

UM PROCESSO DE INSPEÇÃO UTILIZANDO LEITURA BASEADA EM PERSPECTIVA APLICADO À ANÁLISE DE REQUISITOS DO *UNIFIED PROCESS*

PAULO JOSÉ DA COSTA JÚNIOR
WALCÉLIO L. MELO

UNIVERSIDADE CATÓLICA DE BRASÍLIA - UCB
Rua Delenda Rezende de Melo, nº561 Setor Universitário
Goiânia - GO - CEP: 74605-050

paulocostajr@beg.com.br
walcelio_melo@yahoo.com

Resumo

Este trabalho destaca a importância das inspeções para a melhoria da qualidade dos artefatos produzidos durante o desenvolvimento do software. Para tal, o Processo Unificado de Desenvolvimento de Software (*Unified Process* - UP) foi aprimorado através da integração da técnica Leitura Baseada em Perspectiva - PBR, a qual permite validar os artefatos produzidos pelas atividades da disciplina Requisitos do UP. A PBR visa auxiliar a identificação dos defeitos a partir do ponto de vista, ou perspectiva, daqueles que irão usar os artefatos, dando suporte ao trabalho de leitura dos documentos a serem inspecionados indicando como ler e o que procurar a fim de encontrar possíveis defeitos. Para tanto, originalmente, a PBR mune os revisores com cenários que os orientem a procurar defeitos que só seriam percebidos por aqueles que efetivamente utilizam os artefatos, a saber, em se tratando de especificações de requisitos: os projetistas, testadores e usuários. Para integrar de maneira efetiva a PBR com o UP, foram elaborados cenários específicos a partir das perspectivas dos atores envolvidos no processo de análise de requisitos do UP, ou seja, o analista de sistema, arquiteto, especificador de casos de uso e o projetista de interface com o usuário. Para avaliar a efetividade da identificação de defeitos e o custo eficiência da técnica, batizada de PBR-UP, foram realizados estudos de casos segundo as diretrizes do AINSI e do paradigma GQM - *Goal/Question/Metric*. À luz de tais estudos, conclui-se que a PBR-UP permitiu encontrar de maneira eficiente vários defeitos nos artefatos produzidos pelo processo de análise de requisitos do UP.

Palavras chave: Análise de Requisitos, GQM, Inspeção, Leitura Baseada Em Perspectiva - PBR, Qualidade de Software e *Unified Process*.

1. Introdução

Está comprovado que quanto mais cedo for detectado e corrigido um defeito, menor será o impacto e a propagação nas demais fases do ciclo de vida [2]. Desta forma, devemos procurar identificar os defeitos o mais breve dentro do processo de desenvolvimento, ou seja, já a partir da identificação dos requisitos do sistema. O pior resultado obtido por um sistema de informação é quando seus requisitos não correspondem àqueles esperados por seus usuários. Se a análise de requisitos for “pobre”, todo o restante do processo estará comprometido e o risco de insucesso do projeto será bastante alto. Empresas do porte de IBM e HP têm avaliado os custos de erros que ocorrem em várias fases do desenvolvimento de software. Embora sejam estudos independentes, eles chegam à mesma conclusão: o valor para detecção e correção de um defeito na fase de codificação é entre 5 a 10 vezes superior ao custo para detectar e corrigir um defeito durante a fase de requisitos. Se o custo for calculado na fase de manutenção, ele pode chegar a 200 vezes o valor identificado durante a fase de requisitos [9].

Para auxiliar na identificação de defeitos durante a etapa de análise de requisitos podemos recorrer às inspeções. Inspeções são uma das poucas técnicas de revisão disponíveis para aplicação em artefatos de software que não o código-fonte, podendo ser aplicadas a qualquer tipo de documento escrito, sejam eles especificações de requisitos, documentos ou modelos de projeto, planos de teste e até aos próprios códigos-fonte. Embora importantes, os procedimentos genéricos de inspeção geralmente não fornecem nenhum tipo de assistência individual ao revisor no sentido de ajudá-lo a se preparar para as reuniões de inspeção. Na intenção de auxiliar os revisores, em suas atividades individuais de inspeção de requisitos, Basili e colaboradores desenvolveram uma técnica chamada Leitura Baseada em Perspectiva - PBR¹ [1]. A PBR fornece instruções concretas aos leitores indicando como ler e o que procurar possibilitando aos revisores encontrar mais defeitos em menos tempo. Esta técnica explora o fato de que, durante o ciclo de vida de um sistema, existem vários clientes/consumidores para os documentos produzidos pela etapa de análise de requisitos e procura oferecer a cada um dos revisores uma visão, ou perspectiva, específica e associada a estes clientes, a partir da qual os requisitos são inspecionados. Desta forma, procura-se garantir que todos os clientes / consumidores estariam representados dentro da equipe de inspeção através dos papéis desempenhados pelos revisores, cada qual utilizando uma determinada perspectiva. O princípio utilizado é que se vários inspetores podem revisar os documentos a partir de diferentes pontos de vistas, então, quando estes forem colocados juntos, adquirir-se-á muito mais confiabilidade sobre a qualidade dos documentos.

As perspectivas inicialmente definidas para esta técnica, e presentes em quase todos processos de desenvolvimento, são: usuários, testadores e projetistas. Para cada perspectiva, o inspetor é orientado a criar uma abstração em alto nível de um artefato de software que normalmente seria criado por estes clientes/consumidores [5]. Na criação deste artefato, utilizando a perspectiva do usuário, testador ou projetista, o revisor descobriria os defeitos. Ao contrário de outros métodos de inspeção, a PBR oferece a cada revisor um único e específico foco. Isto possibilita criar procedimentos específicos e experimentados para auxiliar os revisores a avaliar os documentos a partir de importantes pontos de vista associados ao processo de desenvolvimento.

