

## OORTs/ProDeS: Definição de Técnicas de Leitura para um Processo de Software Orientado a Objetos

Regiane Aparecida Marucci  
Sandra C. P. F. Fabbri

Universidade Federal de  
São Carlos  
Departamento de Computação  
Rodovia Washington  
Luis, km 235  
Cx. Postal 676, 13565-905,  
São Carlos, SP, Brasil  
{marucci, sfabbri}@dc.ufscar.br

José Carlos Maldonado

Universidade de São Paulo  
Instituto de Ciências  
Matemáticas e Computação  
Av. do Trabalhador São-  
Carlense, 400, Centro,  
Cx. Postal 668, 13560-970,  
São Carlos, SP, Brasil  
jcmaldon@icmc.sc.usp.br

Guilherme H. Travassos

Universidade Federal do  
Rio de Janeiro  
Programa de Engenharia de  
Sistemas e Computação  
Cx. Postal 68511,  
21945-970,  
Rio de Janeiro, RJ, Brasil  
ght@cos.ufrj.br

### Resumo

Observa-se que a atividade de inspeção e as técnicas de leitura associadas são, em geral, discutidas sem considerar explicitamente um processo subjacente de desenvolvimento de software. Em particular, quando se considera a abordagem OO (Orientada a Objetos), as técnicas são, em geral, associadas às notações definidas pela UML (*Unified Modeling Language*). O fato da UML não estar vinculada a um processo de desenvolvimento dificulta a definição e implantação de atividades de garantia de qualidade de software, em especial, atividades de VV&T (Validação, Verificação e Teste). A falta de um processo permite uma livre escolha do conjunto de diagramas selecionados para o desenvolvimento de uma aplicação, além do que a forma de utilização dos mesmos e a interpretação dos conceitos da própria UML podem variar dentro de um mesmo ambiente de desenvolvimento. Ressalta-se ainda que a atividade de inspeção é essencial para que um determinado processo atinja o nível 3 do CMM (*Capability Maturity Model*). As técnicas de leitura OORTs (*Object Oriented Reading Techniques*) foram definidas para um subconjunto da notação UML, considerando um processo de software genérico e simplificado e se aplicam essencialmente na fase de projeto. Neste artigo define-se um conjunto de técnicas de leitura para apoiar atividades de inspeção em um processo de desenvolvimento de software OO baseado na notação UML, denominado ProDeS/UML, que inclui atividades de teste ao longo de suas fases de desenvolvimento. Esse conjunto de técnicas de leitura denomina-se OORTs/ProDeS e consiste da integração das técnicas OORTs e da definição de novas técnicas de leitura no contexto do processo ProDeS/UML.

**Palavras-Chave:** Garantia de Qualidade de Software, VV&T, Inspeção, Processo de Software OO, UML.

### Abstract

Usually the establishment of inspection activities and of reading techniques is discussed without explicitly considering a software development process. Particularly, for the OO (Object Orientation) approach the techniques are associated with the UML (Unified Modeling Language) notation. The fact that the UML being independent of a particular development method or process makes difficult defining and implementing the software quality assurance activities, such as VV&T (Validation, Verification and Testing) activities. The lackness of a process allows that the set of UML diagrams can be freely chosen by the developer for a specific application development. Moreover, the way the diagrams are used and the interpretation of the UML concepts may vary inside the same development environment. It should be pointed out that the inspection activity is one of the essential activities required to a process be considered CMM level 3. The OORTs (Object Oriented Reading Techniques) are defined for a subset of the UML notation, considering a generic, simplified software process and are to be applied, essentially, in the design phase. In this paper it is defined a set of reading techniques to integrate inspection activities into an OO software development process based on the UML notation, named ProDeS/UML, which already defines testing activities along its development phases. This set of reading techniques is named OORTs/ProDeS and consists of both the integration of OORTs and the definition of new reading techniques in the context of ProDeS/UML.

**Keywords:** Software Quality Assurance, VV&T, Inspection, OO Software Process, UML.

## 1. Introdução

Desde sua padronização pelo *Object Management Group* (OMG) em 1997, a UML (*Unified Modeling Language*) [1] tem sido amplamente adotada na indústria de software e se tornou uma notação de modelagem bastante utilizada para a análise e projeto Orientados a Objetos (OO). No entanto, a UML apenas padroniza alguns artefatos e suas notações sem definir um processo de desenvolvimento. Em muitos casos, a falta de definição de um processo de desenvolvimento faz com que os recursos propostos pela UML sejam utilizados de forma não uniforme entre os projetos, além de tornar mais difícil a manutenção da homogeneidade dos diagramas utilizados e construídos. Em decorrência disso, o estabelecimento e a aplicação de Atividades de Garantia de Qualidade de Software, como por exemplo, estratégias de Validação, Verificação e Teste (VV&T), são fundamentais e permitem garantir que um mesmo conjunto de diagramas seja utilizado de forma consistente.

A crise da qualidade da década de 90 evidenciou que a qualidade do produto está diretamente relacionada à qualidade do processo pelo qual ele é desenvolvido [2]. Para os modelos de qualidade de processo como o CMM (*Capability Maturity Model*) [3] ou a Norma ISO 12.207 [4] a definição de um processo de desenvolvimento de software é um ponto fundamental quando se deseja realizar atividades de melhoria na maneira como o software é desenvolvido. Não menos importante que a definição do processo, Atividades de Garantia de Qualidade de Software também constituem um item bastante relevante em relação aos modelos e normas utilizados atualmente. Ressalta-se, por exemplo, que a atividade de inspeção é essencial para que um determinado processo atinja o nível 3 do CMM.

Segundo Andriole [5], uma abordagem sistemática para o desenvolvimento de software deve ser feita através de uma seqüência de fases que permitam um retorno às fases anteriores, pois isto traz uma série de benefícios como, por exemplo, permite uma abordagem de solução gradual para o problema, o estabelecimento de pontos intermediários de monitoração e controle e uma avaliação desde o início de forma continuada da solução do problema. Juntamente com essas fases, Andriole discute atividades de VV&T como uma metodologia que ajuda assegurar a produção de software com qualidade. Tais atividades mantêm o foco na prevenção e detecção de defeitos, isto é, nos desvios daquilo que se tem intenção de fazer e podem ser classificadas em três categorias: i) checagem de integridade, cujo objetivo é verificar a integridade dos produtos em cada fase de desenvolvimento, analisando sua consistência interna e sua completitude; ii) checagem de evolução, cujo objetivo é assegurar a completitude e consistência entre níveis de especificação, sendo que o segundo nível é um refinamento do primeiro e iii) checagem de adequação e suficiência, cujo objetivo é comparar a solução com a compreensão que se tem do problema no momento corrente, para assegurar que a solução seja suficiente e a desejada.

