

Avaliação de uma Metodologia de Medição da Qualidade em um Documento de Requisitos Orientado a Aspectos

Ricardo Argenton Ramos / Jaelson F. B. Castro
Centro de Informática - Universidade Federal de Pernambuco – UFPE
Av. Prof. Luiz Freire S/N, Recife PE, Brasil 50732-970, +55 81 32718430
[rar2, jbc] @cin.Ufpe.br

Resumo

A área de desenvolvimento orientado a aspectos esta cada vez mais ampla, abrangendo todas as fases no processo de desenvolvimento de software. Pesquisas promissoras estão sendo feitas na fase de requisitos cujo o propósito é separar no documento de requisitos os aspectos (requisitos não funcionais) que estão espalhados e entrelaçados aos requisitos funcionais. Contudo, não se tem conhecimento de proposta para avaliar a qualidade da definição dos aspectos que são elicitados na fase de requisitos. Este artigo apresenta uma avaliação realizada através de um estudo de caso, da aplicação de uma metodologia de medição da qualidade de documentos de requisitos em um documento de requisitos orientado a aspectos.

Abstract

The aspect oriented development area is widening its scope, addressing all phases of the software development process. Indeed, recent proposals have looked at the requirements stage, trying to separate at the requirements documents the aspects (non functional requirements) that are spread and tangled with functional requirements. However, we are not aware of any proposals do evaluate the quality of these aspects discover during the requirements elicitation. This paper shows a evaluation based on case study of a measurement methodology of quality on requirements document in a aspect oriented requirements document.

1. Introdução

O uso de processos de desenvolvimento de software de modo inadequado gera grandes custos nas fases de manutenção deste software. De acordo com Pressman [11], o custo da correção de erros na fase de projeto é cerca de três a seis vezes mais alto do que na fase de definição de requisitos. Este custo aumenta ainda mais quando a correção de erros é realizada em fases mais avançadas no processo de desenvolvimento.

No sentido de melhorar o entendimento e conseqüentemente a qualidade do documento de requisitos, desta maneira antecipando a correção de

possíveis erros, alguns trabalhos propõem a utilização de técnicas de separação de interesses (do inglês *concerns*) utilizando o Desenvolvimento Orientado a Aspectos (DOA) na fase de requisitos [13], cuja a idéia principal é que documentos de requisitos são melhores entendidos quando os seus requisitos funcionais e não funcionais estiverem especificados separadamente. Além dessa separação de interesses as pesquisas nessa área fornecem algumas técnicas para modelagem e ainda como e onde posteriormente combinar o que foi separado [13, 16].

Apesar das prováveis facilidades que o DOA sugere para a fase de requisitos, não se encontra na literatura trabalhos que avaliem qualitativamente um documento de requisitos orientado a aspectos. Santa'Ana e outros [15] avaliam qualitativamente e quantitativamente os benefícios de implementações de padrões de projeto em sistemas orientados a aspectos e orientados a objetos, porém em seu trabalho [15] os documentos de requisitos não são avaliados.

Nesse artigo foi realizado um estudo de caso, em que se avalia a atividade de medição da qualidade de um documento de requisitos orientado a aspectos, gerado por uma abordagem guiada a casos de uso [16], aplicando uma metodologia que foi primeiramente elaborada para medir a qualidade de documentos de requisitos de sistemas Web [14].

O objetivo deste estudo de caso foi avaliar se a aplicação das métricas proposta por Reis [14] podem ser aplicadas (sem ou com modificações) em um documento de requisitos orientado a aspectos e desta maneira apontar possíveis vantagens e desvantagens nessa combinação.

Na Seção 2 estão sendo relatados alguns trabalhos na área do desenvolvimento orientado a aspectos na fase de requisitos, na Seção 3 é apresentada a metodologia de medição de qualidade de documentos de requisitos. A descrição de como foi realizado o estudo de caso é relatada na Seção 4, na Seção 5 são apresentadas algumas métricas sugeridas para a medição da qualidade dos aspectos e os comentários finais são apresentados na Seção 6.

2. Orientação a Aspectos na Fase de Requisitos

Produzir software com qualidade e num menor tempo possível é um dos principais objetivos da Engenharia de Software [11]. Para alcançar tal objetivo, essa área de pesquisa estabeleceu alguns princípios que devem ser aplicados ao longo do processo de desenvolvimento e nos artefatos de software: separação de Interesses, modularidade, rigor e formalidade, incrementalidade, abstração, generalidade e antecipação de mudanças [11].