Para auxiliar o revisor dos documentos durante o processo de leitura, são desenvolvidas certas descrições operacionais, chamadas de cenários, para cada um dos papéis. Um cenário consiste de um conjunto de questões e atividades que direcionam o leitor a respeito do que se deve fazer e como se deve ler o documento. Enquanto responde as várias questões e executa as atividades, o leitor deve ir anotando todos os defeitos encontrados no produto. Um cenário descreve certas atividades que devem ser executadas pelos inspetores durante a leitura dos documentos. Junto com as atividades, os cenários contêm algumas questões relacionadas às atividades. Mais precisamente, um cenário é uma coleção de procedimentos que operacionalizam estratégias para a detecção de defeitos.

A meta deste artigo é apresentar a PBR-UP, ou seja, uma técnica de inspeção baseada na PBR integrada ao *Unified Software Development Process* [7]. Para tal, introduzimos no modelo de processo de desenvolvimento de software - *Unified Process*, procedimentos de inspeção, inspirados na PBR, visando melhorar a qualidade dos artefatos produzidos pela disciplina requisitos. Para integrar de maneira efetiva a PBR com o UP, elaboramos cenários específicos baseados nas perspectivas dos diferentes atores / papéis envolvidos nas atividades de Análise de Requisitos

¹ Do inglês *Perspective-Based Reading*

prevista no UP, a saber: o analista de sistema, arquiteto, especificador de casos de uso e o projetista de interface com o usuário. Objetivou-se com isso atribuir mais qualidade aos produtos gerados por este modelo de processos no que tange a Análise de Requisitos e por conseqüência a todo o projeto. Para avaliar a efetividade da identificação de defeitos e o custo eficiência da técnica, um estudo de caso foi realizado. Tal estudo de caso foi conduzido segundo as diretrizes do AINSI [3] e utilizando o paradigma GQM - *Goal/Question/Metric* [10]. A luz de tal estudo de caso conclui-se que a PBR-UP permitiu encontrar de maneira eficiente vários defeitos nos artefatos produzidos pelo processo de análise de requisitos do UP.

2. PBR-UP

Cada participante envolvido no processo de desenvolvimento de software executa um conjunto de atividades procurando alcançar certos resultados que se traduzem em artefatos pertencentes ao modelo de processo. Estes artefatos são necessários para cumprir os objetivos específicos das disciplinas do processo. Cabe a nós garantir, através de mecanismos de inspeção (aqui baseados em PBR), que os requisitos possibilitarão que cada artefato seja produzido em conformidade com os objetivos da disciplina, identificando questões e cenários que ajudariam cada participante a desempenhar melhor o seu papel. Os participantes e os artefatos que são de responsabilidade da disciplina Requisitos do *Unified Process*, estão descritos na Figura 1 adaptada de [7], página 133:

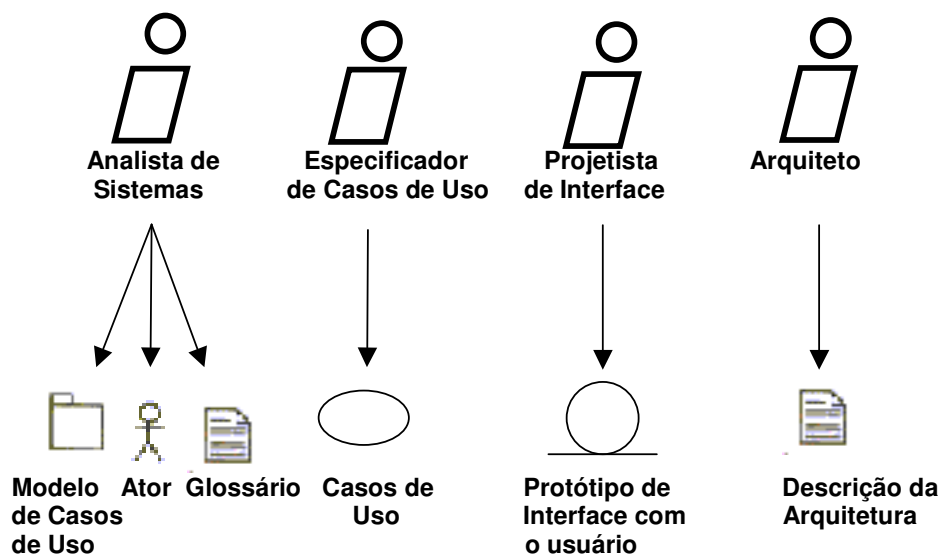


Figura 1 – Participantes e Artefatos da Disciplina Requisitos do *Unified Process*

De acordo com a PBR, as perspectivas utilizadas devem refletir o interesse dos participantes que utilizarão ou se beneficiarão dos artefatos que serão inspecionados. O conjunto de cenários desenvolvidos deve possibilitar a verificação dos documentos sob a ótica de quem irá utilizá-los, procurando classes de defeitos específicas e inerentes ao trabalho desses usuários. Deve ainda, concentrar o foco nos aspectos relacionados à perspectiva que está sendo usada no momento, evitando redundância em relação às demais, criando condições para a identificação antecipada destas classes de defeitos e impedindo que sejam descobertos pelos verdadeiros usuários dos produtos somente após os mesmos já terem sido entregues.

Ao observarmos a disciplina de requisitos do *Unified Process*, identificamos como participantes ativos os seguintes papéis: o Analista de Sistema, o Arquiteto, o Especificador de Casos de Uso e o Projetista de Interfaces com o Usuário. Em se tratando de qualidade de software, podemos dizer que todos os participantes ativos também são beneficiários do processo, pois precisam assegurar que os produtos gerados por eles terão o nível de qualidade esperado/exigido pela organização. Nas palavras de Humphrey: "Independentemente dos métodos e das ferramentas

disponíveis, o fator mais importante em um programa de qualidade é o compromisso pessoal do engenheiro de software em desenvolver um produto de qualidade. Quando os desenvolvedores estão comprometidos com a qualidade, eles tomam mais cuidado com o seu trabalho e se orgulham dos produtos que produziram” [6]. Sendo assim, devemos considerar a perspectiva de cada um dos participantes ativos, no sentido de criarmos cenários que possibilitem a revisão dos artefatos produzidos por eles de acordo com as normas estabelecidas em cada modelo de processos.