No contexto da abordagem de desenvolvimento de software OO, foram definidas as técnicas de leitura OORTs (*Object Oriented Reading Techniques*) para um subconjunto da notação UML, considerando um processo de software genérico e simplificado que se aplicam essencialmente na fase de projeto em relação ao processo ProDeS/UML [6].

Sendo assim, considerando-se o amplo uso do paradigma de desenvolvimento OO e da linguagem UML, a importância da definição e uso de um processo de desenvolvimento de software que englobe atividades de garantia de qualidade e inspirando-se nas técnicas OORTs, o objetivo deste artigo é apresentar um conjunto de técnicas de leitura no contexto de um processo de desenvolvimento de software OO, denominado ProDeS/UML [7]. O

ProDeS/UML é baseado no método Fusion [8], usa a notação UML na confecção dos artefatos produzidos e define atividades de teste ao longo de todas as fases de desenvolvimento.

Na Seção 2 apresenta-se uma visão do ProDeS/UML. Na Seção 3, discute-se a viabilidade de aplicação das técnicas de leitura OORTs no processo ProDeS/UML e apresentam-se novas técnicas que foram definidas para inspeção dos artefatos gerados por esse processo. Referencia-se por OORTs/ProDeS ao conjunto de todas essas técnicas utilizadas no contexto do ProDeS/UML. Na Seção 4, comenta-se sobre um estudo de caso que foi realizado para uma avaliação preliminar de algumas dessas técnicas. Na Seção 5, apresentam-se as conclusões e trabalhos futuros.

## 2. ProDeS/UML

O ProDeS/UML [7] é um processo de desenvolvimento para software OO, apoiado pela UML [1]. Ele é composto por quatro fases: Engenharia de Requisitos, Análise, Projeto e Implementação, como pode ser visto na Figura 1. Observe que esse processo possui agregado a ele uma estratégia de teste que propõe a elaboração de Modelos de Teste com base em alguns dos artefatos que são produzidos em cada fase. Neste artigo, propõem-se e agregam-se técnicas de leitura a esse processo.

A Tabela 1 apresenta a relação entre os artefatos produzidos no ProDeS/UML e a sua correspondência com artefatos propostos pela notação UML. O Modelo de Operações e a Descrição das Classes são oriundos do método Fusion e, em geral, fazem parte do Dicionário de Dados.

Na *Fase de Engenharia de Requisitos* o objetivo é descrever o que o sistema deve fazer e permitir que os desenvolvedores e clientes concordem com essa descrição. Os modelos gerados nessa fase são: diagrama de casos de uso e suas especificações que correspondem à compreensão dos requisitos do sistema com base no documento de requisitos; diagrama de classes do domínio que contém as classes candidatas do sistema e os relacionamentos existentes entre elas.

Na *Fase de Análise* o objetivo é especificar o comportamento esperado do sistema. Para isso, é definida a interface do sistema com o ambiente, por intermédio das operações que podem ser executadas. Além disso, os limites do sistema também são definidos nessa fase, excluindo-se algumas classes e relacionamentos e dando origem aos agentes externos e às classes de interface. Os modelos gerados nessa fase são: os cenários de uso do sistema que podem ser elaborados para detectar as operações e os eventos de saída do sistema; diagrama de classes da análise que corresponde ao diagrama de classes do domínio gerado na fase anterior, acrescido das classes necessárias para a implementação do software; modelo de operações que especifica o comportamento de uma operação individual de forma declarativa em termos de mudança de estado do sistema e geração de eventos de saída e modelo de ciclo de vida que descreve o comportamento completo de como o sistema se comunica com o ambiente, desde sua criação até seu término.

