interface.

Para esta situação, pode-se usar dois passos:

- Usuário sinaliza ao sistema para recuperar dados do sistema B.
- 2. O sistema recupera os dados secundários do sistema B.

Porém, embora aceitável, eles são ineficazes e redundantes. Neste caso, seria melhor escrever:

Usuário tem o sistema recuperando os dados secundários do sistema B.

Com essa pequena mudança, indicamos que o usuário controla o tempo e que a ação passa do usuário para o sistema A e para o sistema B, e mostra as responsabilidades dos três sistemas. Os detalhes de como o usuário inicia a ação não são especificados, como de fato não deveriam ser (COCKBURN, 2005).

# 5.8. Diretriz 8: expressão idiomática "faça passos x-y até condição"

Pode-se querer marcar que alguns passos podem ser repetidos, quando for necessário. Nesta situação, novamente vemos o benefício de se estar escrevendo em prosa simples ao invés de formalismos de programação. Apenas escreva que o passo ou os passos irão se repetir.

Se há apenas um passo a ser repetido, pode-se colocar a repetição direto na descrição do passo (COCKBURN, 2005).



## 6. CONSIDERAÇÕES FINAIS

A utilização da UML é um dos fatores de grande influência no sucesso de um projeto de desenvolvimento de software orientado a objetos nos dias atuais, sendo que os casos de uso são um dos pontos mais importantes deste processo. Contudo, é necessário que esses casos de uso sejam bem elaborados e atinjam os objetivos a que se destinam, ou seja, apresentar, de forma clara e objetiva, a forma como o sistema será utilizado, de fato.

Para se construir tais casos de uso, devemos seguir algumas diretrizes imprescindíveis, como por exemplo, a utilização de uma gramática simples, clareza, visão geral do sistema, intenções, validações etc., para que sua concepção seja coerente, funcional e clara para todos os envolvidos no projeto, desde usuários até desenvolvedores.

# REFERÊNCIAS BIBLIOGRÁFICAS

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. The Unified Modeling Language user guide, Addison-Wesley, 1999.

BROERING. MDA - Model Driven Architecture. 2/7/2005. Disponível em http://www.portaljava.com/ home/modules.php?name=News&file=article&sid=419.Acessado em 7/9/2005.

|ACOBSON, I.; BOOCH, G; RUMBAUGH, |. The unified software development process. Boston: Addison-Wesley, 1999.

LARMAN, Craig. Applying UML and patterns - an introduction to object-oriented analysis and design. Prentice-Hall, 1998.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos. Porto Alegre: Bookman, 2000.

MEDEIROS, Ernani. Desenvolvendo software com UML 2.0. Pearson Makron Books, 2004.

MULLER, P.A. Instant UML. Wrox Press, 1997.

NOGUEIRA, Admilson. UML. 21/2/2005. Disponível em http://www.imasters.com.br/artigo. php?cn=2994&cc=204.Acesso em 15/6/2005.

PONTIFÍCIA UNIVERSIDADE CATÓLICA. OMG. Rio de Janeiro, 2005. Disponível em http://www.inf. puc-rio.br/~casanova/ReferenciasBD/corba-Pires.pdf. Acesso em 22/8/2005.

RUMBAUGH, J; JACOBSON, I.; BOOCH, G. The Unified Modeling Language reference manual. Boston: Addison-Wesley, 1999.

TAVARES, António. UML - visão geral. Portugal, 2003. Disponível em http://www.ipca.pt/prof/ajtavares/disciplinas/aesi02-03/download/aulas/AESI%202002-03%20(06)%20UML%20-%20visao%20geral.pdf. Acesso em 20/9/2005.

WIKIPEDIA. Caso de uso. 4/10/2005. Disponível em http://pt.wikipedia.org/wiki/Caso de uso. Acesso em 13/10/2005.