Existem também alguns participantes passivos, ou seja, aqueles que não participam diretamente de determinada atividade, mas, em algum outro instante, utilizarão os artefatos produzidos por outrem para a realização de suas próprias tarefas, seja durante a fase em questão ou *a posteriori*, todos identificados dentro do modelo de processo de software como tendo alguma responsabilidade específica.

Todas estas perspectivas somadas deverão ser consideradas, quando do trabalho de composição dos cenários, gerando um conjunto de questões que permitirá a detecção antecipada de defeitos, evitando que eles sejam descobertos apenas após a entrega dos produtos. Para auxiliar a elaboração das questões com base nas várias perspectivas, elaboramos um quadro de referência cruzada entre os artefatos que serão produzidos durante os requisitos *versus* os participantes que os utilizam, destacando o papel desempenhado por cada um deles e identificando se no momento são participantes ativos ou passivos, ou seja, se são respectivamente os produtores ou consumidores do artefato em questão. Como as inspeções são realizadas sobre os artefatos, vamos relacioná-los um a um e identificar quem se beneficia deles ao longo do processo de desenvolvimento do software. Os beneficiários serão classificados com base nos papéis descritos pelo *Unified Process*. Este cruzamento dará origem a algumas perspectivas que irão se somar àquelas já definidas na PBR, quais sejam: usuários, projetistas e testadores.

Embora as inspeções sejam realizadas sobre artefatos, a intenção aqui é criar perspectivas que reflitam o ponto de vista dos participantes. A estrutura da Tabela 1 demonstra com o QUÊ cada perfil de participante deve se preocupar quando certo artefato é inspecionado. Tomemos como exemplo a perspectiva do Analista de Sistemas, ilustrada a seguir:

Tabela 1 – Artefatos Relacionados com o Analista de Sistemas

Participantes que produz/consume	Artefato	Prod. / Cons.	Atividade onde é produzido/consumido	Disciplina onde é produzido/consumido
Analista de Sistema	Modelo de Casos de Uso	P	Encontrar os atores e os casos de uso	Requisitos
Analista de Sistema	Glossário	P	Encontrar os atores e os casos de uso	Requisitos
Analista de Sistema	Modelo de Casos de Uso	C	Estruturar o modelo de casos de uso	Requisitos
Analista de Sistema	Casos de Uso	C	Estruturar o modelo de casos de uso	Requisitos
Analista de Sistema	Glossário	C	Estruturar o modelo de casos de uso	Requisitos

Neste caso, ao desenvolvermos as questões e cenários considerando a perspectiva do Analista de Sistemas, preocupar-nos-emos se os requisitos estão devidamente declarados de forma a permitir que se encontre todos os atores e os casos de uso, se obtenha uma definição dos principais termos a fim de compor o glossário e que seja possível estruturar adequadamente o modelo de casos de uso. Ou seja, estaremos focando os objetivos específicos da perspectiva em questão para garantir que os artefatos são suficientes para cumprí-los.

2.1 Desenvolvendo novos Cenários para a PBR

Durante o processo de inspeção utilizando os cenários da PBR, freqüentemente o inspetor será solicitado a criar modelos de alto nível que são utilizados pelas respectivas perspectivas. Durante a confecção destes artefatos, o inspetor terá a oportunidade de se confrontar com uma classe de defeitos específicos que se apresenta sempre que aquela determinada perspectiva é invocada. A criação antecipada destes modelos por parte dos inspetores pode sugerir que um

retrabalho será realizado, entretanto, estes modelos poderão, e deverão, ser usados como base para a especificação dos modelos definitivos, economizando tempo e esforço quando da realização destas atividades.

Os cenários consistem em um conjunto de questões a serem respondidas e uma descrição das atividades que o inspetor deverá realizar a fim de possibilitar a identificação do maior número de defeitos sobre os requisitos. Como a própria técnica exige, as atividades de leitura devem manter o foco sobre uma classe específica de defeitos e evitar redundância em relação às responsabilidades dos demais inspetores. As questões serão formuladas com o propósito de identificar defeitos potenciais que atacam os artefatos manipulados exclusivamente pela perspectiva que está sendo considerada. Um cenário adequado para certa perspectiva deve orientar o revisor na identificação de problemas ou dificuldades que impeçam a consecução dos objetivos previstos nesta fase para aquela perspectiva, ou seja, deve procurar a partir dos requisitos, contradições que venham comprovar alguma inconsistência entre o que está definido e o que foi produzido. A seguir são apresentados os pontos que deram sustentação para a composição de cada um dos cenários, utilizando as perspectivas dos principais beneficiários dos requisitos segundo o *Unified Process*.

2.1.1 Cenário para o Analista de Sistema

O objetivo dos Analistas de Sistemas durante a Análise de Requisitos, de acordo com o *Unified Process*, é produzir um modelo de casos de uso bem estruturado e o glossário dos termos utilizados no sistema (ver tabela 1 *Artefatos Relacionados com o Analista de Sistemas*). Para alcançá-los, ele deve desempenhar um conjunto de atividades que visa estabelecer o contexto do sistema. O propósito é identificar claramente as fronteiras do sistema, ou seja, determinar o que deve ficar dentro e o que deve ficar fora do mesmo. As fronteiras serão delimitadas a partir da identificação dos atores e dos casos de uso.

O cenário desenvolvido para esta perspectiva procura atestar se os objetivos do analista de sistema podem ser cumpridos a partir da especificação dos requisitos do sistema. As atividades relacionadas para este cenário conduzem à melhor forma de identificar os elementos envolvidos nesta etapa (atores, casos de uso etc.) a partir da especificação dos requisitos, servindo de subsídio para a confecção dos modelos de alto nível produzidos durante a inspeção. Por outro lado, as questões, associadas às atividades, procuram verificar se os requisitos foram devidamente especificados de forma a permitir o cumprimento destas tarefas.