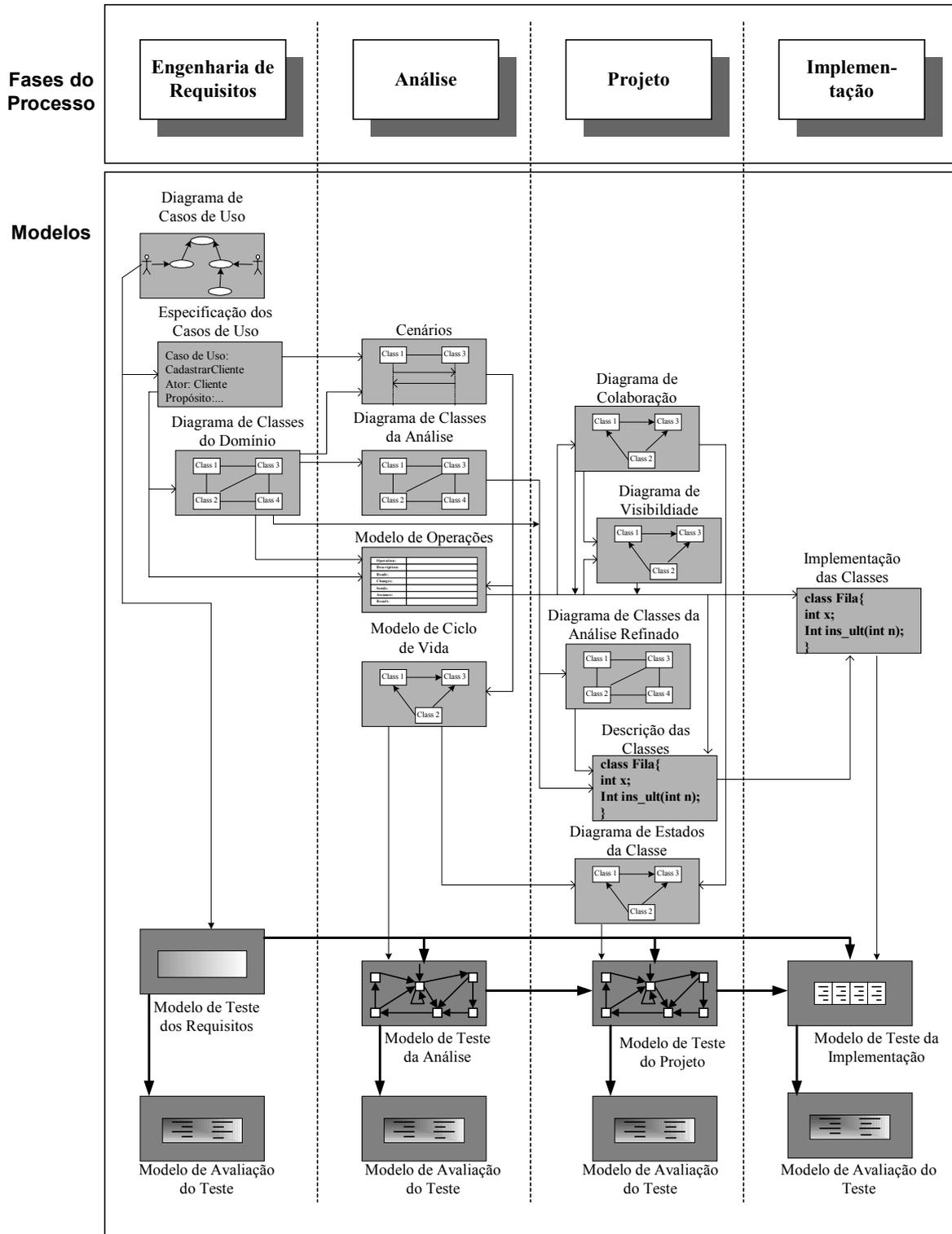


Figura 1. ProDeS/UML [7].

O objetivo fundamental da *Fase de Projeto* é especificar como as funcionalidades do sistema serão implementadas pela interação dos objetos do sistema. Assim, cada operação é especificada de modo a mostrar como ela pode ser implementada por meio da interação de vários objetos que trocam mensagens entre si. Os modelos gerados nessa fase são: diagramas de colaboração que especificam como as classes interagem entre si para a execução de uma operação; diagrama de visibilidade que permite identificar a necessidade de referências entre classes; diagrama de classes da análise refinado que constitui do diagrama de classes da fase anterior, acrescido dos objetos, relações de generalização/especialização e herança, bem como os métodos; descrição das classes que corresponde à descrição da estrutura interna de cada classe, juntamente com seus atributos de dados e de objetos, seus métodos e sua posição hierárquica, caso exista uma hierarquia de herança e diagramas de estados das classes que descrevem o comportamento de uma classe em particular, desde sua criação até seu término.

**Tabela 1. Correspondência entre os diagramas do ProDeS/UML e a notação UML/Fusion.**

Fases do ProDeS/UML	Diagramas elaborados no ProDeS/UML	Diagramas correspondentes	Notação	
			UML	Fusion
Engenharia de Requisitos	Diagrama de Casos de Uso	Diagrama de Casos de Uso	X	
	Especificação dos Casos de Uso	Diagrama de Casos de Uso	X	
	Diagrama de Classes do Domínio	Diagrama de Classes	X	
Análise	Cenários	Diagrama de Seqüência	X	
	Diagrama de Classes da Análise	Diagrama de Classes	X	
	Modelo de Operações	Modelo de Operação		X
	Modelo de Ciclo de Vida	Diagrama de Estados	X	
Projeto	Diagrama de Colaboração	Diagrama de Colaboração	X	
	Diagrama de Visibilidade	Diagrama de Classes	X	
	Diagrama de Classes da Análise Refinado	Diagrama de Classes	X	
	Descrição das Classes	Descrição de Classe		X
	Diagrama de Estados da Classe	Diagrama de Estados	X	
Implementação	Implementação das Classes	-		

### 3. OORTs/ProDeS: Técnicas de Leitura para Inspeção dos Artefatos gerados pelo ProDeS/UML

Com o intuito de agregar ao ProDeS/UML, além da abordagem de teste já existente, atividades de inspeção que possam assegurar uma maior qualidade aos artefatos produzidos ao longo desse processo, fez-se, inicialmente, um estudo sobre a viabilidade de aplicação de técnicas de leitura já existentes nas diversas fases do ProDeS/UML, em especial as técnicas OORTs.

As técnicas de leitura OORTs (*Object Oriented Reading Techniques*) [6] foram definidas para um subconjunto da notação UML, considerando um processo de software genérico e simplificado e se aplicam essencialmente nos artefatos da fase de projeto do processo ProDeS/UML. As OORTs compreendem basicamente sete técnicas de leitura, para documentos UML distintos, referenciadas por (P1 a P7 na Figura 2). Essas técnicas constituem diretrizes de procedimentos que ajudam o leitor direcionar apropriadamente a sua revisão (leitura) ao examinar um determinado artefato de software com o objetivo de identificar defeitos. Elas são classificadas como Horizontal ou Vertical, sendo que na leitura

horizontal, os documentos comparados pertencem à mesma fase de projeto, caracterizando então uma atividade de verificação e na vertical, os documentos da fase de projeto são comparados com aqueles elaborados nas fases de especificação de requisitos e engenharia de requisitos, a fim de verificar se os requisitos foram mantidos durante a descrição do sistema pelos outros artefatos propostos para essa descrição. Dentre essas técnicas, exceto a P6 que não envolve o documento de requisitos e, portanto, é uma técnica de verificação, as demais são técnicas de validação.

Considerando-se as fases de Engenharia de Requisitos, de Análise e de Projeto do ProDeS/UML e considerando-se a abordagem para desenvolvimento de software e a estratégia de VV&T propostas por Andriole [5] foram definidas técnicas de leitura que, juntamente com as OORTs, permitem uma avaliação intermediária dos artefatos da aplicação que está sendo desenvolvida, desde as primeiras fases desse processo, como pode ser observado na Figura 2.