O principal propósito do princípio da separação de Interesses é reduzir o raciocínio para uma quantidade factível [1]. Esse princípio estabelece que um problema deve ser decomposto em unidades menores, claramente separadas, de maneira que cada uma delas represente um único interesse.

O desenvolvimento orientado a aspectos surgiu como uma das áreas de pesquisas promissoras dentro da separação de interesses, onde a idéia principal é que sistemas são melhores entendidos se os aspectos (requisitos não funcionais) ficarem separados da parte funcional do software [6, 9].

Elrad e outros [6] afirmam que o desenvolvimento orientado a aspectos consiste na separação dos interesses de um sistema em unidades modulares e posterior composição (*weaving*) desses módulos em um sistema completo. Os interesses podem variar de noções de alto nível, como segurança e qualidade de serviço [13], a noções de baixo nível, como sincronização, manipulação de *buffers* de memória, tratamento de exceções, persistência entre outros [12].

A Programação Orientada a Aspectos (POA) trata os interesses que entrecortam as classes de modo análogo ao que a programação orientada a objetos faz para o encapsulamento e a herança. Ou seja, provê mecanismos que explicitamente capturam a estrutura de entrecorte, alcançando assim, os benefícios de melhor modularidade, tais como: código mais simples, mais fácil de manter e alterar e, conseqüentemente, aumento da reusabilidade [12].

Grande esforços dos pesquisadores tem sido em direção da implementação de sistemas orientados a aspectos [6, 12], porém recentemente algumas pesquisas surgem utilizando o desenvolvimento orientado a aspectos na fase de requisitos [4, 10, 13], onde tem-se o propósito de separar no documento de requisitos os aspectos (requisitos não funcionais) que estão espalhados e entrelaçados aos requisitos funcionais.

Rashid e outros [13] propuseram um modelo de processo genérico, denominado AORE (*Aspect-Oriented Requirements Engineering*), para separar interesses transversais no nível de requisitos. No modelo AORE, o termo “interesse” é utilizado num sentido mais restrito, correspondendo a um requisito não-funcional de alto nível de abstração (ex: segurança, compatibilidade).

Utilizando como base o processo de desenvolvimento unificado [8], Sousa [16] realiza uma adaptação desse processo para o desenvolvimento orientado a aspectos, onde o principal foco desta pesquisa é o desenvolvimento guiado por casos de uso [7], possibilitando a separação de interesses transversais nas fases de requisitos, análise e projeto utilizando fundamentos da programação orientada a aspectos.

Partindo do princípio de que aspectos são encapsulados nas mesmas unidades de decomposição existentes em cada um dos fluxos considerados, Sousa [16] cria um estereotipo especial para cada fase, como por exemplo, caso de uso aspectual (mostrado na Figura 1), classe de análise aspectual, para indicar que o comportamento do aspecto deverá ser inserido em alguma parte das unidades que ele afeta. O termo *crosscuts* refere-se ao relacionamento entre uma unidade aspectual e uma unidade afetada por ela. Uma tabela denominada de tabela de composição armazena para cada aspecto identificado e em cada fluxo, informações relativas à composição entre um aspecto e a(s) unidade(s) que ele afeta.

Para validar seu trabalho, Sousa [16] desenvolve com sua abordagem um sistema de Internet Banking Orientado a Aspectos. Seguindo a abordagem de Sousa o diagrama de casos de uso aspectual, mostrado na Figura 1, apresenta os casos de uso (requisitos funcionais), como proposto pela UML [7], e os casos de uso aspectuais, que neste caso modularizam o requisito não funcional de segurança. Esses casos de uso aspectuais estão modelados no formato de losangos e nos relacionamentos que indica que o caso de uso aspectual entrecorta um (ou mais) caso de uso, existe o estereótipo *crosscuts*.

