

Técnicas de Leitura de Software: Uma Revisão Sistemática

Sômulo Nogueira Mafra, Guilherme Horta Travassos

COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação, Cx. Postal 68.511,
CEP 21945-970, Rio de Janeiro - RJ - Brasil

{somulo, ght}@cos.ufrj.br

Abstract. *The object-oriented paradigm (OO) has smoothly succeeded in the industrial development environments. However, the accomplishment of verification and validation activities concerned with OO system still remains a challenge. We believe that the use of reading techniques for software inspections can represent a feasible alternative into this context. In this paper the results from a systematic review regarding reading techniques applied to OO software have been described, including the discussion about research challenges concerned with such techniques.*

Resumo. *A orientação a objetos (OO) alcançou considerável sucesso no desenvolvimento industrial. Entretanto, a condução de atividades de verificação e validação em software OO ainda é um desafio. Acreditamos que a utilização de técnicas de leitura em inspeções de software seja uma alternativa viável para garantir a qualidade do software OO. Em vista disso, o presente artigo descreve os resultados de uma revisão sistemática conduzida com o objetivo de identificar, analisar e avaliar técnicas de leitura aplicáveis na garantia de qualidade no desenvolvimento de sistemas OO, cujos resultados apontam para novos desafios de pesquisa nesta área.*

1. Introdução

A busca por qualidade no desenvolvimento de sistemas tem sido uma tarefa árdua para a indústria de software. Em geral, gerentes são constantemente desafiados com o seguinte dilema: como desenvolver software de qualidade, a baixo custo, com restrições de recursos e cronograma cada vez mais irreais, de forma a atender satisfatoriamente às necessidades do cliente?

Para agravar a situação, o advento de novas tecnologias parece não estar contribuindo de forma satisfatória para a resolução desse problema. O aclamado benefício do paradigma orientado a objetos (OO), que supostamente diminui o custo de manutenção evolutiva, não foi observado em [Hatton 1998]. Além disso, algumas características intrínsecas do paradigma OO como polimorfismo, *encapsulamento*, generalização, entre outros, dificultam a predição do caminho de execução do software OO, tornando a aplicação de testes estruturais um desafio [Lima e Travassos 2004].

Nesse sentido, a condução sistemática de inspeções tem se mostrado uma alternativa viável para contornar essa situação, ao permitir que a qualidade do software OO seja avaliada ainda em suas fases iniciais de desenvolvimento. Proposta inicialmente por Fagan [Fagan 1976], a inspeção de software caracteriza-se por ser um tipo particular de revisão que pode ser aplicada a todos os artefatos de software, possuindo um processo de detecção de defeitos rigoroso e bem definido. A importância da condução de inspeções tem sido reforçada explicitamente por modelos de maturidade

como o MR mps [Weber *et al.* 2004] e o CMMI [Chrissis *et al.* 2003].

Porém, embora o processo de inspeção de software tenha evoluído consideravelmente no âmbito gerencial, até o início dos anos 90s pouca ênfase tinha sido dada para o trabalho individual do principal responsável pela detecção de defeitos: o revisor de software [Shull 1998]. Em geral, as abordagens de inspeção propostas até então utilizavam uma abordagem de leitura *ad hoc*, não fornecendo orientação sobre como o artefato deveria ser examinado em busca de defeitos; elas simplesmente assumiam que todos os revisores inspecionavam o artefato efetivamente [Knight e Myers 1993]. Uma provável consequência dessa falta de orientação efetiva pôde ser constatada na experiência vivenciada pela introdução de inspeções no Laboratório de Engenharia de Software do GSFC/NASA no final da década de 80 [Ciolkowski *et al.* 2002]. Naquele momento, observou-se que a condução sistemática de inspeções não contribuiu de forma satisfatória para o aumento da taxa de detecção de defeitos.

Diante desses resultados, pesquisadores começaram a se preocupar com a definição de técnicas que apoiassem de maneira sistemática as tarefas individuais de leitura de um artefato. Os resultados promissores obtidos com a introdução da técnica *cleanroom* [Mills 1991], no contexto de inspeções de código-fonte, motivaram a pesquisa por técnicas de leitura que pudessem ser aplicadas a outros tipos de artefatos, como documentos de requisitos e modelos de projeto [Ciolkowski *et al.* 2002].

Desde então, a pesquisa envolvendo técnicas de leitura tem sido direcionada por resultados de estudos experimentais executados ao longo dos últimos anos. À medida que novos estudos são executados, as técnicas são refinadas e novas hipóteses geradas, proporcionando assim um maior corpo de conhecimento.

Entretanto, não há registros na literatura de revisões formais visando à identificação dos diversos estudos experimentais existentes sobre técnicas de leitura. Algumas revisões existentes [Regnell *et al.* 2000] [Thelin *et al.* 2003] [Leite *et al.* 2005] não apresentam protocolos de pesquisa ressaltando o escopo da revisão nem os critérios de avaliação utilizados, resumindo-se, assim, a um agrupamento de estudos com pouca representatividade do universo dos estudos. Como consequência, muitas dessas revisões possuem pouca abrangência. Além disso, por não apresentarem um protocolo de revisão estabelecido, não são passíveis de repetição, o que as torna dependentes dos revisores que a executaram, aumentando possivelmente o viés de seus resultados.

Em razão disso, conduzimos uma revisão sistemática [Biolchini *et al.* 2005] com o objetivo de identificar, analisar e avaliar as técnicas de leitura propostas na literatura. Para cada técnica identificada, descrevemos seus conceitos e resultados obtidos experimentalmente. Esperamos dessa forma contribuir para o aprimoramento das atividades de garantia da qualidade de software ao relatar o que supostamente funciona e o que não funciona na aplicação dessas técnicas de leitura. O texto encontra-se dividido em sete seções incluindo esta que introduz o artigo. A seção 2 descreve os principais conceitos envolvendo técnicas de leitura. A seção 3 relata a revisão sistemática conduzida apresentando detalhadamente o protocolo utilizado. A seção 4 apresenta os resultados obtidos da revisão, descrevendo as técnicas identificadas e os resultados obtidos experimentalmente. A seção 5 discute esses resultados, a seção 6 contextualiza a aplicação de técnicas de leitura na garantia da qualidade de sistemas OO e, finalmente, a seção 7 conclui o artigo.

2. Técnicas de Leitura para Artefatos de Software

As técnicas de leitura surgiram como forma de melhorar o desempenho da inspeção de software, no que se refere à atividade de detecção de defeitos. O termo “leitura” foi escolhido de forma a enfatizar as similaridades com o processo mental que as pessoas utilizam quando tentam entender o significado de algum texto [Shull 1998]. Nas subseções seguintes apresentamos a definição de técnica de leitura, além de uma comparação com a técnica de inspeção de software mais utilizada na indústria.