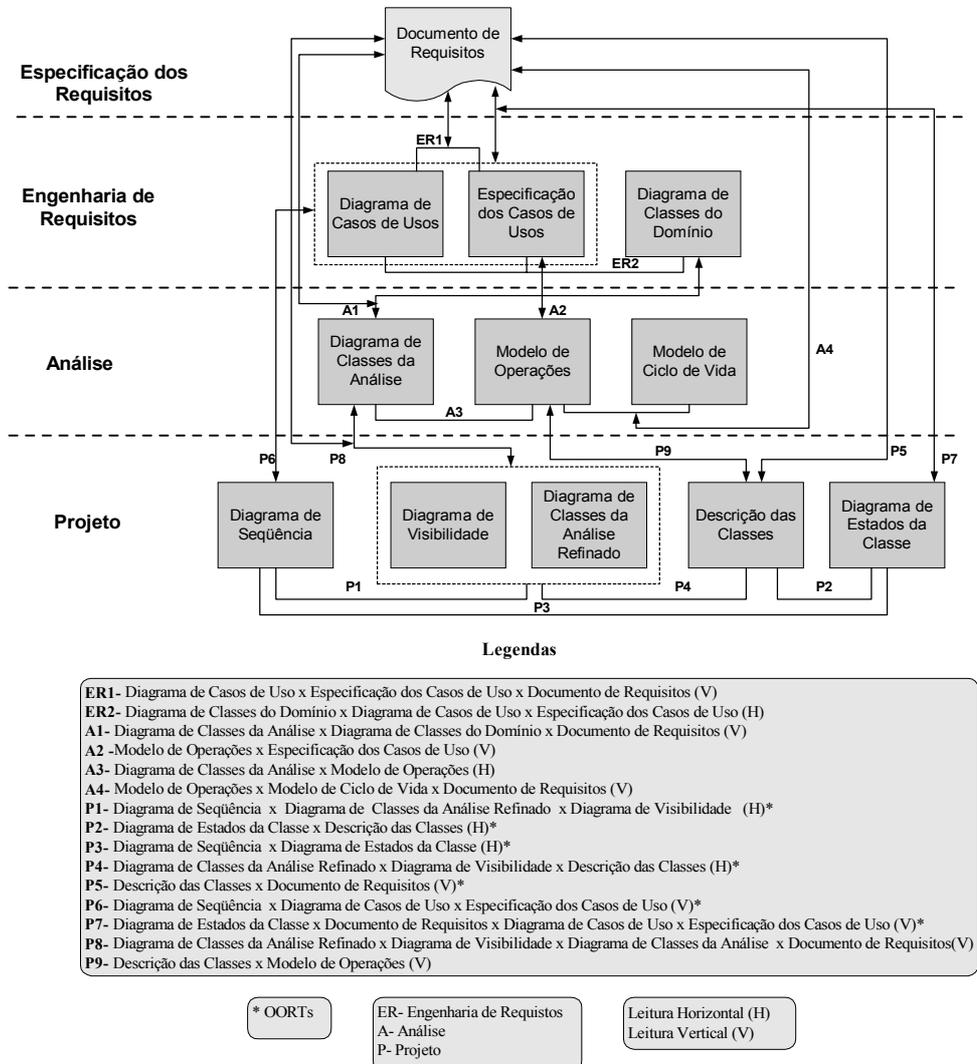


Figura 2. Técnicas de Leitura para o processo ProDeS/UML.

Apresenta-se a seguir um estudo detalhado sobre a viabilidade de aplicação das técnicas de leitura OORTs no ProDeS/UML e, posteriormente, apresentam-se as novas técnicas elaboradas para outras fases e artefatos não cobertos pelas OORTs. Ao conjunto total dessas técnicas denomina-se OORTs/ProDeS e, para se fazer uma avaliação preliminar de algumas das técnicas desse conjunto, foi realizado um estudo de caso, cujos resultados são comentados na Seção 4. A definição das técnicas OORTs/ProDeS foi inspirada no conjunto das técnicas P1 a P7, denominadas OORTs [6], as quais abordam apenas alguns dos diagramas da UML utilizados no ProDeS/UML. Como pode ser observado pela Figura 2, esses diagramas correspondem àqueles que estão envolvidos nas leituras P1 a P7 e são os seguintes: diagrama de casos de uso, diagrama de classes acompanhado de sua descrição, diagramas de estado e diagramas de seqüência representando os diagramas de interação.

### 3.1. Avaliação da Aplicação das OORTs nos Artefatos de Projeto do ProDeS/UML

Para avaliar a viabilidade de se aplicar OORTs no contexto do ProDeS/UML realizou-se inicialmente um mapeamento entre a correspondência dos artefatos utilizados pelas técnicas de leitura orientadas a objetos com aqueles produzidos ao longo do processo. A Tabela 2 apresenta esta relação. Nota-se que dois deles são construídos durante a fase de Engenharia de Requisitos e os demais, na fase de Projeto do ProDeS/UML.

**Tabela 2. Artefatos utilizados para a aplicação das OORTs e sua correspondência com os artefatos produzidos no ProDeS/UML.**

OORTs	ProDeS/UML	
Artefatos utilizados	Artefatos Elaborados	Fases
Casos de Uso	Diagrama de Casos de Uso	<i>Engenharia de Requisitos</i>
	Especificação dos Casos de Uso	
Diagrama de Seqüência	Diagrama de Colaboração	<i>Projeto</i>
Diagrama de Classes	Diagrama de Visibilidade	
	Diagrama de Classes da Análise Refinado	
<i>Template</i> para Descrição das Classes	Descrição das Classes	
Diagrama de Estados	Diagrama de Estados da Classe	

Dentre esses artefatos, tem-se que o Diagrama de Casos de Uso, as Especificações dos Casos de Uso e os Diagramas de Estados das Classes do ProDeS/UML não necessitam de adaptações para a aplicação das OORTs. No caso dos dois primeiros, Diagrama e Especificações dos Casos de Uso, a informação que eles contêm corresponde àquela contida no Caso de Uso da UML, utilizado pelas OORTs, com a diferença de que nas Especificações do ProDeS/UML é utilizado um padrão predefinido para descrever as funcionalidades do sistema, mas que não traz impedimento para a aplicação direta da técnica de leitura.