2.1.2 Cenário para o Especificador de Caso de Uso

O propósito principal deste cenário é verificar se é possível a partir dos requisitos obter uma descrição detalhada do fluxo de eventos dos casos de uso, elucidar as condições para que eles se iniciem e terminem, bem como identificar todas as interações realizadas com os atores. As fontes principais para a especificação dos casos de uso são o Modelo de Casos de Uso e o Glossário, conforme mostra a tabela 2.

Tabela 2 – Artefatos Relacionados com o Especificador de Casos de Uso

Participantes que produz/consume	Artefato	Prod. / Cons.	Atividade onde é produzido/consumido	Disciplina onde é produzido/consumido
Especificador de Casos de Uso	Casos de Uso	P	Detalhar os casos de uso	Requisitos
Especificador de Casos de Uso	Modelo de Casos de Uso	C	Detalhar os casos de uso	Requisitos
Especificador de Casos de Uso	Glossário	C	Detalhar os casos de uso	Requisitos

2.1.3 Cenário para o Arquiteto

Através da arquitetura do sistema, o arquiteto pode controlar o desenvolvimento a partir de uma perspectiva técnica, utilizando elementos estruturais, tais como classes, subsistemas e componentes. A arquitetura do sistema é construída através das várias iterações realizadas principalmente na fase de Elaboração e materializada pelo artefato denominado *descrição da arquitetura do sistema*. Cada iteração atravessa as disciplinas de requisitos, análise, projeto

implementação e testes desenvolvendo seus próprios modelos arquiteturais guiados pelos casos de uso mais relevantes para o sistema. Casos de uso relevantes são aqueles idealizados para combater os principais riscos do projeto, ou para implementar as atividades mais importantes para os usuários, ou ainda, aqueles que implementam funcionalidades críticas descritas pelos requisitos suplementares.

A descrição da arquitetura relaciona as partes mais importantes do sistema, aquelas que devem ser plenamente entendidas pelos desenvolvedores e demais pessoas envolvidas no sistema. É elaborada a partir de visões dos principais modelos utilizados no processo de desenvolvimento: casos de uso, análise, projeto, implementação e entrega. Ela vai sendo construída e complementada à medida que as disciplinas evoluem; guiada pelos casos de uso relevantes. Sendo assim, o cenário criado a partir da perspectiva do arquiteto consiste basicamente em verificar se é possível criar um modelo de casos de uso sintético que contemple os atores e casos de usos mais significantes arquiteturalmente (ou seja, os casos de uso relevantes). A tabela 3 mostra as fontes principais para a descrição da arquitetura e as atividades executadas pelo arquiteto durante o desenvolvimento do sistema.

Tabela 3 – Artefatos Relacionados com o Arquiteto

Participantes que produz/consome	Artefato	Prod. / Cons.	Atividade onde é produzido/consumido	Disciplina onde é produzido/consumido
Arquiteto	Descrição da Arquitetura	P	Priorizar os casos de uso	Requisitos
Arquiteto	Modelo de Casos de Uso	C	Priorizar os casos de uso	Requisitos
Arquiteto	Glossário	C	Priorizar os casos de uso	Requisitos
Arquiteto	Modelo de Casos de Uso	C	Analisar a arquitetura	Análise
Arquiteto	Descrição da Arquitetura	C	Analisar a arquitetura	Análise
Arquiteto	Modelo de Casos de Uso	C	Projetar a arquitetura	Projeto
Arquiteto	Descrição da Arquitetura	C	Projetar a arquitetura	Projeto
Arquiteto	Descrição da Arquitetura	C	Implementar a arquitetura	Implementação

Apesar de o arquiteto ter uma participação efetiva em todos os fluxos de trabalho do processo, na definição deste cenário o foco foi apenas no papel que ele desempenha frente aos requisitos de software.

2.1.4 Cenário para o Projetista de Interface com o Usuário

A função do Projetista de Interface com o Usuário é desenvolver os protótipos de interface do sistema com os usuários, ou seja, conceber modelos lógicos de interfaces para posteriormente implementá-los fisicamente (geralmente utilizando alguma ferramenta de prototipação) com a intenção de expressar e obter concordância em relação à forma como os atores irão interagir com o sistema, colocando à prova os atributos de usabilidade do sistema. Os artefatos básicos utilizados por ele para a especificação das interfaces são o Modelo de Casos de Uso, os Casos de Uso e o Glossário, conforme mostra a tabela 4, bem como os requisitos suplementares do sistema.

Tabela 4 – Artefatos Relacionados com o Projetista de Interface

Participantes que produz/consome	Artefato	Prod. / Cons.	Atividade onde é produzido/consumido	Disciplina onde é produzido/consumido
Projetista de Interfaces	Protótipo da Interface	P	Desenvolver os Protótipos de Interface do Sistema com Usuário	Requisitos

Projetista de Interfaces	Modelo de Casos de Uso	C	Desenvolver os Protótipos de Interface do Sistema com Usuário	Requisitos
Projetista de Interfaces	Casos de Uso	C	Desenvolver os Protótipos de Interface do Sistema com Usuário	Requisitos
Projetista de Interfaces	Glossário	C	Desenvolver os Protótipos de Interface do Sistema com Usuário	Requisitos

A preocupação básica deste cenário é validar se todos os elementos de interface do sistema com os usuários estão presentes nos protótipos.

2.1.5 Verificar Atributos de Qualidade dos Requisitos

Existe uma avaliação que, independentemente da perspectiva, deve ser conduzida antes do processo de avaliação final do resultado das inspeções. Estas questões servem de parâmetro para verificar a qualidade dos requisitos mediante alguns atributos preestabelecidos. Quanto mais aderente a estes atributos, melhor tende a ser especificação dos requisitos. Os atributos a serem avaliados são os seguintes:

Clareza

- Os requisitos foram escritos em linguagem não-técnica permitindo um bom grau de entendimento?
- Existe ambigüidade em alguma declaração de requisito?
- Foi elaborado um glossário de termos para auxiliar no entendimento?