2.1. Definição

Uma técnica de leitura pode ser caracterizada como uma série de passos para a análise individual de um artefato de software de forma a permitir a extração do entendimento necessário para a execução de uma determinada tarefa [Basili 1996]. Shull [Shull 1998] observou três importantes critérios nessa definição: (i) *uma série de passos*: a técnica deve prover ao desenvolvedor orientação na condução da leitura. Essa orientação pode variar desde um procedimento passo a passo a um conjunto de questões formuladas com o objetivo de manter o foco da leitura; (ii) *análise individual*: técnicas de leitura devem se preocupar com o processo de compreensão individual. Ainda que alguns métodos de garantia da qualidade, nos quais a técnica esteja inserida, devam requerer o trabalho em equipe (como numa reunião de inspeção), a compreensão de certos aspectos do artefato ainda assim é uma tarefa individual; (iii) *entendimento necessário para a execução de uma tarefa*: a técnica deve apoiar a obtenção de um certo nível de entendimento de alguns (todos) aspectos de um artefato.

Basili [Basili 1997] aponta que as técnicas de leitura necessitam serem dependentes de contexto, bem definidas e orientadas a objetivos. Baseados nisso, estabeleceram os requisitos para uma técnica de leitura: (i) estar associada a um tipo de artefato (ex.: documento de requisitos) e a notação pela qual o artefato é descrito (ex.: língua portuguesa); (ii) ser adaptável de acordo com as características intrínsecas da organização e do desenvolvimento; (iii) ser detalhada, provendo ao desenvolvedor um processo bem definido. A existência de um processo bem definido permitiria que a técnica fosse executada em outros projetos, atuando como um meio de disseminação de conhecimento; (iv) ser avaliada experimentalmente para determinar sua viabilidade e seu grau de efetividade na detecção de defeitos.

2.2. Técnicas de Inspeção de Software

Segundo um estudo apresentado em [Laitenberger 1998], a abordagem de leitura utilizando *checklist* é a técnica mais empregada para a detecção de defeitos no contexto de inspeções de software, entre as organizações que empregam algum apoio à leitura. A abordagem de *checklist* tenta aumentar a efetividade da inspeção e diminuir a relação de custo por defeito ao focar a atenção do revisor num conjunto pré-definido de questões. Os revisores respondem a essas questões, atribuindo respostas no formato de “sim/não”, enquanto lêem um artefato de software.

A seguir, descrevemos algumas das deficiências na abordagem de *checklist* apontando como as técnicas de leitura supostamente poderiam resolvê-las.

Orientação de leitura - Na abordagem de *checklist*, freqüentemente faltam instruções concretas sobre como utilizar o *checklist*. Isto é, geralmente o revisor não dispõe de orientação sobre quando, e baseado em quais informações, ele deve responder a uma

determinada questão. O uso de técnicas de leitura, por sua vez, apresenta um procedimento bem definido que orienta o revisor a localizar *qual* informação é relevante ao propósito da revisão. Uma vez localizada a informação, o revisor é orientado a *como* proceder para detectar defeitos.

Escopo de leitura - A abordagem de *checklist* geralmente requer que cada revisor verifique todas as informações no artefato sob inspeção levando-o a lidar com uma quantidade considerável de informação pouco relevante. Técnicas de leitura contornariam esse problema, ao focar a atenção do revisor nos aspectos do artefato relevantes ao propósito da revisão.

Formulação das questões - As questões presentes em um *checklist* são freqüentemente baseadas em dados históricos de defeitos detectados na organização [Chernak 1996]. Caso tais dados não estejam disponíveis, as questões do *checklist* são geralmente extraídas da literatura. Em ambos os casos, revisores podem não prestar atenção em classes de defeitos ainda não detectadas em inspeções anteriores. Por sua vez, as questões providas pelas técnicas de leitura são provenientes de um acúmulo de conhecimento adquirido por anos de experimentos executados na indústria e academia.

Documentação - *Checklist*, ao contrário de técnicas de leitura, não requer que o revisor documente sua inspeção. Assim, o resultado da inspeção não é passível de repetição, sendo dependente do revisor que executou a inspeção, restringindo a disseminação de conhecimento e diminuindo a oportunidade de melhoria de processos.

Aquisição de Conhecimento sobre o Problema - A utilização de técnicas de leitura requer que o revisor trabalhe ativamente com o artefato sob inspeção possibilitando o ganho de conhecimento sobre o problema descrito. Esta característica não é observada em *checklists*, visto que o revisor é orientado apenas a responder questões.

Foco em Defeitos - Ao proporcionar o ganho de conhecimento sobre o problema, técnicas de leitura levariam à descoberta de defeitos críticos. *Checklist*, por não possuir tal característica, limitaria a detecção de defeitos de tal severidade.

Direcionamento para Testes - A condução de testes é uma atividade custosa para o desenvolvimento de software [Lima e Travassos 2004]. O uso de técnicas de leitura, ao fornecer uma categorização de defeitos e facilitar a descoberta de defeitos críticos permite direcionar os esforços de testes, o que não ocorre com o uso de *checklist*.

Em vista disso, a aplicação de técnicas de leitura aparentemente traz mais benefícios para a condução de inspeções do que a abordagem com *checklist*. Entretanto, as comparações acima devem ser lidas com cautela pois, como veremos na seção 5, nem todos os supostos benefícios das técnicas de leitura ainda foram observados experimentalmente. Na seção seguinte, detalhamos a revisão sistemática conduzida.

3. Aplicando uma Revisão Sistemática: Técnicas de Leitura de Software

Uma revisão sistemática é um meio de identificar, avaliar e interpretar toda pesquisa disponível e relevante sobre uma questão de pesquisa, um tópico ou um fenômeno de interesse [Kitchenham 2004]. A condução de uma revisão sistemática supostamente apresenta uma avaliação justa do tópico de pesquisa à medida que utiliza uma metodologia de revisão rigorosa, confiável e passível de auditoria. Além disso, uma revisão sistemática deve obrigatoriamente conter o protocolo de busca executado de forma a permitir que a revisão seja repetida por outros pesquisadores interessados.

Os estudos individuais que são analisados em uma revisão sistemática são chamados de estudos primários. A revisão sistemática em si é considerada um tipo de

estudo secundário. Em particular, pesquisadores conduzindo revisões sistemáticas devem despende esforços na identificação e relato de pesquisas que apóiam e não apóiam suas hipóteses. Se os estudos identificados apresentam resultados consistentes, a revisão sistemática provê indícios de que o fenômeno é robusto e generalizável a outros contextos. Caso os resultados dos estudos sejam inconsistentes, as fontes de variação desses resultados podem ser estudadas. Maiores informações sobre revisões sistemáticas podem ser encontradas em [Biolchini *et al.* 2005] e [Kitchenham 2004].