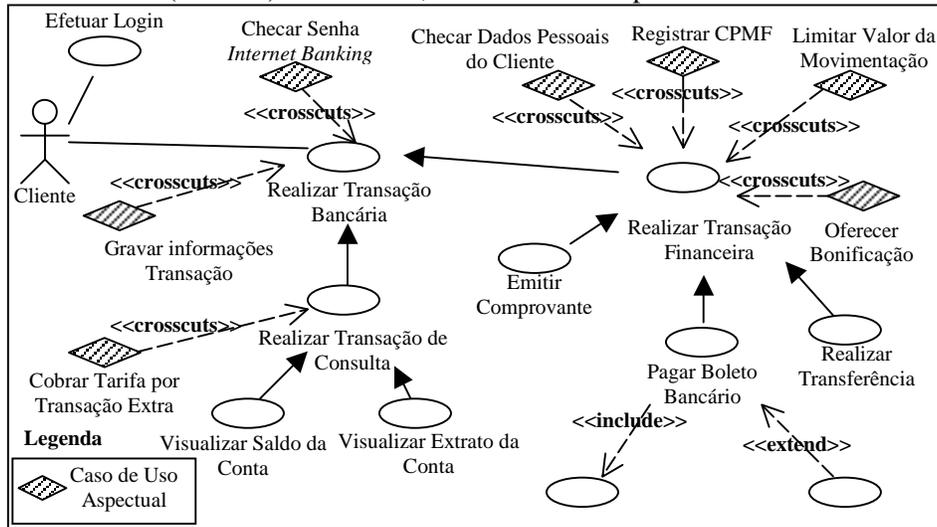


Figura 1 - Diagrama aspectual de casos de uso do sistema Internet Banking.

Para o contexto desse trabalho o termo interesse refere-se a um requisito não funcional, sendo que esse pode ser implementado em um ou mais aspectos. Portanto, no diagrama de casos de uso aspectual um interesse pode ser modelado em um ou mais casos de uso aspectuais.

A pesar das vantagens propostas pelo desenvolvimento orientado a aspectos na fase de requisitos, como um documento de requisitos melhor organizado e de mais fácil manutenção, não existe uma metodologia que avalie a qualidade desses documentos. Na Seção seguinte é apresentada uma metodologia relacionada com a verificação de qualidade. Será avaliado (na Seção 4) o comportamento desta metodologia com um documento de requisitos orientado a aspectos.

3. Metodologia para Medição de Qualidade de Aplicações Web na Fase de Requisitos

A comunidade de engenharia de software vem adaptando padrões existentes para que desta maneira se permita fornecer um suporte para que haja conformidade e assim possibilitando elaborar métricas para medir a qualidade

de um produto de software. São muitas as vantagens de se medir a qualidade de um produto de software e corrigir seus erros, inconsistências e sua completude, porém quanto mais cedo nas fases de desenvolvimento do software for descoberto os erros, menos custoso é o seu processo de correção [11]. Por esse motivo a medição da qualidade de um produto de software na fase de requisitos tem sido uma área bastante promissora.

Reis [14] propõe uma metodologia para medição da qualidade de aplicações Web na fase de requisitos denominada de REQE (*Requirements Engineering Quality Evaluation*). Esta metodologia se propõe a avaliar a qualidade de aplicações Web na fase inicial do processo de desenvolvimento, tendo assim como benefício à possibilidade de descobrir erros de uma forma antecipada. A aplicação da metodologia REQE consiste em cinco fases:

1. Representação das características, subcaracterísticas e atributos de qualidades;

Nessa primeira fase serão elicitados e especificados os requisitos de qualidade, culminando com uma árvore que, hierarquicamente, lista todas as características, subcaracterísticas e atributos de qualidade, capazes de modelar a qualidade segundo as necessidades do usuário. Cada atributo de qualidade deve ser medido por meio de um escore de conformidade. Este escore verifica se o produto atenderá ou não (com base no documento de requisitos) àquele atributo de qualidade.

2. Especificação descritiva da árvore de característica, subcaracterísticas e atributos de qualidade;

Nessa fase, todos os nós da Árvore de Características, Subcaracterísticas e Atributos de Qualidade (ACSAQ) devem ser especificados de forma descritiva. É preciso descrever cada nó detalhadamente, a fim de dirimir eventuais dúvidas a cerca de seus significados. A título de documentação, Reis [14] propõe que para cada característica, subcaracterística e atributo de qualidade, deve ser preenchida uma planilha de informação.

3. Associação de pesos aos nós;

Nessa fase da metodologia REQE deve-se determinar o grau de importância dos nós no contexto do domínio da aplicação, por meio da associação de peso a cada um dos nós. Primeiramente atribui-se pesos às características, em seguida as subcaracterísticas e aos atributos de qualidade.