Completeza

- Todos os termos necessários foram definidos?
- Algum requisito deveria ter sido especificado em mais detalhes?
- Algum requisito deveria ter sido especificado em menos detalhes?
- Existe algum requisito que não possibilita seu entendimento?
- Existem requisitos associados aos atributos de qualidade do software (requisitos não-funcionais)?

Consistência

- Existe algum requisito que descreve duas ou mais ações que conflitam logicamente?
- Existe algum requisito impossível de ser implementado?
- Cada um dos requisitos é relevante para a solução do problema?

2.2 Avaliando os Resultados

Para medir a efetividade da identificação de defeitos e o custo eficiência do processo de inspeção de requisitos com PBR, vamos escolher uma perspectiva específica e recorrer ao paradigma *Goal / Question / Metric* – GQM. O GQM é um paradigma orientado a objetivos que permite estabelecer uma estrutura mestra de avaliação desenvolvendo questões acerca de como estes objetivos ou metas podem ser alcançados. Através dele, estabeleceremos os objetivos (*Goal*) do processo de inspeção, ou seja, o que pretendemos alcançar implantando processos de revisão. A fim de quantificar estes objetivos formularemos algumas questões (*Question*) que serão respondidas utilizando-se métricas (*Metrics*) específicas que permitam avaliar os resultados. Devido à restrições de espaço, estaremos mostrando aqui somente os objetivos.

Objetivos:

1. Avaliar o custo eficiência do processo de inspeção de requisitos com PBR-UP.
O custo eficiência refere-se à quantidade de recursos despendidos em uma atividade necessários para a produção dos resultados obtidos. É definida pela relação entre resultados produzidos / recursos consumidos. Este objetivo possibilitará detectar se os procedimentos adotados para a realização das inspeções estão sendo realizados de forma racional e positiva e será expresso principalmente pelo custo relativo do processo de inspeção.

2. Avaliar a efetividade na identificação de defeitos do processo de inspeção de requisitos com PBR-UP.

A efetividade na identificação de defeitos refere-se à quantidade de defeitos verdadeiros encontrados pela técnica. É definida pela relação entre resultados pretendidos / resultados obtidos. Este objetivo possibilitará detectar se os procedimentos adotados para a realização das inspeções alcançarão os resultados esperados e será expresso principalmente pela quantidade de defeitos encontrada durante o processo de inspeção.

3. Estudo de Caso

O Estudo de Caso descrito a seguir foi elaborado a partir de um projeto real e teve seus requisitos inspecionados por seis revisores. Os resultados das inspeções foram coletados com base no Registro de Inspeção de Requisitos de Software, um formulário específico para as inspeções baseadas em PBR-UP (ver Anexo I), e em entrevistas efetuadas com os revisores após a realização das inspeções. Devido às limitações de espaço, não será possível mostrar neste artigo tal formulário.

3.1 Preparação para o estudo de caso

Escolha do projeto: o primeiro passo para a realização do estudo de caso foi a escolha de um projeto para a aplicação da técnica PBR-UP. Os pré-requisitos para este projeto eram: ter sido desenvolvido utilizando-se a técnica de análise orientada a objetos e conter em sua especificação de requisitos os artefatos previstos pelo *Unified Process*. Optou-se então por um projeto de médio porte com 4 atores e 30 casos de uso, cujo objetivo principal é controlar e gerenciar o atendimento médico e odontológico de uma clínica.

Seleção dos inspetores: com o projeto em mãos, buscou-se a identificação de profissionais que pudessem realizar as atividades de inspeção utilizando os cenários previstos na PBR-UP. Os pré-requisitos eram: ter formação na área de informática (analistas, programadores, projetistas etc.) e possuir conhecimento em análise orientada a objetos e UML. Ao final foram escolhidos 6 analistas de sistemas com perfil profissional variado entre mais e menos experientes.

3.2 Preparação para a inspeção

Preparação do material: foi entregue a cada um dos analistas selecionados o conjunto das especificações de requisitos do referido sistema bem como os cenários definidos pela PBR-UP. Junto com esta documentação, foi entregue o Registro de Inspeção de Requisitos de Software (em branco), formulário sugerido pela PBR-UP para auxiliar o processo de inspeção.

Preparação dos inspetores: foi ministrada uma palestra de 01:30h aos inspetores a fim de explicar a técnica, como ela devia ser aplicada e os objetivos do estudo de caso. Foi solicitado aos inspetores que realizassem as inspeções rigorosamente conforme sugeriu os cenários da técnica. Todos deveriam preencher os formulários de apoio e principalmente o Registro de Inspeção de Requisitos de Software, observando rigorosamente o tempo gasto em cada atividade. O tempo gasto para ler o material de referência foi computado ao tempo total de preparação.

3.3 Realização da Inspeção Individual

Todos os inspetores realizaram suas inspeções de forma independente um dos outros. Durante a inspeção foi solicitado auxílio no preenchimento de um ou outro formulário e explicações sobre o nível de detalhamento que seria exigido. A explicação sobre o formulário foi dada, mas quanto ao nível de detalhamento foi sugerido que se aplicasse aquele que o inspetor achasse mais conveniente. Ao final do processo de inspeção, todos entregaram o Registro de Inspeção de Requisitos de Software preenchido.