3.1. Material Utilizado

Para possibilitar a documentação da revisão sistemática, facilitando a extração e sumarização dos dados, foram elaborados os seguintes formulários:

- Formulário de Condução da Revisão: formulário contendo campos para a armazenagem de informações sobre a fonte onde a busca foi realizada, a data de realização da busca, as combinações de palavras-chave que proporcionaram a busca dos artigos e a lista dos artigos encontrados.
- Formulário de Seleção de Estudos: formulário contendo campos informando: nome do artigo, lista de autores, data de sua publicação, veículo de publicação do artigo, fonte de onde foi obtido e situação do artigo (pendente, incluído e excluído).
- Formulário de Extração de Dados: formulário contendo campos para o resumo do artigo, objetivo do estudo, descrição do estudo experimental, resultados do estudo, além de comentários adicionais do pesquisador que extraiu os dados, entre outros.

3.2. Protocolo da Revisão

O protocolo de revisão sistemática abaixo descrito foi baseado nos modelos disponíveis em [Biolchini *et al.* 2005] e [Mendes e Kitchenham 2005].

Objetivo: Apesar do foco inicial da pesquisa estar na identificação de técnicas de leitura para documentos de requisitos que apóiem inspeções de software OO, decidimos não restringir a busca apenas a esse tipo de técnicas. A justificativa foi a possibilidade de obtermos as lições aprendidas e os resultados decorrentes de estudos experimentais envolvendo todas as técnicas de leitura.

Porém, não estávamos interessados apenas na identificação desses estudos. Necessitávamos avaliar quão efetiva e eficiente¹ foi a utilização da técnica no contexto dos estudos realizados. Acreditávamos que as respostas a essas questões constituiriam um passo fundamental para a identificação de quaisquer necessidades de pesquisa envolvendo técnicas de leitura. Caso as respostas fossem positivas, pretendíamos executar novos estudos com o objetivo de identificar o grau de evidência experimental obtido, visando a relatar os benefícios e as limitações da tecnologia. Caso fossem negativas, pretendíamos executar novos estudos para tentar responder nossas perguntas.

Dessa forma, conduzimos uma revisão sistemática cujo *objetivo* foi identificar, analisar e avaliar estudos experimentais envolvendo técnicas de leitura de documentos de requisitos descritos em linguagem natural *com o propósito* de caracterizá-las com respeito à usabilidade, eficiência na identificação de defeitos e efetividade na cobertura dos defeitos identificados *do ponto de vista* dos participantes dos estudos experimentais *no contexto* no qual os estudos foram executados.

¹ Definimos efetividade como a quantidade de defeitos encontrados e eficiência como a quantidade de defeitos encontrados por unidade de tempo, por exemplo, hora de inspeção.

Formulação da pergunta: Quais são as técnicas de leitura de documentos de requisitos descritos em linguagem natural?

Intervenção: Técnicas de leitura de documentos de requisitos de software.

Controle: *Abordagem de checklists* e leitura *ad hoc*.

População: Processos de inspeção de documentos de requisitos de software.

Resultados: Técnicas de leitura de documentos de requisitos de software.

Aplicação: Projetos de desenvolvimento utilizando inspeções de requisitos.

Crítérios de seleção de fontes: disponibilidade de consulta de artigos através da *web*; presença de mecanismos de busca através de palavras-chave; garantia de resultados únicos através da busca de um mesmo conjunto de palavras-chave; possuir nível A de avaliação Qualis Capes [Qualis 2004].

Métodos de busca de fontes: as fontes serão acessadas via *web*, portanto, no contexto desta revisão, não será considerada a busca manual.

Palavras-chave: requirement specification, requirement document, user requirement. reading technique; review, inspection, fault detection, defect detection; experimental software engineering, empirical software engineering.

Listagem de fontes: periódicos da CAPES (IEEE journals and conferences, IEE journals, IEE conferences, IEEE conferences, ACM journals, ACM conferences, Kluwer journals e Elsevier journals).

Tipo dos artigos: artigos contemplando a execução de estudos experimentais.

Idioma dos artigos: Inglês. A escolha do inglês como língua oficial deve-se à universalidade do idioma. A não inclusão do português justifica-se pela necessidade dos estudos deverem ser passíveis de repetição em diferentes contextos presentes em diferentes países. Um estudo em português claramente limitaria essa possibilidade.

Crítérios de inclusão e exclusão dos artigos

Os artigos devem estar disponíveis na *web*;

Os artigos devem apresentar textos completos dos estudos em formato eletrônico;

Os artigos devem estar descritos em inglês;

Os artigos devem contemplar técnicas de inspeção de documentos de requisitos descritos em linguagem natural;

Os artigos devem contemplar a execução de estudos experimentais investigando técnicas de inspeção de documentos de requisitos;

A escala envolvida nos cinco critérios é nominal envolvendo duas opções: *sim* ou *não*.

Crítérios de qualidade dos estudos primários: a avaliação da qualidade dos estudos será baseada nos critérios para avaliação de estudos experimentais em engenharia de software presentes em [Kitchenham *et al.* 2002].

Processo de seleção dos estudos primários

O processo de seleção dos estudos primários consistirá dos seguintes passos:

1. Um pesquisador executa o processo de busca em cada uma das fontes selecionadas para identificar os artigos que contenham estudos experimentais;
2. Os artigos encontrados são obtidos da fonte e documentados na lista de artigos encontrados, presente no Formulário de Condução da Revisão;
3. Os artigos encontrados pelo processo de busca são selecionados pelos demais pesquisadores através da verificação dos critérios de inclusão e exclusão estabelecidos; a verificação é executada mediante a leitura do *abstract* do artigo.

4. Os artigos incluídos e excluídos são documentados na lista de artigos incluídos e lista de artigos excluídos, respectivamente, presentes no Formulário de Condução da Revisão, juntamente com a justificativa de sua inclusão ou exclusão.
5. Os artigos incluídos são avaliados mediante a leitura do artigo inteiro; os artigos incluídos são documentados no Formulário de Seleção de Estudos e encaminhados para a avaliação da qualidade dos estudos primários. Os artigos excluídos são documentados na lista de artigos excluídos junto com a justificativa de exclusão.

Havendo divergência, os pesquisadores entram em consenso sobre a seleção dos artigos. **Avaliação da qualidade dos estudos primários:** os estudos selecionados mediante a execução do processo de seleção dos estudos primários serão avaliados pelos pesquisadores de acordo com os critérios de qualidade estabelecidos. O resultado da avaliação da qualidade de cada estudo irá determinar sua inclusão ou exclusão da lista dos estudos de onde serão extraídos os dados.

Estratégia de extração de informação: para cada estudo selecionado durante a execução do processo de avaliação da qualidade dos estudos primários será utilizado uma cópia do Formulário de Extração de Dados.

Sumarização dos resultados: os resultados serão tabulados. Nenhuma meta-análise será realizada.

O protocolo de revisão foi avaliado por um especialista gerando duas versões até sua aprovação final.