No entanto, para a aplicação das OORTs nos demais artefatos da Fase de Projeto do ProDeS/UML, algumas adaptações são necessárias:

- i) Como o ProDeS/UML propõe a elaboração de diagramas de colaboração, é necessário gerar, a partir desses, os respectivos diagramas de seqüência. Observa-se que, para facilitar a aplicação das leituras, o ideal é que cada diagrama de seqüência represente um único

curso de execução (o curso normal e os cursos alternativos). Além disso, tendo-se a oportunidade de usar uma ferramenta de apoio do tipo da *Rational Rose* [9], a geração dos diagramas de seqüência é feita automaticamente, partindo-se do diagrama de colaboração.

- ii) O diagrama de visibilidade e o diagrama de classes da análise refinado devem ser utilizados em conjunto, uma vez que assim representam todas as características de um diagrama de classes da UML.
- iii) A descrição de classes produzida pelo ProDeS/UML deve ser adaptada ao *template* utilizado para a aplicação das OORTs, o qual corresponde à documentação do diagrama de classes registrada junto à ferramenta *Rational Rose* [9]. Optou-se por esse *template* por ele apresentar-se mais completo do que a descrição de classes proposta pelo ProDeS/UML, tendo em vista abranger mais características durante o processo de inspeção.

Feitas essas adequações, torna-se possível a aplicação das técnicas de leitura OORTs para se fazer inspeção nos artefatos gerados pelo ProDeS/UML. Ressalta-se que a viabilidade de aplicação dessas técnicas acontece quando se chega na fase de projeto do ProDeS/UML e não antes disso.

Além do estudo sobre a viabilidade de aplicação das OORTs nos artefatos da Fase de Projeto, verificou-se também a viabilidade de aplicação dessas técnicas nos artefatos das fases de Engenharia de Requisitos e Análise não abordados por elas. A Tabela 3 apresenta os artefatos dessas fases e a que tipo de diagrama UML eles correspondem, no contexto das técnicas OORTs.

Embora aparentemente haja uma forte correspondência entre os tipos dos artefatos gerados pelo ProDeS/UML e utilizados nas leituras propostas pelas técnicas OORTs (P1 a P7 da Figura 2), alguns desses artefatos refletem características da notação Fusion, como por exemplo, os Cenários, o Modelo de Ciclo de Vida e o Modelo de Operações, o que inviabiliza uma aplicação direta das técnicas existentes para se fazer a inspeção desses documentos.

**Tabela 3. Correspondência entre os artefatos elaborados pelo ProDeS/UML (fases de Engenharia de Requisitos e Análise) e os artefatos utilizados pelas OORTs.**

ProDeS/UML		Correspondência com tipos de artefatos utilizados pelas OORTs
Fases	Artefatos Elaborados	
<i>Engenharia de Requisitos</i>	Diagrama de Classes do Domínio	Diagrama de Classes
<i>Análise</i>	Cenários	Diagramas de Seqüência
	Diagrama de Classes da Análise	Diagrama de Classes
	Modelo de Operações	
	Modelo de Ciclo de Vida	Diagrama de Estados

Assim, como pode ser observado pela Tabela 3, embora os Cenários sejam especificados como diagramas de seqüência, o objetivo deles é mostrar apenas a interação entre os atores e o sistema, diferenciando-se do formato utilizado para a aplicação das OORTs, nas quais a interação no diagrama de seqüência ocorre entre os atores e as classes que contribuem para a realização de uma determinada funcionalidade do sistema.

O Modelo de Ciclo de Vida, por sua vez, ainda que seja representado como um diagrama de estados, tem por objetivo mostrar o ciclo de vida do sistema e não de cada classe, como é necessário para a aplicação das OORTs.

Já o Modelo de Operações especifica cada operação (caso de uso) individual de forma declarativa em termos de mudança de estado do sistema e eventos de saída. Para sua especificação é utilizada a notação Fusion, uma vez que a UML não define uma padronização para modelagem de informação textual.

Os Diagramas de Classes do Domínio e da Análise, embora apresentem as características de um diagrama de classes da UML, constituem versões mais abstratas do Diagrama de Classes da Análise Refinado e não precisam, obrigatoriamente, considerar os métodos da classe. Tal informação é modelada apenas na Fase de Projeto sendo que, para a aplicação das OORTs, ela é indispensável para que todos os aspectos propostos pelas leituras que utilizam esse artefato possam ser analisados.

Portanto, de acordo com as considerações apresentadas anteriormente, pode-se notar que embora a UML seja definida como uma notação padrão e o processo ProDeS/UML utilize, prioritariamente, essa notação para especificar os artefatos, ainda assim, a representação de alguns conceitos pode diferenciar.

Dessa forma, após esse estudo de viabilidade da aplicação das OORTs nos artefatos gerados no ProDeS/UML verificou-se que tanto a fase de Engenharia de Requisitos como a fase de Análise, necessitavam da proposta e definição de novas técnicas de leitura para que se pudesse implantar atividades de inspeção ao longo de todo o processo de desenvolvimento. Foram então definidas novas técnicas de leitura para abordar alguns documentos e aspectos relevantes produzidos nessas fases do processo. Tais técnicas são apresentadas na seção seguinte.

### **3.2. Definição de Novas Técnicas de Leitura para os Artefatos do ProDeS/UML**

Como mencionado anteriormente, de acordo com Andriole [5] o ideal é que se tenha, ao longo de todo o processo de desenvolvimento, atividades de validação e verificação. As atividades de validação têm como objetivo contrapor artefatos de software com o documento de requisitos, para assegurar que o software continue representando-os apropriadamente e também para assegurar que esses requisitos permaneçam atualizados durante todo o processo (checagem de adequação). As atividades de verificação têm como objetivo contrapor os diversos artefatos de software entre eles mesmos, para assegurar que eles estão sendo gerados de forma correta e consistente (checagem de integridade e de evolução). Baseando-se nessa abordagem e também na seqüência em que os diagramas são construídos ao longo das fases do ProDeS/UML, a qual pode ser observada pela direção das setas na Figura 2, definiu-se um conjunto de novas técnicas de leitura com o objetivo de manter a rastreabilidade das informações entre os documentos, bem como a consistência das informações entre eles e entre os requisitos. Na Tabela 4 apresentam-se essas técnicas, juntamente com o seu tipo e seu principal objetivo.