3.4 Reunião de Inspeção

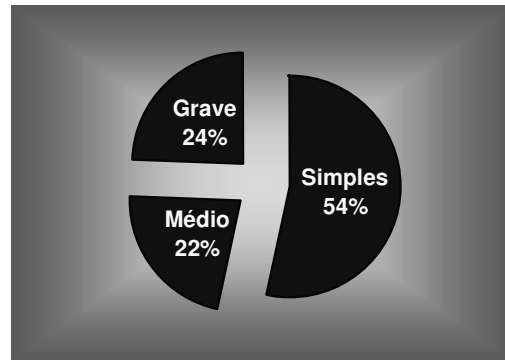
Após todos os inspetores terem realizado a inspeção, convidamos um dos produtores para discutir sobre os resultados da inspeção. As explicações do produtor e a troca de idéias durante a reunião de inspeção permitiram identificar os falsos erros descobertos pela inspeção. O percentual de acertos efetivos foi levantado a fim de avaliar a efetividade da técnica.

3.5 Análise dos Resultados

Severidade dos Erros: na Tabela 5 os resultados foram agrupados conforme a severidade dos erros. Este critério tenta definir a relevância da técnica em relação à importância dos erros que são identificados. Basicamente são considerados erros graves a inobservância de atores e casos de uso arquiteturalmente importantes para o projeto. Os erros simples concentraram-se, em sua grande maioria, no glossário e na nomenclatura indevida de atores e casos de uso; ao passo que os erros médios encontrados relacionaram-se às generalizações e relacionamentos não considerados ou considerados indevidamente. O maior número de erros foram os simples denotando pouca preocupação dos analistas em montar um glossário de termos, fato este aliás, freqüentemente esquecido por muitos desenvolvedores. Mesmo assim, a média de erros encontrados por inspetor para os tipos médios e graves foi substancial em relação ao tamanho do projeto.

Tabela 5 - Severidade dos Erros

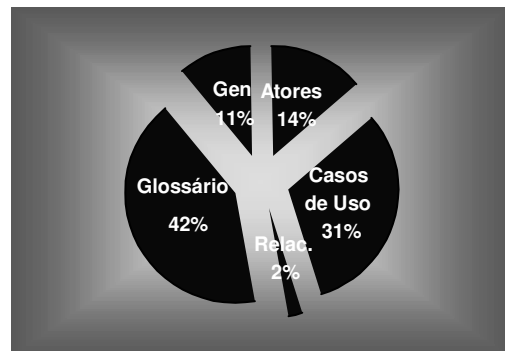
Inspetor	Erros Simples	Erros Médios	Erros Graves	Total
#1	15	6	2	23
#2	8	3	2	13
#3	0	1	3	4
#4	10	4	8	22
#5	9	6	5	20
#6	6	0	2	8
Total	48	20	22	90
Média	8,0	3,3	3,6	



Origem dos Erros: na Tabela 6 os resultados foram agrupados conforme a origem dos erros. Este critério tenta definir nos artefatos, os elementos mais suscetíveis à falhas. Pelo observado, o principal foco de dúvida recai sobre os atores, os casos de uso e seus relacionamentos. Nas observações dos inspetores (colhidas no Registro de Inspeção de Requisitos), fica claro a dependência em relação à qualidade da especificação da SRS, neste exemplo considerada de média qualidade (medida apurada com auxílio de um questionário para avaliação da qualidade da SRS). Note que foram encontrados mais erros nos atores e casos de uso que no próprio glossário, tal é o apego à preocupação com as funcionalidades do sistema, característica natural de quase todo desenvolvedor.

Tabela 6 - Origem dos Erros

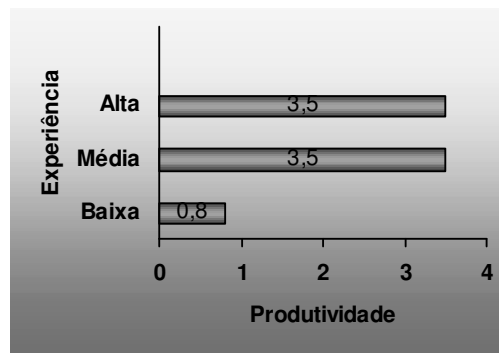
Inspetor	Ator	Caso de Uso	Glossário	Gene rali	Relac
#1	1	5	12	5	1
#2	3	5	5	0	0
#3	3	1	0	0	0
#4	3	9	6	4	0
#5	2	7	9	1	1
#6	1	1	6	0	0
Total	13	28	38	10	2
Média	2,16	4,66	6,3	1,66	0,33



Produtividade dos Inspetores: na Tabela 7 os resultados apresentados procuram registrar a produtividade obtida na aplicação da técnica. O inspetor que gastou menos tempo obteve índice de produtividade superior aos demais, isto foi conseguido principalmente porque ele se omitiu no preenchimento de alguns formulários, contribuindo para a diminuição do tempo total da inspeção e apoiado em sua experiência os erros se confirmaram como verdadeiros ao final. Nos casos de baixos índices de produtividade, um apresentou pouca motivação para a inspeção e baixa aderência à técnica; o outro não encontrou uma quantidade expressiva de erros devido à sua inexperiência nas atividades de desenvolvimento de sistema, resultando em um tempo de inspeção baixo em relação à média e poucos erros encontrados. Estes fatores contribuíram para que eles tivessem mais falsos positivos que os demais, impactando na efetividade de seus resultados. Os inspetores que demoraram mais tempo, seja devido à maior motivação ou à aderência à técnica, apresentaram resultados mais consistentes que seus colegas.

Tabela 7- Produtividade dos Inspetores

Inspetor	Experi- ência(*)	Tempo (h:m)	Qte. Erros	Erros/ Horas
#1	3	06:00	23	3,83
#2	2	02:00	11	5,50
#3	2	02:15	4	1,77
#4	2	05:00	22	4,4
#5	2	04:30	11	2,44
#6	1	02:30	2	0,8
Total		22:15	73	
Média	2	03:40	12,16	3,12

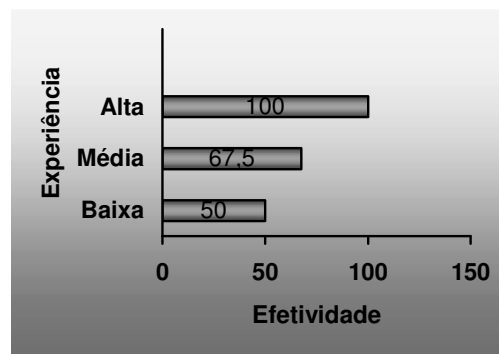


* A experiência do revisor foi medida através da aplicação de um questionário específico.