3.3. Condução da Revisão

A revisão sistemática foi conduzida entre o final de 2004 e início de 2005, durante um período de três meses. Ao todo foram analisados 278 artigos dos quais 38 foram selecionados e tiveram os seus dados extraídos e analisados, de acordo com as recomendações previstas no protocolo de revisão.

Algumas fontes reconhecidamente importantes no contexto da pesquisa não foram contempladas pelo protocolo de revisão, haja vista que não atendiam aos critérios de seleção de fontes. Entre elas destacamos o ISERN² que não possui máquina de busca. Entretanto, decidimos incluir os estudos presentes nessas fontes que se encaixavam ao contexto da revisão, para não prejudicar a abrangência da revisão. Além disso, alguns estudos que não estavam disponíveis na *web* foram obtidos mediante contato com seus autores. Porém, as formas pelas quais esses estudos foram obtidos foram devidamente documentadas no formulário de condução da revisão.

A revisão sistemática conduzida, desde sua concepção até a extração dos resultados encontra-se documentada [Mafra e Travassos 2005], de forma a permitir sua repetição e auditoria por pesquisadores interessados no estudo.

3.4. Lições Aprendidas da Condução da Revisão Sistemática

Como algumas das lições aprendidas decorrentes da condução da revisão sistemática, observamos que a definição de um protocolo de revisão possibilitou o planejamento do cronograma para a condução da revisão, permitindo o estabelecimento de marcos e pontos de controle. Em vista disso, a revisão sistemática pôde ser monitorada. Além disso, a elaboração do protocolo de revisão obrigou o estabelecimento de um escopo

²International Software Engineering Research Network. In: www.iese.fhg.de/network/ISERN/pub/isem_biblio_tech.html

bem definido, representado pelo objetivo da revisão e pelas questões de pesquisa. Dessa forma, fomos orientados a pensar primeiro no escopo da pesquisa antes de começarmos a busca por estudos primários de forma precipitada. Essa manutenção do foco de pesquisa, minimizou a possibilidade do desperdício de esforço na busca e leitura de estudos irrelevantes para o contexto da pesquisa, fato comum em pesquisadores pouco experientes. A elaboração dos formulários para a extração dos resultados, por sua vez, facilitou a comunicação entre os pesquisadores sobre os principais aspectos de cada estudo investigado. Na próxima seção, apresentamos os principais resultados obtidos.

4. Resultados Obtidos

A seguir, descrevemos sucintamente as técnicas de leitura encontradas pela revisão, destacando os principais resultados obtidos através da execução de estudos experimentais. As referências dos estudos encontram-se em [Mafra e Travassos 2005].

DBR (Defect-Based Reading) - DBR [Porter e Votta 1994] representa uma família de técnicas de leitura para a detecção de defeitos em documentos de requisitos descritos em SCR [Heninger 1980], uma notação formal baseada em máquinas de estado. Cada técnica DBR pode ser projetada para focar uma determinada classe de defeitos específicos à notação SCR (omissão de funcionalidade, inconsistência de tipos etc). No contexto da revisão conduzida não estávamos interessados em DBR, haja vista que o protocolo de revisão restringia a pesquisa a requisitos escritos em linguagem natural.

PBR (Perspective-Based Reading) - A leitura baseada em perspectiva (PBR) [Shull 1998] representa uma família de técnicas de leitura elaborada inicialmente para a detecção de defeitos em documentos de requisitos escritos em linguagem natural. Entretanto, sua utilização foi estendida para inspeções de projeto e código-fonte. No contexto da revisão não consideramos a aplicação de PBR em inspeção de código-fonte.

Descrição - PBR auxilia a detecção de defeitos ao observar quais informações nos requisitos possuem maior relevância para as diferentes formas de utilização do documento. Por exemplo, em um modelo de ciclo de vida de software, podemos esperar que o documento de requisitos tenha três usos principais: (a) como base para a construção do modelo de projeto, (b) como base para a elaboração do plano de teste do sistema e (c) como descrição das funcionalidades do sistema, como mostra a Figura 1.



Figura 1 - uso do documento de requisitos ao longo do ciclo de vida do software
Adaptado de [Shull 1998]

Nesse sentido, PBR assegura que cada revisor avaliará o documento segundo uma dessas perspectivas ao fornecer um procedimento orientando-o a criar um modelo físico com base nos requisitos. A construção do modelo ajudaria o revisor a focar sua revisão. Dessa forma, o revisor que assumisse a perspectiva de testador construiria um conjunto de casos de testes para um potencial plano de teste, o revisor responsável pela perspectiva de projetista construiria um modelo de projeto e o revisor responsável pela perspectiva de usuário construiria um manual contendo as funcionalidades envolvidas no sistema. O objetivo não é duplicar o trabalho realizado em outros pontos do ciclo de

desenvolvimento do software, mas criar representações que possam ser usadas como base para a criação futura de artefatos mais específicos e que possam revelar quão bem os requisitos conseguem apoiar as tarefas seguintes.

Durante a criação do modelo físico, os revisores tentariam identificar os possíveis defeitos existentes. Para facilitar esse trabalho, a PBR fornece um conjunto de questões ajustadas para cada passo do procedimento de criação do modelo, que são respondidas à medida que o revisor percorre as etapas de sua construção. Quando os requisitos não fornecem informações suficientes para responder às questões, isso significa que também não fornecem informações suficientes para apoiar o usuário dos requisitos em suas tarefas, caracterizando a presença de um potencial defeito.

Resultados Experimentais - No contexto da revisão conduzida, encontramos 19 estudos experimentais envolvendo PBR (Tabela 1).

Tabela 1 - Série de experimentos envolvendo PBR

Estudo (Instituição – Ano)	Ambiente	Questão de Pesquisa
Nasa-1996	Industrial	PBR vs. Técnica usual da NASA.
Wales-1996	Acadêmico	PBR (FPS) vs. <i>ad hoc</i> .
UFSCar-1997	Acadêmico	PBR vs. <i>checklist</i> vs. <i>ad hoc</i> .
Kaiserslautern-1997	Acadêmico	PBR vs. <i>ad hoc</i> .
Maryland-1998	Acadêmico	PBR vs. <i>ad hoc</i> .
Lund-1998	Acadêmico	Diferenças entre perspectivas de PBR.
Kaiserslautern-1999	Acadêmico	PBR vs. <i>checklist</i> (aplicados a modelo OO).
Bari-2000	Acadêmico	PBR vs. <i>checklist</i> vs. <i>ad hoc</i> .
Viena-2000	Acadêmico	PBR vs. <i>checklist</i> .
Viena-2001	Acadêmico	PBR vs. <i>checklist</i> .
Osaka-2001	Acadêmico	PBR vs. <i>checklist</i> (aplicados a modelo OO).
Ericsson-2003	Industrial	PBR(<i>testador</i>) vs. <i>checklist</i> .
Kaiserslautern-2004	Acadêmico	Orientação ativa de PBR.
Bari-2004	Acadêmico	Orientação ativa de PBR.
Ericsson-2004	Industrial	PBR (<i>testador</i>) vs. <i>checklist</i> .
COPPE-2003a	Acadêmico	Viabilidade de apoio ferramental.
COPPE-2003b	Acadêmico	Viabilidade de apoio ferramental.
UFSCar-2004	Acadêmico	PBR vs. <i>checklist</i> .
USP-2004	Acadêmico	PBR vs. <i>checklist</i> .