Observando-se na Tabela 4 os documentos envolvidos nas leituras que foram definidas e comparando-se com os documentos gerados no ProDeS/UML, nota-se que os Cenários, que são elaborados na fase de Análise são os únicos que não são abordados por nenhuma delas. Isso ocorreu pois considera-se que os Cenários são documentos que servem apenas de apoio para a construção do Modelo de Ciclo de Vida e do Modelo de Operações e que, portanto, não precisam de uma leitura especial para avaliar seu conteúdo. Observa-se também que, para a fase de Projeto, além das OORTs, são propostas mais duas leituras (P8 e P9) que têm como objetivo fazer a consistência entre alguns documentos da fase de Análise e de Projeto.

**Tabela 4. Técnicas de Leitura desenvolvidas para apoiar o ProDeS/UML.**

<b>Técnica</b>	<b>Documentos inspecionados</b>	<b>Tipo</b>	<b>Objetivo</b>
<b>ER1</b>	Diagrama de Casos de Uso x Especificação dos Casos de Uso x Documento de Requisitos	Validação	Verificar se os conceitos e funcionalidades que estão descritos pelo Documento de Requisitos foram capturados apropriadamente pelo Diagrama de Casos de Uso e se as Especificações dos Casos de Uso estão descritas de forma precisa.
<b>ER2</b>	Diagrama de Classes do Domínio x Diagrama de Casos de Uso x Especificação dos Casos de Uso	Verificação	Verificar se os conceitos (atores, classes candidatas ou atributos) e os relacionamentos representados no Diagrama de Classes do Domínio capturam apropriadamente os conceitos sobre os atores, entidades a serem mantidas no sistema e atributos relacionados a elas descritos no Diagrama e Especificações dos Casos de Uso.
<b>A1</b>	Diagrama de Classes da Análise x Diagrama de Classes do Domínio x Documento de Requisitos	Validação	Verificar se os conceitos (atores, classes ou atributos) e os relacionamentos representados no Diagrama de Classes da Análise estão compatíveis com as informações contidas no Diagrama de Classes do Domínio e se as novas informações presentes nesse diagrama estão condizentes com o Documento de Requisitos.
<b>A2</b>	Modelo de Operações x Especificação dos Casos de Uso	Verificação	Verificar se a descrição de cada operação representada pelo Modelo de Operações está condizente com a Especificação dos Casos de Uso.
<b>A3</b>	Diagrama de Classes da Análise x Modelo de Operações	Verificação	Verificar se as operações descritas no Modelo de Operações são compatíveis com os relacionamentos e conceitos (atores, classes ou atributos) representados no Diagrama de Classes da Análise.
<b>A4</b>	Modelo de Operações x Modelo de Ciclo de Vida x Documento de Requisitos	Validação	Verificar se o Modelo de Ciclo de Vida descreve apropriadamente os eventos que disparam a troca de estados do sistema, se esses eventos correspondem a cada operação descrita no Modelo de Operações e se o estado alcançado pelo sistema após a execução da operação é apropriado. Além disso, verificar também se as representações em ambos os artefatos estão precisas conforme a descrição contida no Documento de Requisitos.
<b>P8</b>	Diagramas de Classes da Análise Refinado x Diagrama de Visibilidade x Diagrama de Classes da Análise x Documento de Requisitos	Validação	Verificar se os conceitos (atores, classes ou atributos) e os relacionamentos representados no Diagrama de Classes da Análise Refinado estão compatíveis com as informações contidas no Diagrama de Classes da Análise e se as novas informações presentes nesse diagrama, bem como se as relações de dependência e navegabilidade do Diagrama de Visibilidade estão condizentes com o Documento de Requisitos.
<b>P9</b>	Descrição de Classes x Modelo de Operações	Verificação	Verificar se as Descrições das Classes contêm toda a informação necessária para realização das operações descritas pelo Modelo de Operações.

Similarmente às técnicas OORTs, a estrutura dessas técnicas foi elaborada como uma seqüência de passos, como pode ser observado na Figura 3.

**Leitura ER1 - Diagrama de Casos de Uso x Especificação dos Casos de Uso x Documento de Requisitos (V)**

**Objetivo:** Verificar se os conceitos e funcionalidades que estão descritos pelo Documento de Requisitos foram capturados apropriadamente pelo Diagrama de Casos de Uso e se as Especificações dos Casos de Uso estão descritas de forma precisa.

**Entradas para o processo:**

1. Um conjunto de Requisitos Funcionais e Não Funcionais.
2. Um Diagrama de Casos de Uso.
3. As Especificações dos Casos de Uso.

**I. Leia o Documento de Requisitos para entender a funcionalidade descrita.**

**ENTRADAS:** Um conjunto de Requisitos Funcionais (RFs) e Não Funcional (RNFs).

**SAÍDAS:** Substantivos candidatos a atores, entidades que serão mantidas (ou geradas) no sistema ou atributos das entidades (marcados em azul nos RFs);  
Verbos ou descrições de ações candidatos a funcionalidades do sistema (marcados em verde nos RFs);  
Restrições ou condições associadas aos substantivos ou verbos (marcadas em amarelo nos RFs e não funcionais).

Para cada Requisito Funcional do Documento de Requisitos faça:

- A. Leia o Requisito para entender a funcionalidade que ele descreve.
- B. Encontre os substantivos que constam desse Requisito e que sejam candidatos a tornarem-se atores ou entidades que serão mantidas (ou geradas) no sistema, ou atributos dessas entidades. Sublinhe esses substantivos com uma caneta azul.

...

**II. Identifique a funcionalidade descrita pela Especificação do Caso de Uso e os conceitos importantes do sistema que são necessários para alcançar essa funcionalidade.**

**ENTRADAS:** Diagrama de Casos de Uso (DCU);  
Especificação dos Casos de Uso (ECU).

**SAÍDAS:** Substantivos candidatos a atores, entidades que serão mantidas (ou geradas) no sistema ou atributos das entidades (marcados em azul na ECU);  
Verbos ou descrições de ações candidatos a funcionalidades do sistema (marcados em verde na ECU);  
Restrições ou condições associadas aos substantivos ou verbos (marcadas em amarelo na ECU);  
Relatório de Discrepâncias.

Para cada caso de uso do Diagrama de Casos de Uso faça:

- A. Encontre a Especificação do Caso de Uso correspondente.
  1. **Caso ela não seja encontrada, então existe um caso de uso no Diagrama de Casos de Uso que não foi especificado. Relate esse fato no Relatório de Discrepâncias.**
- B. Leia a Especificação do Caso de Uso identificada no passo anterior para entender a funcionalidade que ela descreve.