Efetividade: para se medir a efetividade da detecção de defeitos optamos por retirar da análise os erros simples que constituíram-se na sua grande maioria em falta de definição de termos no glossário, isto de justifica pelo fato de que qualquer que seja o termo ele sempre poderá suscitar dúvida ao leitor, não cabendo a nós declarar tal erro como falso. A partir daí, procuramos o percentual de acerto efetivo que na média se apresentou acima de 70%. Este índice pode ser considerado elevado para uma técnica experimental, entretanto, acreditamos que ele possa ser alcançado, sempre que houver uma disposição dos inspetores em seguir sistematicamente os preceitos sugeridos pela técnica, embora neste caso também esteja previsto um aumento do custo da inspeção face à maior quantidade de tempo que esta iniciativa ocasiona.

Tabela 8 - Efetividade

Inspetor	Erros Falsos	Qte. Erros Encontrados *	Taxa de Efetividade **
#1	0	8	100%
#2	2	5	60%
#3	2	4	50%
#4	3	12	75%
#5	1	7	85%
#6	1	2	50%
Média	1,5	6,3	70,0%



* Sem considerar os erros simples

** $(\text{Qte. Erros} - \text{Erros Falsos}) / \text{Qte. Erros} * 100$

4. Conclusão

Um estudo elaborado por Capers Jones, em 1994, demonstra que a taxa de efetividade média para a remoção de defeitos durante a fase de requisitos em organizações que utilizam uma combinação de testes, inspeções e outras estratégias gira em torno de 77% [9]. Através da PBR-UP, obtivemos em alguns casos taxas superiores a esta, estabilizando-se em uma média próxima a 70%. Baseado nos resultados obtidos, podemos dizer que inspetores de média a muita experiência que realizarem as inspeções, seguindo fielmente os cenários, podem obter altas taxas de efetividade na identificação e remoção dos defeitos.

Segundo [11], a implantação de processos de inspeção agregados aos processos de desenvolvimento de software podem consumir de 5 a 15% do orçamento do projeto. Em nosso caso, obtivemos um custo relativo de inspeção em torno de 4,2% do valor gasto com análise de requisitos utilizando apenas uma perspectiva. Se considerarmos toda as perspectivas, e se o custo das inspeções mantiver-se constante durante as demais fases do ciclo de vida, acreditamos que também encontraremos valores dentro dos limites aceitáveis para processos de inspeção. Futuros estudos deverão ser realizados para se comprovar tal afirmação.

Em [4] foi mostrado os resultados de um estudo onde 300 inspeções foram realizadas em mais de 10 projetos. A autora calculou que as inspeções custaram em média U\$105 por defeito. De acordo com o custo eficiência apurado pela PBR-UP, identificamos o valor de R\$ 22,00 por defeito. Vale ressaltar que o custo apurado no primeiro estudo envolve as etapas do processo de inspeção que vão desde a identificação do defeito até sua correção, ao passo que na PBR-UP apuramos apenas o custo de identificação do defeito. Além disso, o primeiro estudo foi realizado nos Estados Unidos, que, notadamente, representa uma realidade bem diferente da brasileira.

Em [8], são mostrados alguns resultados de um estudo comparativo entre inspeções realizadas com listas de verificação (*checklists*) versus leitura baseada em perspectiva. Neste estudo, ficou evidenciado que o grupo que realizou inspeções utilizando PBR conseguiu encontrar mais defeitos do que aqueles que utilizaram listas de verificação. O tempo gasto para se encontrar um defeito também foi menor. No caso da PBR, registrou-se uma taxa de 56 minutos por erro. Enquanto que utilizando listas de verificação o tempo registrado foi de 132 minutos para cada erro encontrado.

Acerca da utilização da PBR-UP, conforme os depoimentos dos inspetores, a técnica demonstrou ser de grande valia no auxílio ao trabalho de inspeção e por conseqüência na melhoria da qualidade do software. Os principais pontos positivos da técnica destacados nas entrevistas com os inspetores foram:

- Fornece uma diretriz consistente para o trabalho de inspeção de requisitos orientados a casos de uso;
- Identifica aspectos geralmente esquecidos pelos analistas de desenvolvimento;
- Cobre boa parcela dos principais defeitos encontrados no processo de levantamento de requisitos;
- Requer pouco tempo de aprendizado;
- Dissemina a cultura de busca da qualidade dentro da organização.

E os negativos:

- A atividade de confeccionar modelos de alto nível deve ser realizada com moderação, pois eleva o tempo de preparação para a reunião de inspeção;
- Exige-se conhecimento prévio de assuntos como *Unified Process* e UML para tirar melhor proveito da técnica;

Aplicando-se a PBR pudemos comprovar a efetividade do processo de inspeção na identificação de defeitos. Por exemplo, quando utilizamos a perspectiva do analista de sistemas, nos preocupamos em encontrar determinadas classes de defeitos, tais como funcionalidades não percebidas ou relacionamentos com certos atores que não estavam tão evidentes. Focar o processo de inspeção sob um ponto de vista específico, torna-o mais produtivo, pois substitui-se uma avaliação realizada freqüentemente *ad hoc*, por um processo dirigido e controlado. Com os cenários propostos pela PBR, o inspetor pôde identificar a maioria dos defeitos durante a preparação para a reunião de inspeção, enquanto lia os documentos orientando-se pela técnica proposta, haja vista que dispunha de todos os procedimentos declarados para validar os artefatos. Este fato influencia a produtividade das inspeções, uma vez que as reuniões poderão ser usadas apenas para apresentar e consolidar

os defeitos encontrados. Os artefatos produzidos durante o processo de inspeção não só validam o que foi feito, bem como subsidiam a criação dos documentos utilizados pelo analista, propiciando um intercâmbio de idéias benéfico para o desenvolvimento de sistemas.