Dos 14 estudos avaliando a efetividade de PBR, seja comparando a técnica com a abordagem de *checklist* ou com leitura *ad hoc*, 57% apresentaram resultados confirmando a superioridade de PBR, 29% falharam em demonstrar tal superioridade, enquanto 14% não apresentaram diferenças significativas.

Resultados de Bari-2004 apontaram que a discrepância entre os resultados obtidos pelos estudos pode ser explicada pela observação do baixo nível de conformidade com o processo proposto por PBR. Além disso, em Bari-2000, apenas 20% dos participantes admitiram ter seguido efetivamente PBR. Tal efeito também foi observado em NASA-1996. Nesse estudo, revisores tenderam a abandonar PBR, preferindo utilizar seu próprio processo tácito de leitura, quando inspecionaram um documento com domínio de problema familiar. Para minimizar esse efeito, Silva e Travassos [Silva e Travassos 2004] propuseram um apoio ferramental para a aplicação de PBR, permitindo a coleta automática de medidas para a avaliação do nível de conformidade com o processo de PBR. Os estudos de COPPE-2003a e COPPE-2003b

avaliaram de forma positiva a viabilidade de utilização desse apoio ferramental.

Uma das principais características de PBR, a sua adaptabilidade a diferentes contextos, foi estudada em Wales-1996, onde foram utilizados cenários para sistemas de informação, utilizando a técnica de análise por pontos de função (FPS).

Resultados qualitativos de Kaiserslautern-2004 apontaram a orientação ativa de PBR como o principal fator para o aumento na quantidade de defeitos detectados. No estudo de Ericsson-2003, participantes afirmaram que a criação de casos de testes, como resultado da aplicação de PBR (perspectiva de testador), aumentou o conhecimento sobre as expectativas do cliente e conseqüentemente a qualidade da especificação de requisitos final. Em Ericsson-2004, participantes mostraram-se surpresos com o número relativamente alto de defeitos encontrados por PBR, uma vez que eles se dedicaram ao máximo na detecção de defeitos, utilizando *checklist*, em uma inspeção realizada anteriormente sobre o mesmo documento de requisitos. Entretanto, no estudo de Bari-2004 não foram observados os supostos benefícios da orientação ativa de PBR.

O ganho de conhecimento sobre o problema, supostamente decorrente da orientação ativa de PBR, pode ser o responsável pelo aumento na detecção de defeitos críticos para o sistema, como observado nos estudos de Viena-2000 e Ericsson-2004. Os resultados obtidos em Wales-1996 chamam a atenção sobre a quantidade de defeitos sintáticos (muitas vezes pouco relevantes), como a localização incorreta de informações, detectados pela leitura *ad hoc*. Tal efeito pode ser explicado pela falta de orientação provida aos revisores. Porém, em Kaiserslautern-1997 não foi possível encontrar diferenças significativas na detecção de defeitos críticos. Uma outra provável conseqüência desse aumento do grau de conhecimento sobre o problema, como observado no estudo de Bari-2004, é o melhor desempenho de PBR comparado com *checklist*, durante a inspeção de documentos maiores e mais complexos.

A diminuição na sobreposição dos defeitos detectados pelos revisores, devido à utilização de diferentes perspectivas que focam em diferentes aspectos dos requisitos, foi observada de forma satisfatória nos estudos de Viena-2000, Viena-2004 e Wales-1996. Entretanto, os autores do estudo de Wales-1996 levantaram uma importante hipótese: não é a utilização de diferentes cenários que aumenta a taxa de detecção de defeitos, mas sim a utilização de uma estratégia bem definida para a decomposição do problema (*separation of concerns*). Em Lund-1998 não foi observado tal efeito.

Os estudos de Viena-2000 e Viena-2001 apontaram uma correlação positiva entre o nível de experiência dos revisores com o aumento do desempenho na detecção de defeitos. Entretanto, os resultados de Nasa-1996, Maryland-1998 e Wales-1996 não apontaram diferenças significativas entre os desempenhos de participantes mais experientes e menos experientes aplicando PBR. Além disso, os resultados de Viena-2001 mostraram que a leitura baseada em *checklist* é mais dependente da qualificação dos revisores do que PBR. Situação similar ocorreu em Wales-1996, onde participantes que tiveram maior desempenho aplicando leitura *ad hoc* foram os mais experientes.

No estudo de Viena-2000 foi observado que a efetividade dos revisores aumenta com o esforço de inspeção até um certo limite, quando começa a decair. Para revisores aplicando a abordagem de *checklist*, esse pico foi entre 4 e 6 horas. Para revisores aplicando PBR, o pico ficou entre 6 e 8 horas. Revisores com esforço abaixo ou acima

desse intervalo tiveram menor desempenho.

OORT's (Object-Oriented Reading Techniques) – OORT's [Travassos *et al.* 1999] representa uma família de sete técnicas de leitura que fornecem um procedimento para as revisões individuais dos diferentes diagramas e documentos de projeto OO.

Descrição - O processo de leitura utilizando OORT's precisa ser em duas dimensões: leitura horizontal e vertical. Na leitura horizontal, diferentes diagramas de projeto são verificados para assegurar que estejam consistentes entre si. Na leitura vertical, é necessária a validação dos modelos de projeto, para assegurar que o projeto do sistema está correto com relação aos requisitos.

Como exemplo de aplicação da leitura horizontal OORT's, citamos uma situação onde diagramas de seqüência necessitassem ser comparados a máquinas de estado, identificando-se para um determinado objeto os eventos, as restrições ou os dados que poderiam mudar a forma como as mensagens são enviadas para ele. Uma vantagem da família de técnicas OORT's é que poderíamos selecionar apenas o subconjunto de técnicas que correspondessem aos artefatos que devam ser revisados ou que são especialmente importantes em um determinado momento. O conjunto completo de técnicas é ilustrado na Figura 2. Cada linha entre os artefatos representa uma técnica de leitura para ler um documento comparando-o com outro.