...

**III. Compare ambos os documentos para verificar se os conceitos descritos no Documento de Requisitos foram corretamente capturados pelo Diagrama e Especificação dos Casos de Uso.**

**ENTRADAS:** Diagrama de Casos de Uso (DCU);  
Especificação dos Casos de Uso com conceitos marcados;  
Requisitos Funcionais com conceitos marcados.

**SAÍDAS:** Relatório de Discrepâncias.

Para cada Requisito Funcional do Documento de Requisitos, que ainda não foi rotulado com um “número”, faça:

- A. Localize no Diagrama de Casos de Uso o caso de uso que está tratando desse Requisito Funcional e em seguida separe a respectiva Especificação do Caso de Uso.
  1. **Caso você não encontre o caso de uso que esteja tratando desse Requisito Funcional, preencha o Relatório de Discrepâncias, pois alguma funcionalidade foi omitida do Diagrama e Especificação de casos de uso.**
- B. Leia a Especificação do Caso de Uso selecionada no item anterior e verifique se ela não trata de outras funcionalidades além daquela selecionada inicialmente no Documento de Requisitos. Caso ela trate, identifique essas funcionalidades e a partir deste ponto, considere esse Conjunto de Requisitos Funcionais (CRF) para utilizar nos passos seguintes. Marque esses Requisitos Funcionais com o número correspondente a especificação em questão.

...

**Figura 3. Exemplo da Estrutura da Técnica de Leitura ER1.**

Em geral, essas técnicas são compostas de duas fases principais, de acordo com os diagramas que estão sendo comparados, uma explorando mais a questão sintática dos documentos e procurando por conceitos básicos presentes e outra explorando o lado semântico, conduzindo o leitor a procurar pelas discrepâncias levando-se em consideração os conceitos marcados anteriormente.

Com o intuito de fazer uma avaliação preliminar principalmente das técnicas definidas neste trabalho (as apresentadas na Tabela 4), foi conduzido um estudo de caso utilizando-se algumas delas, o qual é brevemente descrito na próxima seção.

#### **4. Síntese do Estudo de Caso conduzido para Avaliar as Técnicas de Leitura**

Para avaliar a dificuldade de compreensão e a viabilidade da utilização das técnicas de leitura no contexto do ProDeS/UML um subconjunto de OORTs/ProDeS, composto de uma leitura para cada fase do ciclo de vida, foi utilizado para a condução de um estudo de caso.

O exemplo selecionado para o estudo de caso foi o SAPES (Sistema de Apoio à Escrita), que tem como principais objetivos o auxílio à pesquisa bibliográfica e a elaboração de documentos científicos, no que diz respeito à geração das referências utilizadas [10]. O documento de requisitos desse sistema passou por um processo de inspeção com o uso das técnicas de leitura PBR (*Perspective Based Reading*) [11,12], sendo corrigido em decorrência desse processo. Seguindo-se o ProDeS/UML, os modelos do SAPES foram construídos por um grupo de professores. Esses modelos não passaram por nenhuma revisão formal e também não continham erros previamente definidos e/ou conhecidos.

O conjunto inicial de participantes deste estudo era composto de vinte e quatro alunos de graduação do curso de Bacharelado em Sistemas de Informação do ICMC-USP. Desse conjunto inicial um subgrupo de 20 alunos chegou ao final e realizou todas as atividades planejadas, as quais ocorreram no contexto de uma das disciplinas cursadas por eles e correspondiam a atividades regulares do curso. A desistência dos alunos foi decorrente de vários motivos, desde ordem pessoal como também de falta de entendimento da UML e do processo utilizado.

Para a aplicação das técnicas OORTs/ProDeS, avaliou-se a restrição de tempo e o objetivo de dar aos alunos uma visão geral das técnicas em todas as fases do ProDeS/UML. Por isso optou-se por aplicar uma técnica para cada fase do processo:

- ER1 (Diagrama de Casos de Uso x Especificação dos Casos de Uso x Documento de Requisitos);
- A2 (Modelo de Operações x Especificação dos Casos de Uso); e
- P1 (Diagrama de Seqüência x Diagrama de Visibilidade x Diagrama de Classes da Análise Refinado).

Essa atividade teve como principal objetivo avaliar a dificuldade e compreensão das técnicas, bem como caracterizar alguns defeitos encontrados. Durante a disciplina foi dado o treinamento no processo ProDeS/UML e na notação UML. Os participantes receberam também um treinamento de uma hora e quarenta minutos, no qual foram apresentados os conceitos necessários para aplicação das técnicas. Durante esse treinamento eles tiveram a oportunidade de acompanhar a aplicação das técnicas em um trecho de um artefato selecionado como exemplo para cada uma das três técnicas escolhidas para o estudo.

Considerando-se as técnicas de leitura ER1 e A2, que foram definidas no âmbito deste trabalho, uma análise preliminar dos resultados obtidos mostra que, do ponto de vista qualitativo, 8 dos 11 participantes que responderam ao questionário de avaliação da atividade, consideraram que as técnicas contribuíram para a identificação de defeitos.

Do ponto de vista quantitativo do total de 20 alunos que aplicaram as técnicas, houve um total de 30 discrepâncias relatadas com relação à leitura ER1 e 53 discrepâncias para a leitura A2. Desse total, uma análise inicial dessa lista de discrepâncias levou à determinação de 7 defeitos para a leitura ER1 e 9 defeitos para a leitura A2, dos quais a maioria são defeitos de omissão e de fato incorreto. Salienta-se que esses tipos de defeitos coincidem com aqueles que vêm sendo relatados nos resultados obtidos pelos estudos experimentais conduzidos com as técnicas OORTs, por Travassos et al. [6]. Assim, ainda que nesse estudo de caso não tenha sido possível aplicar todo o conjunto das técnicas de leitura OORTs/ProDeS, esses resultados preliminares dão indícios de que as técnicas podem auxiliar a identificação de defeitos em artefatos baseados em UML gerados pelo ProDeS/UML.