Quanto ao custo eficiência do processo fica claro a dependência em relação à qualidade das especificações e à experiência do revisor. Especificações de requisitos que utilizam algum formalismo (por exemplo o uso de *templates*) pressupõem a existência de um caminho conhecido e experimentado para a definição dos requisitos. Estes formulários auxiliam na validação dos requisitos a partir do momento que induzem o analista a preencher informações que servirão como referência para sua posterior comprovação. Por outro lado, lidar com analistas mais experientes no papel de revisores algumas vezes leva à substituição dos procedimentos descritos nos cenários pela intuição ou pelo conhecimento prévio que os mesmos possuem, alterando a percepção sobre o verdadeiro resultado da inspeção realizada a partir do método referenciado. Entretanto, esta vinculação dos resultados obtidos ao conhecimento tácito dos analistas jamais poderá ser dissociada, pois trata-se de unidade indivisível já que sempre procuramos empregar todos os nossos recursos no desempenho de certa atividade.

Referências:

- [1] BASILI, Victor R., *"The Empirical Investigation of Perspective-Based Reading"*, University of Maryland, College Park, MD, USA., 1996
- [2] BOEHM, Barry, *"Industrial Software Metrics Top 10 List"*, IEEE Software, setembro de 1987, pp. 84-85. 1987
- [3] BRIAND, L. & EL EMAM, K. & MELO, L. Walcélío, *"AINSI - An Inductive Method for Software Process Improvement: Concrete Steps and Guidelines"*, ESI-ISCN'95: Measurement and Training Based Process Improvement, Viena, Áustria, 1995
- [4] BUSH, Marilyn, *"Improving Software Quality: The Use of Formal Inspections at the Jet Propulsion Laboratory"*, In *Proceedings of the 12th International Conference on Software Engineering*, páginas 196-199, 1990
- [5] CARVER, Jeffrey, *"Impact of Background and Experience on Software Inspections"*, Proposta de tese de doutorado. University of Maryland, 2000
- [6] HUMPHREY, Watts S., *"Introduction to the Personal Software ProcessSM"*, Carnegie Mellon University, The SEI Series in Software Engineering, Ed. Addison-Wesley, 1997.
- [7] JACOBSON, Ivar & BOOCH, Grady & RUMBAUGH, James, *"The Unified Software Development Process"*, Ed. Addison-Wesley, 1999.
- [8] LAITENBERGER, Oliver & EL EMAM, K. & ATKISON, Colin, *"Using Inspection Technology in Object-oriented Development Projects"*, Fraunhofer Institut Experimentelles Software Engineering, 2000. [On Line] 2003. [Citado em 10/03/2003]; Disponível da Internet em: http://www.iese.fhg.de/Publications/iese_reports_3/
- [9] LEFFINGWELL, Dean & WIDRIG, Don, *"Managing Software Requirements A Unified Approach"*, Ed. Addison-Wesley, 2000.
- [10] SOLINGEN, Rini Von & BERGHOUT, Egon , *"The Goal/Question/Metric Method"*, Ed. McGraw Hill, 1999.
- [11] WIEGERS, Karl E., *"Improve Quality Through Software Inspections"*, 1999. [On Line] 2002. [Citado em 12/04/2003]; Disponível da Internet em: <http://www.processimpact.com/>

Anexo I – Registro de Inspeção de Requisitos de Software

Registro de Inspeção de Requisitos de Software		
Nome do Projeto: Descrever o nome do projeto de forma a permitir o entendimento macro do problema		
Responsável pelo Projeto: Nome das pessoas envolvidas no projeto		
Horário de Início: Hora de início da inspeção	Encerramento: Hora de encerramento da inspeção	Duração: Duração da inspeção
Perspectiva utilizada: <input type="checkbox"/> Analista de Sistemas <input type="checkbox"/> Arquiteto <input type="checkbox"/> Especificador de Caso de Uso <input type="checkbox"/> Projetista de Interface	Nome do Revisor Nome do revisor	Tempo de Treinamento: Tempo de treinamento na técnica
ERROS VERDADEIROS	ERROS FALSOS	RESUMO
Qte. Atores Novos: G Qte. Atores Exc.: G Qte. Casos de Uso Novos: G Qte. Casos de Uso Exc.: G Qte. Generalizações Novas: M Qte. Generalizações Exc.: S Qte. Termos Novos: S Qte. Termos Exc.: S Qte. Relacionamentos Novos: M Qte. Relacionamentos Exc.: S	Qte. Atores Novos: : Qte. Atores Exc.: : Qte. Casos de Uso Novos: : Qte. Casos de Uso Exc.: : Qte. Generalizações Novas: : Qte. Generalizações Exc.: : Qte. Termos Novos: : Qte. Termos Exc.: : Qte. Relacionamentos Novos: : Qte. Relacionamentos Exc.: :	Qte. Erros Graves ...: G Médios ...: M Simples ...: S
Requisitos suplementares não declarados: Declaração dos atributos não funcionais aplicáveis ao projeto como um todo		
Observações Adicionais: Qualquer observação adicional que o revisor queira fazer sobre o projeto		
Assinatura Revisor:	Data: Data da inspeção	Resultado: <input type="checkbox"/> Reprovado <input type="checkbox"/> Aprovado c/ restrição <input type="checkbox"/> Aprovado

Obs.: a qualificação dos erros em relação à severidade pode ser afetada pela percepção do revisor ou pelo contexto da aplicação. Por exemplo: exclusão de ator é erro grave, porém pode ser considerado médio ou simples se a exclusão foi em decorrência da transformação de um ator em uma generalização ou apenas pela substituição de um outro já existente.