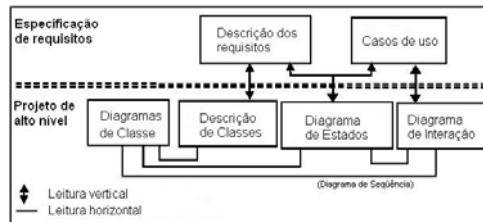


Figura 2 – OORT's — uma família de técnicas para leitura de projeto orientado a objetos. Adaptado de [Travassos *et al.* 1999]

Resultados Experimentais – Em 1998, [Travassos *et al.* 1999] conduziram um estudo na Universidade de Maryland com o objetivo de avaliar a viabilidade da utilização de OORT's. O estudo utilizou 44 estudantes universitários como participantes.

Uma descoberta interessante foi que participantes aplicando técnicas verticais tenderam a detectar mais defeitos de omissão de funcionalidades e fatos incorretos, enquanto participantes aplicando técnicas horizontais detectaram significativamente mais defeitos de ambigüidade e inconsistência entre os modelos de projeto. Como resultado, revisores aplicando as diferentes técnicas supostamente detectaríamos diferentes tipos de defeitos, tornando a posterior reunião de inspeção mais eficiente.

O estudo serviu para identificar oportunidades de melhoria de OORT's. Segundo os autores de OORT's, a principal razão para o sucesso de PBR durante a inspeção de requisitos, é que a técnica requer um significativo processamento semântico do leitor; o leitor deve entender o documento o suficiente para produzir um artefato adicional. OORT's por sua vez exige um processamento sintático, onde os leitores são instruídos a verificar se determina informação está presente ou não em outras partes do documento. Em razão disso, os autores propuseram estender a técnica para instruir o leitor a criar um modelo físico (ex.: cenários de teste) para aumentar o seu entendimento semântico.

UBR (*Usage-Based Reading*) - UBR [Thelin 2004] é uma técnica de leitura que auxilia a detecção de defeitos severos do ponto de vista do usuário.

Descrição - UBR foca na leitura orientada por um modelo de casos de uso priorizados em ordem de importância para o usuário do sistema, durante a fase de preparação de um processo de inspeção de software. A premissa básica é permitir que as expectativas do usuário governem a inspeção. Os modelos de caso de uso poderiam ser utilizados para inspeções em todas as fases de desenvolvimento (requisitos, projeto, código etc) de um projeto específico. Durante a inspeção, os revisores leriam o documento executando manualmente os casos de uso e tentando detectar defeitos que são mais importantes de acordo com a prioridade estabelecida e, por conseguinte, para o usuário.

Segundo seus autores, conceitualmente UBR está relacionada com a perspectiva de usuário de PBR. Entretanto, os autores de UBR apontam algumas diferenças: (a) na perspectiva de usuário de PBR, revisores desenvolvem casos de uso baseados no documento de requisitos de forma a detectarem defeitos. Em UBR os casos de uso são utilizados como guias para os artefatos inspecionados; (b) o objetivo de UBR é melhorar a eficiência e a efetividade ao direcionar os esforços de inspeção para os casos de uso mais importantes sob o ponto de vista dos usuários; PBR tem como objetivo melhorar a efetividade da inspeção ao minimizar a sobreposição entre os defeitos encontrados pelos diferentes revisores; (c) os cenários de PBR são genéricos, ou seja, os cenários desenvolvidos para um tipo de artefato (ex.: requisitos) podem ser utilizados para todos os artefatos do mesmo tipo. Os cenários UBR, por sua vez, são específicos para cada projeto, o que significa que os casos de uso poderiam ser utilizados apenas dentro do projeto nos quais foram desenvolvidos. Entretanto, poderiam ser utilizados para inspeções de requisitos, projeto e código e apoiar a especificação de teste. Os autores não tecem comentários comparando UBR com a técnica de leitura vertical de OORT's que avalia modelos de projeto OO em relação a diagramas de casos de uso.

Resultados Experimentais - No contexto da revisão conduzida, encontramos quatro estudos experimentais envolvendo UBR (Tabela 2). A série de experimentos foi realizada em ambientes acadêmicos, na Suécia, utilizando estudantes universitários como participantes. O material inspecionado consistiu de um documento de requisitos descrito em linguagem natural e um modelo de projeto descrito em SDL [SDL 1993].

Em Lund-2000 foi observado que a atividade de ordenar os casos de uso, segundo a importância para o usuário, contribui para tornar a inspeção mais efetiva e eficiente, permitindo a detecção de defeitos críticos para o sistema. A atividade de ordenação provavelmente aumenta o entendimento do revisor sobre o funcionamento do sistema, representado pelos cenários dos casos de uso.

Tabela 2 - Série de experimentos envolvendo UBR

Estudo(Instituição–Ano)	Questão de Pesquisa
Lund-2000	A priorização de casos de uso afeta os resultados de UBR?
Blekinge-2001	UBR é mais eficiente e efetiva que <i>checklist</i> ?
Lund-2001	É necessário desenvolver casos de uso durante a inspeção UBR?
Lund-2003	UBR é mais eficiente e efetiva que <i>checklist</i> ? (repetição de Blekinge-2001)

Os estudos de Blekinge-2001 e Lund-2003 observaram que UBR foi mais eficiente e efetiva do que *checklist*. Além disso, UBR detectou defeitos que não foram detectados pelo uso de *checklist*. Finalmente, UBR possibilitou a detecção de defeitos

considerados críticos para o sistema, o que não ocorreu com *checklist*.

Em Lund-2001 foi observado que a utilização de casos de uso pré-desenvolvidos deixa os revisores mais focados na detecção de defeitos. Por outro lado, o desenvolvimento de casos de uso durante a inspeção levou à descoberta de novos defeitos, provavelmente devido ao aumento do grau de entendimento dos revisores sobre o sistema, confirmando [Travassos *et al.*1999]. Dessa forma, foi concluído que uma abordagem híbrida de UBR, onde casos de uso detalhados fossem desenvolvidos apenas para as funcionalidades mais cruciais do sistema, tornaria a inspeção mais efetiva. Uma observação, que merece atenção, é que a série de experimentos sobre UBR foi realizada pela mesma equipe que definiu a técnica, o que poderia ter acarretado inconscientemente o viés dos resultados [Silva e Travassos 2004].

MBR (*Metric-Based Reading*) - MBR [Bernardez *et al.* 2004] é uma técnica de leitura para detecção de defeitos em modelos de casos de uso.

Descrição - MBR adota um conjunto de heurísticas para orientar os revisores a detectarem defeitos em modelos de casos de uso. A premissa por trás dessas heurísticas é que os valores de certas métricas de casos de uso podem ser vistos como potenciais indicadores de defeitos. Os autores pesquisaram o relacionamento causa-efeito entre os valores dessas métricas e a presença de certos tipos de defeitos. Dessa forma, cada heurística apresenta uma faixa de valores onde, valores fora dessa faixa, seriam indícios de prováveis defeitos nos modelos de caso de uso.