Reis decidiu adotar o valor 10 (dez) para ser distribuído entre os filhos de cada nó da ACSAQ. Primeiramente deve-se distribuir 10 (dez) pontos entre as características presentes na árvore. São permitidos apenas números inteiros positivos. Essa pontuação vai representar o grau de importância daquela característica, na qualidade global da aplicação Web.

4. Associação de escores aos atributos de qualidade;

Ao iniciar essa etapa já se obteve a árvore de características, subcaracterísticas e atributos de qualidade estruturada, juntamente com as planilhas de informações preenchidas e os pesos associados a todos os nós das

árvore (que identificam a importância relativa de cada item). Para a atribuição de valores Reis [14] propõe a seguinte escala de conformidade, baseado em [2] e [3]:

<i>Conformidade</i>	<i>Escore</i>
Não atende o atributo de qualidade	0
Atende com sérias restrições	3
Atende parcialmente	6,5
Atende	10

Figura 2 – Tabela de conformidade x escore.

Para cada atributo de qualidade da ACSQA deve ser associado um escore, seguindo a escala da Figura 2. Caso o atributo não seja previsto no documento de requisitos, deve receber a conformidade “não atende” (escore 0), pois como o documento não prevê, é considerado que não atende o atributo de qualidade, conseqüentemente, o produto final dificilmente atenderá.

5. Cálculo geral.

O objetivo dessa fase é efetuar o cálculo que consolida todos os escores de conformidade associados, levando em consideração os pesos dos nós. Ao final desta fase a avaliação da qualidade é concluída e a possibilidade de corrigir os erros identificados (atributos de qualidade mal avaliados, isto é mal especificados). Neste estágio do processo, todos os atributos estão com seus escores associados. A nota de cada nó-pai é calculada através dos escores e dos pesos dos nós filhos.

$$\text{Nota-Nó-Pai} = \left(\sum_{i=1}^n (\text{peso} * \text{escore}) \right) / 10$$

Onde i varia do primeiro ao último filho.

Figura 3 - Fórmula para cálculo da nota de nó pai.

Seguindo a fórmula para o cálculo da nota de nó pai, Figura 3, o peso de cada atributo filho é multiplicado pelo seu escore, todos esses valores são adicionados, e a soma deve ser dividida por 10 (dez), resultando na nota do pai. Esta divisão por 10 (dez) é uma espécie de normalização, uma vez que cada nó teve o valor 10 (dez) dividido entre os pesos de seus nós filhos.

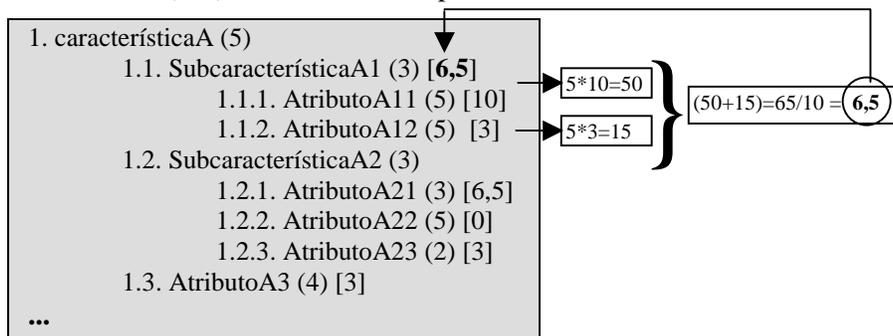


Figura 4 - Exemplo de cálculo geral.

Na Figura 4, a subcaracterística SubcaracterísticaA1 foi calculada com base nos pesos e escores de conformidade de seus filhos AtributoA11 eAtributoA12.

Na Seção seguinte é demonstrado como foi realizado o estudo de caso com a aplicação da metodologia proposta por Reis [14] em um documento de requisitos orientado a aspectos baseado no trabalho de Sousa [16].

4. Estudo de Caso

Nesse artigo a técnica de avaliação escolhida foi o estudo de caso, pois se esta interessado em avaliar o benefício da aplicação de uma metodologia de medição de qualidade em um outro contexto, diferente para o qual a metodologia foi elaborada.