Observou-se ainda que 7 dos 11 participantes que responderam o questionário consideraram que o tempo de treinamento poderia ser expandido, o que pode ter comprometido a assimilação adequada dos conceitos e evitado uma prática mais efetiva das técnicas. Esse fator deverá constituir motivo de investigação em estudos futuros. Outra dificuldade apontada pelos participantes foi quanto à classificação dos defeitos encontrados, a qual segue uma taxonomia específica e eventualmente uma maior familiaridade com essa classificação teria facilitado e tornado mais eficaz o uso das técnicas de leitura.

## 5. Conclusões e Trabalhos Futuros

Apresentou-se neste trabalho um conjunto de técnicas de leitura, denominado OORTs/ProDeS, para inspeção de artefatos de software no contexto de um processo de desenvolvimento OO denominado ProDeS/UML. Um subconjunto dessas técnicas, denominado OORTs, foi previamente definido por Travassos et al. [6] e, no contexto desse processo, podem ser utilizadas para inspecionar artefatos da fase de Projeto.

As OORTs estão baseadas em um processo genérico e simplificado e como a própria notação UML não propõe um processo para o desenvolvimento do software, mostrou-se, tomando-se por base a fase de Projeto do ProDeS/UML, que as OORTs não eram plenamente adequadas aos artefatos gerados nessa fase desse processo específico, muito embora a notação usada seja a mesma das OORTs (UML).

Assim, evidenciou-se a importância da definição e utilização de uma estratégia de inspeção que fosse baseada em um processo de desenvolvimento, o qual tenha como objetivo garantir uma uniformidade de uso, definição e entendimento da notação utilizada em um ambiente de desenvolvimento de software. A definição de um processo facilita a determinação e implantação de atividades de garantia de qualidade de software, como o uso das técnicas de leitura aqui propostas.

Apresentou-se uma análise da viabilidade de aplicação das técnicas OORTs nos artefatos da fase de Projeto do ProDeS/UML e as adaptações necessárias para que essa aplicação pudesse ser efetivada. A mesma análise foi realizada para as demais fases desse processo, chegando-se à definição de novas técnicas de leitura para os documentos gerados nessas outras fases, as quais foram inspiradas nas leituras já existentes.

Finalmente, mostrou-se que, através de um estudo de caso preliminar que teve como um de seus objetivos fazer uma primeira avaliação das técnicas aqui propostas, os resultados dão indícios da contribuição desse novo conjunto na identificação de defeitos em artefatos não cobertos pelas leituras definidas anteriormente. Aparentemente, os resultados tendem a confirmar aqueles que já vêm sendo obtidos pelos diversos experimentos conduzidos no âmbito das técnicas OORTs, ou seja, essas técnicas ajudam o revisor a direcionar a leitura dos artefatos gerados, aumentando a efetividade na identificação de defeitos.

Observa-se que os princípios e conceitos utilizados neste trabalho para a definição das OORTs/ProDeS podem ser facilmente reutilizados no contexto de outros processos de desenvolvimento de software OO.

Com base no estudo realizado, as técnicas OORTs/ProDeS estão sendo revisadas e um novo estudo para avaliação do conjunto completo das técnicas está sendo planejado para o 2º semestre 2002. Dessa forma, contribui-se para que instituições que adotem o paradigma OO para o desenvolvimento de software tenham uma base para o estabelecimento de técnicas de leitura/inspeção efetivas e satisfaçam uma das atividades para atingir o nível 3 do CMM.

## 6. Referências Bibliográficas

- [1] RATIONAL. **Documentação Oficial da UML**. Versão 1.3, junho, 808 p, 1999. Disponível em: <http://www.rational.com/uml/resources/documentation>. Acesso em 10/10/2000.
- [2] PRESSMAN, R. S. **Software Engineering - A Practitioner's Approach**, 5th ed., Mc Graw Hill, 2001.
- [3] PAULK, M. C.; WEBER, C. V.; CURTISS, B.; CHRISIS, M. B. **The Capability Maturity Model: Guidelines for Improving the Software Process**. CMU/SEI, Addison-Wesley, 1995.
- [4] NBR ISO-IEC 12207:1999, **Tecnologia da Informação - Processos de Ciclo de Vida de Software** (corresponde à ISO/IEC 12207: 1995).
- [5] ANDRIOLE, S. J. **Software Validation, Verification, Testing and Documentation**. New Jersey: Petrocelli Books, 1986.
- [6] TRAVASSOS, G. H.; SHULL, F.; CARVER, J; BASILI, V. R. **Reading Techniques for OO Design Inspections**, 2002, 56 p. Technical Report CS-TR-4353, UMIACS-TR-2002-33, University of Maryland, Maryland. Disponível em: <http://www.cs.umd.edu/Library/TRs/>. Acesso em 04/04/2002.
- [7] COLANZI, T. E. **Uma Abordagem Integrada de Desenvolvimento e Teste de Software Baseada na UML**, 1999, 143 p. Dissertação (Mestrado em Ciência da Computação) - Instituto de Ciências Matemáticas e Computação, USP, São Carlos.
- [8] COLEMAN, D. et al. **Object-Oriented Development: The Fusion Method**. New Jersey: Prentice Hall International, Englewood Cliffs, 1994.
- [9] RATIONAL ROSE v2001. Disponível em: <http://www.rational.com/products/rose>. Acesso em 18/04/2002.
- [10] TURINE, M. A. S.; MASIERO, P. C. **Especificação de Requisitos: Uma Introdução**, 1996, 25 p. Relatório Técnico - Instituto de Ciências Matemáticas e Computação, USP, São Carlos.

- [11] BASILI, V. R.; GREEN S.; LAITENBERGER, O.; LANUBILE, F.; SHULL, F.; SORUMGARD, S.; ZELKOWITZ, M. The Empirical Investigation of Perspective-Based Reading. **Empirical Software Engineering: An International Journal**, v.1, n.2, p. 133-164, 1996b.
- [12] SHULL, F.; RUS, I.; BASILI, V. R. How Perspective-Based Reading can Improve Requiriments Inspections. **IEEE Computer**, v. 33, n.7, p. 73-79, 2000.

**Agradecimento:** os autores agradecem a colaboração de Elisa Yumi Nakagawa, Erika Nina Höhn e Luciana Andreia Fondazzi Martimiano na elaboração e execução do estudo de caso.