Resultados Experimentais - A proposta da técnica MBR é relativamente recente (ano de 2004). Por isso, apenas um estudo experimental foi realizado investigando seus supostos benefícios, comparando MBR com a abordagem de *checklist*. O estudo utilizou 146 estudantes de um curso de graduação de ciência da computação que inspecionaram um documento contendo modelos de caso de uso. MBR foi mais efetiva do *checklist* no que diz respeito à quantidade de defeitos detectados. No que se refere à eficiência, os resultados não apontaram diferenças significativas do ponto de vista estatístico entre MBR e *checklist*. Os autores do experimento relatam a necessidade de novos experimentos para possibilitar a investigação dos pontos fortes e fracos de MBR.

5. Discussão dos Resultados

Diante dos resultados obtidos, PBR apresenta um campo mais maduro para discussão, visto que há uma quantidade significativa de estudos disponíveis na literatura investigando seus supostos benefícios e limitações. As outras técnicas encontradas (OORT's, UBR e MBR), embora possuam resultados de estudos incluindo aplicação industrial, ainda encontram-se em fase de investigação. Porém, os resultados promissores obtidos nos levam a crer também na viabilidade de utilização dessas técnicas. A seguir, discutimos as características de PBR sobre as quais foi possível identificar certo grau de consenso na comunidade acadêmica e o que ainda necessita de mais investigação, além das principais perspectivas futuras apontadas pelos estudos.

Indícios Obtidos: de acordo com os resultados obtidos, encontramos indícios que PBR apresenta melhor desempenho do que a abordagem *checklist*, no que se refere à efetividade de detecção de defeitos em documentos de requisitos. Além disso, a aplicação de PBR possibilita ganho considerável sobre o conhecimento do problema permitindo a identificação de defeitos mais severos para o sistema. Uma indicação forte desses estudos é de que PBR deve ser recomendada para inspeções curtas (2 horas) com equipes pequenas (3 revisores).

Questões em Aberto: apesar dos resultados promissores obtidos, certos pontos sobre PBR ainda não foram esclarecidos de forma satisfatória, requerendo maior investigação. Como exemplo, citamos a necessidade de estudo sobre o efeito da aplicação de PBR em documentos de requisitos maiores, que sejam representativos do ambiente industrial. Muitos dos estudos observados utilizaram documentos que não possuíam o tamanho normalmente encontrado na indústria. Outro importante fator a ser investigado é qual seria a combinação ótima entre limite de tempo e quantidade de participantes durante uma inspeção PBR. A hipótese de que PBR diminui a sobreposição de defeitos encontrados pelas diferentes perspectivas ainda não foi estudada de forma satisfatória. Uma possível alternativa seria direcionar a aplicação de cada perspectiva a um determinado conjunto de características de qualidade. Além disso, é necessário avaliar o impacto de fatores culturais nos resultados da leitura PBR [Shull *et al.* 2004].

Perspectivas Futuras: como sugestão para estudos futuros, indicamos a necessidade de uma avaliação mais rigorosa sobre a conformidade com a qual a técnica PBR foi executada. A avaliação da qualidade dos modelos físicos gerados pela aplicação de PBR poderia prover indícios de que o participante aplicou PBR de forma satisfatória. Um outro ponto sugerido é a caracterização dos participantes do estudo. A aplicação de pré-avaliações poderia minimizar possíveis discrepâncias entre o nível de conhecimento que o participante assumiu possuir e o que ele realmente possui.

[Biffi e Halling 2000] sugerem a utilização de um banco de dados central de defeitos durante a inspeção, para minimizar a detecção de defeitos repetidos. Além disso propõem a utilização em conjunto de PBR com a abordagem de *checklists*, visando a avaliar qual o nível de complementaridade entre as técnicas.

[Laitenberger *et al.* 2002] sugerem a caracterização do tamanho do documento inspecionado sob várias unidades de medida, além do número de páginas (número de requisitos, cenários, interfaces etc). Uma hipótese dos autores é que um grande número de requisitos ou cenários aumenta a complexidade mental requerida para inspecionar o documento e, portanto, a probabilidade de injeção de defeitos.

6. Garantia da Qualidade em Sistemas OO – Perspectivas Futuras

A aplicação de técnicas de leitura poderia aumentar a qualidade no desenvolvimento de sistemas OO ao permitir, ainda na fase inicial do projeto: (a) a identificação de defeitos críticos e (b) o aprendizado sobre o problema para o engenheiro de software.

Quanto à identificação de defeitos críticos, resultados do estudo de Maryland-1998, onde participantes aplicando a leitura vertical OORT's comparando modelos de projeto OO com documentos de requisitos detectaram defeitos do tipo omissão de funcionalidades e fatos incorretos, reforçam essa idéia.

Além disso, a aplicação de técnicas de leitura seria uma alternativa viável para a construção de modelos iniciais de projeto OO de qualidade. Estudos envolvendo PBR demonstraram que a construção de modelos físicos, como forma de possibilitar a detecção de defeitos em documentos de requisitos, permitiu um maior aprendizado sobre o problema descrito nos requisitos. Por sua vez, estudos envolvendo OORT's e UBR demonstraram a viabilidade da utilização de modelos de casos de uso na avaliação do comportamento dinâmico de sistemas OO.

Diante dos resultados obtidos, notamos a carência na literatura de técnicas de leitura que explorem a construção de modelos de projeto OO em sua aplicação. Estudos envolvendo a perspectiva de projetista adotada por PBR utilizaram técnicas de análise

estruturada para construção de diagramas de fluxo de dados. Por sua vez, UBR e OORT's, embora relacionadas a inspeções de modelos de projeto OO, não prevêm a construção de tais modelos no processo de inspeção. Assim, a definição e avaliação experimental de técnicas de leitura que explorem a construção de modelos de projeto OO passa a ser um desafio de pesquisa interessante.

7. Conclusões

A aplicação de atividades de verificação e validação em sistemas OO tem se mostrado difícil e pouco eficiente devido ao aumento de complexidade alcançado com o advento do paradigma OO. Nesse sentido, descrevemos como a condução de inspeções poderia aumentar a qualidade do software OO, ao permitir a remoção de defeitos no ponto onde eles são menos custosos de serem removidos: a fase de requisitos. Além disso, discutimos como academia e indústria de software têm trabalhado em conjunto para aumentar o desempenho de inspeção propondo a utilização de técnicas de leitura.

Como principal contribuição do artigo, destacamos a apresentação de um estudo secundário, também conhecido como revisão sistemática, conduzido com o objetivo de identificar, analisar e avaliar técnicas de leitura para documentos de requisitos propostas na literatura. O artigo descreve explicitamente o objetivo da revisão e os critérios adotados para a seleção de fontes e estudos. A condução da revisão sistemática permitiu, para cada técnica identificada, o levantamento de seus pontos fortes e fracos obtidos experimentalmente, possibilitando assim o direcionamento da pesquisa na área de técnicas de leitura e a indicação do que supostamente funciona e o que não funciona na aplicação dessas técnicas. No futuro, pretendemos conduzir estudos experimentais investigando a aplicação de técnicas de leitura que explorem a construção de modelos de projeto OO. Esperamos assim, estar contribuindo para a busca de qualidade no desenvolvimento de software no que se refere a métodos de garantia da qualidade.