O sistema escolhido para o estudo de caso foi um sistema bancário disponível na Internet, também denominado de sistema de Internet Banking. O principal objetivo desse tipo de sistema é permitir que clientes realizem transações bancárias através da Internet. Os serviços cobertos pelo sistema Internet Banking são: visualizar saldo, visualizar extrato de conta, fazer transferência e realizar pagamento de boleto bancário. O requisito não funcional que será o foco da aplicação da metodologia REQE [14] é o de segurança. Sendo que esse foi o interesse modularizado em aspectos pela abordagem de Sousa [16], Figura 1.

O estudo de caso foi elaborado com o foco de se avaliar a aplicabilidade da abordagem REQE [14] em um sistema orientado a aspectos, onde apenas um engenheiro de software foi envolvido na medição da qualidade. A seguir o estudo de caso é detalhado seguindo as fases da metodologia REQE.

4.1. Fase 1 – Representação das Características, Subcaracterísticas e Atributos de Qualidade

Esta fase é dividida em dois passos, sendo que o no primeiro é feita a escolha do domínio da aplicação e são traçados os possíveis perfis dos usuários interessados. No segundo passo é especificada a Árvore de Características, Subcaracterísticas e Atributos de Qualidade (ACSAQ).

Passo 1 – Escolha do Domínio da Aplicação e do Perfil do Interessado;

Domínio da aplicação: Site de um Sistema Bancário

Perfil do Interessado: Cliente do Banco, visitante intencional.

Na metodologia de Reis [14] esse Passo é realizado com auxílio da engenharia de domínio, porém nesse estudo de caso o domínio da aplicação e o perfil do interessado foram obtidos com base nas descrições do trabalho de Sousa [16].

Passo 2 – Especificação da árvore de Características, Subcaracterísticas e Atributos de Qualidade.

Seguindo a metodologia de Reis [14], para cada domínio de aplicação e perfil do interessado deve-se selecionar os principais requisitos não funcionais. Quando se tratar de uma medição real isso deve ser feito por meio de

questionários orientados aos clientes do banco e discussões entre as partes interessadas. Nesse estudo de caso foi feita uma adaptação do requisito de segurança a partir de catálogos oferecidos pelo *framework* NFR [5]. A árvore de Características, Subcaracterísticas e Atributos de Qualidade pode ser vista na Figura 6.

4.2. Fase 2 - Especificação Descritiva da Árvore de Características, Subcaracterísticas e Atributos de Qualidade

Nesse artigo não serão apresentadas todas as planilhas de subcaracterísticas e atributos de qualidade, devido ao volume dessa documentação. A Figura 5. mostra a planilha da característica de Segurança. O engenheiro de software deve estar ciente de que é preciso descrever cada nó detalhadamente, a fim de dirimir eventuais dúvidas a cerca de seus significados [14].

PLANILHA DE CARACTERÍSTICA
Título: Segurança - Código: 1 - Quantidade de Filhos: 4
Subcaracterísticas: (1.1.) Integridade, (1.2.) Disponibilidade, (1.3.) Confidencialidade, (1.4.) Confiabilidade
Atributos: não há.
Definição: Proteger a Informação de diversas ameaças para garantir a continuidade dos negócios, a integridade e a disponibilidade da mesma.
Peso: 10 (este peso é por ser somente essa característica que esta sendo medida)
Escore: a ser calculado posteriormente

Figura 5 - Planilha preenchida da característica Segurança.

4.3. Fase 3 – Associação de Pesos aos Nós

Nesta fase deve-se atribuir pesos aos nós, sendo que quanto mais importante for a característica, a subcaracterística ou atributo de qualidade, maior deve ser o seu peso em relação aos nós do seu nível. Esses pesos devem ser associados pelos avaliadores. A fim de comparação de aplicações Web, aplicações do mesmo domínio devem ser submetidas à mesma distribuição de pesos. Nesse estudo de caso os pesos foram atribuídos seguindo a ordem de importância dada no *framework* NFR proposto por Chung e outros [5]. A Figura 6 mostra (entre parênteses) como ficou distribuído os pesos de cada nó na ACSAQ.

4.4. Fase 4 – Associação de Escores aos Atributos

Nessa fase os avaliadores devem procurar pelos atributos descritos na árvore no documento de requisitos, sendo que para cada atributo da árvore deve ser atribuído um escore seguindo a tabela de conformidade mostrada na Figura 2.