Agradecimentos

Os autores agradecem o apoio financeiro do CNPq para a realização desse trabalho.

Referências

- Basili, V., Green, S., Laitenberger, O., Lanubile, F., Shull, F., Sørungård, S., Zelkowitz, M. (1996) "The Empirical Investigation of Perspective-Based Reading", *Empirical Software Engineering: An International Journal*, 1(2): 133-164, 1996.
- Basili, V. (1997) "Evolving and Packaging Reading Technologies", *The Journal of Systems and Software*, 38(1): 3-12, July 1997.
- Bernardez, B., Genero, M., Duran, A., Toro, M. (2004) "A controlled experiment for evaluating a metric-based reading technique for requirements inspection", *Software Metrics, 2004. Proceedings of 10th International Symposium on 14-16 Sept. 2004 Pages: 257 - 268*.
- Biffl, S.; Halling, M. (2000) "Software product improvement with inspection", *Euromicro Conference, 2000. Proceedings of the 26th, 5-7, Sept. 2000 Pages:262 - 269 vol.2*
- Biolchini, J., Mian, P.G., Natali, A.C., Travassos, G.H. (2005) "Systematic Review in Software Engineering: Relevance and Utility", *Relatório Técnico, PESC - COPPE/UFRJ*.
- Chernak, Y. (1996) "A Statistical Approach to the Inspection Checklist Formal Synthesis and Improvement", *IEEE Transactions on Software Engineering*, 22(12):866-874, December.
- Ciolkowski, M., Laitenberger, O., Rombach, D., Shull, F., Perry, D. (2002) "Software Inspections, Reviews & Walkthroughs", *24th ICSE, May 19 - 25, 2002, Orlando, Florida*.
- Chrissis, M. B., Konrad, M., Shrum, S. (2003) "CMMI: Guidelines for Process Integration and Product Improvement", *Addison-Wesley*.
- Fagan M. E. (1976) "Design and Code Inspection to Reduce Errors in Program Development", *IBM Systems Journal* 15-3, 182-211.

- Hatton, L. (1998) "Does OO really match the way we think?", *IEEE Software*, 15(3):46-54, May/June.
- Heninger, K. (1980) "Specifying Software Requirements for Complex Systems: New Techniques and their Application", *IEEE Transactions on Software Engineering*, SE-6(1):2-13, January.
- Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., Emam, K., Rosenberg, J. (2002) "Preliminary Guidelines for Empirical Research in Software Engineering", *IEEE Transactions on Software Engineering*, vol. 28, n° 8, August 2002.
- Kitchenham, B. (2004) "Procedures for Performing Systematic Reviews", Joint Technical Report Software Engineering Group, Department of Computer Science Keele University, United King and Empirical Software Engineering, National ICT Australia Ltd, Australia.
- Knight, J., Myers, E. A. (1993) "An Improved Inspection Technique", *Communications of the ACM*, 36(11): 51-61, November, 1993.
- Laitenberger, O., DeBaud, J-M (1998) "An Encompassing Life-Cycle Centric Survey of Software Inspection", *ISERN-98-32*.
- Laitenberger, O.; Beil, T.; Schwinn, T. (2002) "An industrial case study to examine a non-traditional inspection implementation for requirements specifications", *Proceedings of the 8th IEEE Symposium on Software Metrics*, 4-7, June 2002, Pages: 97-106.
- Leite, J. C., Doorn, J., Hadad, G., Kaplan, G. (2005) "Scenario inspections", *Requirements Engineering*, Volume 10, Issue 1, January 2005, Pages: 1-21
- Lima, G., Travassos, G. H. (2004) "Testes de Integração Aplicados a Software Orientado a Objetos: Heurísticas para Ordenação de Classes", In: III SBQS, Brasília, DF.
- Mafra, S., Travassos, G. (2005) "Uma Revisão Sistemática Aplicada a Técnicas de Leitura", *Monografia Final do Curso de Engenharia de Software OO*, PESC, COPPE/UFRJ.
- Mendes, E., Kitchenham, B. (2005) "Protocol for Systematic Review", Available at <http://www.cs.auckland.ac.nz/emilia/srsp.pdf>. Last accessed by 05/10/2005.
- Mills, H. (1991) "Cleanroom Engineering", *American Programmer*, Pages 31-37, May 1991.
- Porter, A, Votta, L (1994) "An experiment to assess different defect detection methods for software requirements inspections", *Proc. of the 16th international conference on Software engineering*, May.
- Qualis (2004) *Qualis - Sistema de Classificação de Periódicos, Anais e Revistas*, <http://qualis.capes.gov.br>, acessado em 20-12-2004.
- Regnell, B., Runeson, P., Thelin, T. (2000) "Are the Perspectives Really Different? - Further Experimentation on Scenario-Based Reading of Requirements", *Empirical Software Engineering*, Volume 5, Issue 4, Pages: 331-356, Dec 2000.
- SDL (1993) *Specification and Description Language*, ITU-T Recommendation Z.100.
- Shull, F. (1998) "Developing Techniques for Using Software Documents: A Series of Empirical Studies", PhD Thesis, Department of Computer Science, University of Maryland, USA.
- Shull, F., Mendonça, M., Basili, V., Carver, J., Maldonado, J., Fabbri, S., Travassos, G., Ferreira, M. (2004) "Knowledge-Sharing Issues in Experimental Software Engineering", *Empirical Software Engineering*, Volume 9 Issue 1-2.
- Silva, L., Travassos, G. (2004) "Tool-supported unobtrusive evaluation of software engineering process conformance", *Proceedings of the International Symposium on Empirical Software Engineering*, 19-20 Aug. 2004 Pages:127 - 135, ISESE '04.
- Travassos, G., Shull, F., Fredericks, M., Basili, V. (1999) "Detecting defects in object-oriented designs: using reading techniques to increase software quality", *Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, Volume 34 Issue 10, October 1999.
- Thelin, T., Runeson, P., Wohlin, C. (2003) "An experimental comparison of usage-based and checklist-based reading", *Proceedings of the 10th International Symposium on Software Metrics (METRICS'04)*, August 2003.
- Thelin, T., Runeson, P., Wohlin, C., Olsson, T, Anderson, C. (2004) "Evaluation of Usage-Based Reading-Conclusions after Three Experiments", *Empirical Software Engineering*, Volume 9, Issue 1-2, March. Pages: 77-110.
- Weber, K., Rocha, A. R. *et al.* (2004) "Modelo de Referência para Melhoria de Processo de Software: uma abordagem brasileira", *XXX Conferencia Latino-americana de Informática, Arequipa - Peru*.