Para buscar no documento de requisitos do sistema orientado a aspecto o atributo 1.1.2.2. Limitar Valor de Movimentação, somente foi necessário procurar entres os casos de uso aspectuais, Figura 1, e para saber quais são as descrições deste caso de uso foi necessário olhar em sua especificação, Tabela 1. Desta maneira o engenheiro de requisitos pode saber que este atributo não esta espalhado por outras partes do documento de requisitos.

Tabela 1 – Caso de uso aspectual Limitar Valor de Movimentação.

OPERACIONALIZACAO #03: Limitar valor de movimentação	
INFORMAÇÕES CARACTERÍSTICAS	
Objetivo	Verificar limite do valor de transações para diminuir riscos de fraudes.
Pré-condições	Valor da transação é dado como entrada.
Ator Primário	Cliente do banco, titular da conta.
CENÁRIO PRINCIPAL DE SUCESSO	
Passo	Ação
1	O sistema verifica se o valor da transação supera limite.
2	O sistema retorna o resultado da verificação

Como o caso de uso aspectual atende totalmente ao atributo de qualidade, para esse é dado o escore 10. Pois foi observado que o aspecto contém descrito o início (dado de entrada), meio (ação de processamento) e fim (resultado esperado). Para os outros atributos o processo de pontuação foi semelhante, sempre seguindo a conformidade mostrada na Figura 2, e buscando os atributos de segurança somente nos aspectos. A Figura 6 mostra como ficou a ACSAQ no final das atividades da fase 4.

1. Segurança (10)
1.1. Integridade (3) []
1.1.1. Completude (4) []
1.1.1.1. Protocolo de correção de erros (10) [0]
1.1.2. Precisão (6) []
1.2.1. Instalar um <i>FireWall</i> (6) [0]
1.2.2. Limitar valor de movimentação (4) [10]
1.2. Disponibilidade (2) []
1.2.1. Espelhar o banco de dados (6) [3]
1.2.2. Duplicar o Servidor (4) [3]
1.3. Confidencialidade (3) []
1.3.1. Criptografar dados (4) [6,5]
1.3.2. Autorizar acesso (6) []
1.3.2.1. Identificar usuário (5) []
1.3.2.1.1. Número da conta (5) [10]
1.3.2.1.2. Número da agência (5) [10]
1.3.2.2. Autenticar usuário (5) []
1.3.2.2.1. Checar senha (5) [10]
1.3.2.2.2. Checar dados pessoais do cliente (4) [10]
1.3.2.2.3. Checar senha adicional (1) [0]
1.4. Confiabilidade (2) []
1.4.1. Arquivo de Log (10) [6,5]

Figura 6 - Árvore para a característica de Segurança em um sistema de banco pela Internet com pesos e escores.

Os nós que não são atributos estão com os colchetes vazios, pois terão suas notas calculadas na próxima fase, seguindo a metodologia.

4.5. Fase 5 – Cálculo Geral

Nessa última fase todos os outros nós (subcaracterísticas e características) terão uma nota associada. Seguindo a formula para o calculo do nó pai, mostrada anteriormente na Figura 3, serão calculadas e atribuídas as notas como mostrado na Figura 7.

A nota final do requisito não funcional de segurança para este sistema Internet Banking, é de **4,645**, que foi calculada com a divisão por 10 da soma dos resultados da multiplicação dos pesos e escores de cada subcaracterística desse aspecto de segurança. Essa medição serviu para identificar uma série de falhas (na verdade requisitos que não foram levados em consideração no documento de requisitos ou mal especificados), que podem ser corrigidas no início do processo de desenvolvimento, acarretando a construção de um produto de melhor qualidade.

<p>1. Segurança (10) [4,645] $(3 * 2,4 + 2 * 3 + 3 * 6,75 + 2 * 6,5) = 46,45/10 = 4,645$</p> <p>1.1. Integridade (3) [2,4] $(4 * 0 + 6*4) = 24/10 = 2,4$</p> <p>1.1.1. Completude (4) [0] $(10 * 0) = 0/10 = 0$</p> <p>1.1.1.1. Protocolo de correção de erros (10) [0]</p> <p>1.1.2. Precisão (6) [4] $(6 * 0 + 4*10) = 40/10 = 4$</p> <p>1.1.2.1. Instalar um <i>FireWall</i> (6) [0]</p> <p>1.1.2.2. Limitar valor de movimentação (4) [10]</p> <p>1.2. Disponibilidade (2) [3] $(6 * 3 + 4*3) = 30/10 = 3$</p> <p>1.2.1. Espelhar o banco de dados (6) [3]</p> <p>1.2.2. Duplicar o Servidor (4) [3]</p> <p>1.3. Confidencialidade (3) [6,75] $(4 * 6,5 + 6*9,5) = 67,5/10 = 6,75$</p> <p>1.3.1. Criptografar dados (4) [6,5]</p> <p>1.3.2. Autorizar acesso (6) [9,5] $(5 * 10 + 5*9) = 95/10 = 9,5$</p> <p>1.3.2.1. Identificar usuário (5) [10] $(5 * 10 + 5*10)/10 = 10$</p> <p>1.3.2.1.1. Número da conta (5) [10]</p> <p>1.3.2.1.2. Número da agência (5) [10]</p> <p>1.3.2.2. Autenticar usuário (5) [9] $(5 * 10 + 4*10 + 1 * 0)/10 = 9$</p> <p>1.3.2.2.1. Checar senha (5) [10]</p> <p>1.3.2.2.2. Checar dados pessoais do cliente (4) [10]</p> <p>1.3.2.2.3. Checar senha adicional (1) [0]</p> <p>1.4. Confiabilidade (2) [6,5] $(10 * 6,5) = 65/10 = 6,5$</p> <p>1.4.1. Arquivo de Log (10) [6,5]</p>
--

Figura 7 - ACSAQ do sistema Internet Banking com pesos e escores.

5. Métricas para Medição da Separação de Interesses, Tamanho e Acoplamento dos Casos de Uso Aspectuais

O estudo de caso realizado mostrou que a metodologia não precisou ser adaptada para se medir a qualidade de um documento orientado a aspectos, Porém com as métricas da metodologia REQE [14] não foi possível se obter informações referentes a orientação a aspectos. Portanto tendo como base o estudo de caso apresentado e as métricas propostas por Sant'anna [15] foi possível sugerir um conjunto de métricas que avaliassem a qualidade dos aspectos (o quão separado e complexo foi a representação de um interesse em aspectos). As métricas sugeridas são divididas em três tipos, como mostrado na Figura 8.

O primeiro tipo de métrica verifica o quão separado está um determinado interesse, ou seja, em quantos casos de uso aspectuais estão modelados um determinado interesse e quantos passos foram necessários para

descrever o interesse. Esse primeiro tipo de métrica terá um valor para cada interesse, por exemplo: “Utilizando as métricas de separação de interesses foi possível verificar que o interesse de Segurança foi modularizado por 7 (sete) casos de uso aspectuais e descrito em 27 (vinte e sete) passos”.

<p>1 - Separação de interesses: Número de caso de uso aspectual que modelam um determinado interesse; Número de passos nas descrições dos casos de uso aspectuais que descrevem um determinado interesse.</p> <p>2 - Tamanho: Número de passos nas descrições de cada caso de uso aspectual; Número de passos condicionais de cada caso de uso aspectual; Número de exceções de cada caso de uso aspectual.</p> <p>3 - Acoplamento: Número de casos de uso (aspectual ou não) que são incluídos ou estendidos em um determinado caso de uso aspectual; Número de casos de uso (aspectual ou não) que são entrecortados por um determinado caso de uso aspectual.</p>

Figura 8 - Métricas para medição da separação de interesses, tamanho e acoplamento dos casos de uso aspectuais.

O segundo conjunto de métricas são relacionadas ao tamanho de cada caso de uso aspectual, indicando fatores como o número de passos, passos condicionais e exceções. Essas métricas terão um valor único para cada caso de uso aspectual, por exemplo: “Utilizando as métricas de tamanho foi possível verificar que o caso de uso aspectual Limitar Valor da Movimentação contém 2 (dois) passos e 0 (zero) exceções e passos condicionais”.

As métricas relacionadas com o acoplamento verifica a quantidade de relacionamentos tanto de entrecorte quanto de extensão e inclusão existentes em cada caso de uso aspectual.

6. Considerações Finais

A metodologia REQE [14] não necessitou ser modificada, pode-se inferir que apenas se tornou mais fácil a medição da qualidade em um documento de requisitos orientado a aspectos. Porém algumas métricas adicionais foram sugeridas para auxiliar o engenheiro de software quando decidir medir a qualidade dos aspectos.

As métricas propostas nesse artigo podem ser utilizadas separadamente da metodologia quando a intenção for saber informações no contexto do desenvolvimento orientado a aspectos. Porém um dos objetivos de utiliza-las juntamente com a metodologia REQE [14] é elaborar uma base de conhecimento para cada interesse medido.

A fase da metodologia que foi necessário a busca de informações sobre atributos foi facilitada, pois apenas foi necessário identificar no diagrama de casos de uso onde se encontrava o atributo procurado e posteriormente localizar a tabela que continha sua descrição. Diferente de quando o documento de

requisitos não é orientado a aspectos, pois neste caso o avaliador teria de fazer uma busca em todo o documento de requisitos a fim de encontrar trechos que tratam do atributo.

Como trabalhos futuros novas avaliações deverão ser realizadas com outras metodologias de medição de qualidade na fase de requisitos, para que deste modo seja possível avaliar a qualidade de qualquer documento de requisitos orientado a aspectos independente do domínio e da plataforma.

Referências Bibliográficas

1. Aksit, M.; Tekinerdogan, B. and Bergmans, L. "The Six Concerns for Separation of Concerns", in Proceedings of ECOOP 2001 Workshop on Advanced Separation of Concerns, Budapest, Hungary, June 18-22, 2001.
2. Alves, C. F., Castro, J. F. B. CRE: A Systematic Method for COTS Componentes Selection. SBES 2002, XV, Rio de Janeiro. Outubro, 2002.
3. Alves, C. F. Seleção de Produtos de Software Utilizando Uma Abordagem Baseada em Engenharia de Requisitos. Dissertação – Programa de Pós-Graduação em Ciência da Computação, UFPE, Recife – Pernambuco, maio 2002.
4. Brito, I. and Moreira, A. "Towards a Composition Process for Aspect-Oriented Requirements". Workshop on Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design, March 17 - Boston, USA, 2003.
5. Chung, L.; Nixon, B.; Yu, E. and Mylopoulos, J. "Non-Functional Requirements in Software Engineering". Boston Kluwer Academic Publishers, 2000.
6. Elrad, T., Filman, R. and Bader, A. "Aspect-Oriented Programming: Introduction", Communications of the ACM, v.44 n.10, p.29-32, Oct. 2001.
7. Jacobson, I; Chriterson, M; Jonsson, P. and Overgaard, G. "Object-Oriented Software Engineering: A Use Case Driven Approach". Addison Wesley, 1992.
8. Jacobson, I.; Booch, G.; and Rumbaugh, J. "The Unified Software Development Process", Addison-Wesley, 1999.
9. Kiczales, G.; Lamping, J.; Mendhekar, A. RG: A Case-Study for Aspect-Oriented Programming. In: SPL97. Palo Alto Research Center, Technical Report, 1997.
10. Moreira, A.; Araújo, J. and Brito, I. "Crosscutting Quality Attributes for Requirements Engineering", SEKE 2002, ACM Press, Italy, July, 2002.
11. Pressman, R. Engenharia de Software. Makron Books, 5ª edição, 2002.
12. Ramos, R. A.; Penteadó, R.; Masiero, P. C.; - "Um Processo de Reestruturação de Código Baseado em Aspectos" - SBES'2004 - Brasília/DF. Outubro, 2004.
13. Rashid, A.; Sawyer, P.; Moreira, A. and Araújo, J. "Early Aspects. A Model for Aspect-Oriented Requirements Engineering". IEEE Joint Conference on Requirements Engineering, Essen, Germany, September, 2002.
14. Reis, T., P., C. "Uma Metodologia para Medição da Qualidade de Aplicações Web na Fase de Requisitos". Dissertação – Programa de Pós-Graduação em Ciência da Computação, UFPE, Recife – Pernambuco, 2004, 163f.
15. Sant'Anna, C.; Garcia, A.; Chavez, C.; Lucena C.; Staa A. "Design Patterns as Aspects: A Quantitative Assessment". In: SBES'2004, Brasilia - DF, 2004.
16. Sousa, G. "Uma Abordagem Direcionada a Casos de Uso para o Desenvolvimento de Software Orientado a Aspectos". Dissertação – Programa de Pós-Graduação em Ciência da Computação, UFPE, Recife – Pernambuco, maio 2004, 180f.